Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 2606 – 2613

doi:10.3906/elk-1602-189

Application of reinforcement learning for active noise control

Seyed Amir HOSEINI SABZEVARI[1,*], **Majid MOAVENIAN**[2]
[1]Department of Mechanical Engineering, Faculty of Engineering, University of Gonabad, Gonabad, Iran
[2]Department of Mechanical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Received: 14.02.2016 • **Accepted/Published Online:** 02.10.2016 • **Final Version:** 30.07.2017

Abstract: Active noise control (ANC) systems are used to reduce the sound noise level by generating antinoise signals. M-Estimators are widely employed in ANC systems for updating the adaptive FIR filter taps used as the system controller. Observing the state-of-the-art M-estimators design shows that there is a need for further improvements. In this paper, a feedback ANC based on the reinforcement learning (RL) method is proposed. The sensitivity of the constant parameter in the RL method is checked. The effectiveness of the proposed method is proven by comparing the results with previous feedforward studies through computer simulations.

Key words: Active noise control, M-estimator, reinforcement learning

1. Introduction

Reinforcement learning (RL) is learning what to do and how to map situations to actions so as to maximize a numerical reward signal. Unlike most machine learning algorithms, the agent cannot decide which actions to take; it must first discover what will yield the most reward by trying each of them [1]. In RL, an agent tries to maximize a scalar evaluation (reward or punishment) obtained as a result of its interaction with the environment. To solve a particular RL problem means finding an optimal policy to map the state of the environment to an action, which in turn will maximize the accumulated future rewards [2]. In most cases, the next states are associated with actions taken by an agent, so the immediate and future rewards are affected by those actions.

RL has been theorized based on how people or animals learn. Many RL-based control systems have been successfully applied in robotics by several researchers [3–5].

Active noise control (ANC) systems are widely used to reduce the noise level, especially at low frequencies [6–8]. These systems are generating noise by a loudspeaker, called antinoise, and based on the superposition rule the primary noise level is reduced. There are two main control types of ANC systems: feedback and feedforward [9].

In the feedforward type, the controller uses the primary noise as an input to generate the antinoise by the loudspeaker and report the error signal using an error microphone. In the feedback type, the ANC system uses only an error sensor and a secondary source, not using an "upstream" reference sensor [7]. The performance of the feedback type is thus not expected to be as good as that of the feedforward, especially when dealing with unpredictable noise.

The objective of this article is to prove the feasibility of using RL techniques when controlling ANC systems and to investigate their performance by comparing the results with those of recently published papers.

*Correspondence: se_ho302@stu-mail.um.ac.ir

This paper is organized as follows: Section 2 gives an overview of the applied algorithms previously used. The new proposed algorithm is described in Section 3. Simulation results that confirm the effectiveness of the proposed algorithm are discussed in Section 4, and the concluding remarks are given in Section 5.

2. Previously applied algorithms

Block diagrams illustrated in Figure 1 show that ANC systems using the filtered-X (FX) least-mean-square algorithm require a reference signal $x(n)$ for generating the control signal $y(n)$. To drive the control signal, the reference signal $x(n)$ is designed to pass through an adaptive filter $W(z)$ to minimize the error sensor signal $e(n)$ [10,11]. The coefficients of $W(z)$ are adapted using an algorithm called the least mean M-estimator (LMME) algorithm. The M-estimator aims to reduce the effect of outliers in the data. An outlier is one that appears to deviate markedly from other members of the sample in which it occurs [6]. Several M-estimators are available so far, such as Huber [12], Hampel [13], FXLMP [14], SUNS [15], modified SUNS [16], and Fair [6]. These are composed of symmetric positive-definite functions that have minimums at zero [6].

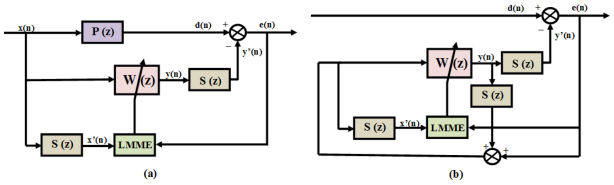

Figure 1. Block diagrams of FXLMME-based ANC system: (a) feedforward type and (b) feedback type.

2.1. Fair algorithm [12]

The Fair algorithm is a slightly modified version of the Huber, smoothing its hard limitations. The Fair function takes advantage of the L_1 and L_2 functions and the Huber function [6]. The parameters of adaptive filter $W(z)$ are updated according to the following equation [17].

$$W(n+1) = W(n) + \mu\phi(e(n))[\hat{s}(n) * x(n)] = W(n) + \mu\frac{e(n)}{1+\frac{|e(n)|}{C}}[\hat{s}(n) * x(n)] \quad (1)$$

Here, n is the time index, C indicates speed, μ is the step size, $\hat{s}(n)$ is the impulse response of $\hat{s}(z)$, and $*$ denotes the linear convolution. The stability, convergence time, and fluctuation of the algorithm are governed by the step size $\hat{s}(n)$ and $\phi(e(n))$ [10]. The effectiveness of this algorithm in comparison to previous ones was demonstrated by Wu and Qiu [6].

2.2. Reinforcement learning

In the machine-learning field, RL is a common algorithm that aims to acquire appropriate action-selection policy based on environmental rewards [5]. In contrast with supervised and unsupervised learning, RL may not use feedback for intermediate steps, but a reward (or punishment) may be given only after a learning trial has been finished [18]. The reward is scalar and indicates whether the result was right or wrong (binary) or how right or wrong it was (real value) [18]. This limited feedback characteristic makes it a relatively slow learning mechanism, but attractive due to its potential to learn action sequences that are not known by a teacher [18]. The unique features of this learning are trial-and-error searching and delayed reward [1].

3. New proposed algorithm

In the new proposed method, antinoise signals, y(n), are generated by implementing RL methods (Figure 2) instead of combining M-estimator and adaptive FIR filters (Figure 1). In the proposed method the primary noise signal is not used directly, but it is categorized as a feedback ANC type.

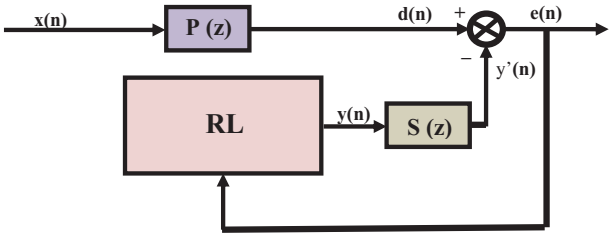

Figure 2. Block diagram of proposed feedback ANC based on RL.

The bounds of signal y(n) are between 0 and 0.025, as illustrated in Figure 3. The curve in Figure 3 is obtained as follows: first, 958 random impulse-like noises are generated and implemented in the FXLMME control algorithm (using the Fair algorithm as the LMME), and for each case the antinoise signal is calculated; then the curve is plotted by averaging the 958 antinoise signals. This band is divided into m equal parts and each part is considered as an action value and a corresponding action number.

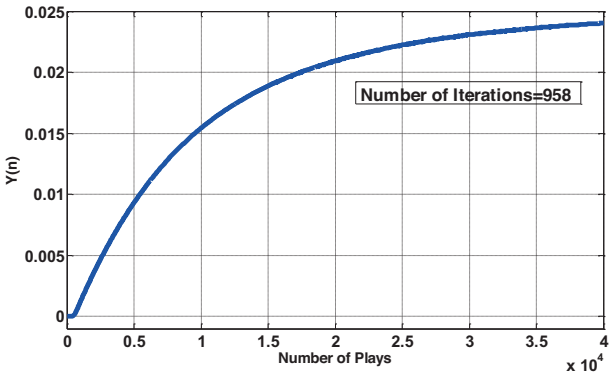

Figure 3. Average of Y(n) in 958 iterations in a FXLMME control.

$$Action\,Value\,(m,\,p) = \frac{0.025}{m-1} \times (p-1) \qquad p = 1,\,\ldots,\,m \tag{2}$$

Here, m is the number of actions and p indicates the action value index.

The rewards are assigned by a function described in the following equation. In the proposed reward function, the agent gains a reward in any state according to the error of the system. The idea here is to use a negative reward (punishment), increased by the system's error.

$$R\,(n) = -A_e\,(n) = -\left[\lambda A_e\,(n-1) + (1-\lambda)\,|e(n)|\right] \tag{3}$$

Here, λ is a constant number and is set to be $\lambda = 0.1$, and $e(n)$ is the error signal in step n illustrated in Figure 1. As a policy, ε-greedy is chosen with different ε values. The challenge here is action selection for minimization of the error signal, which will be described in Section 5.

4. Computer simulation

The performances of the algorithms based on the Fair, Huber, and Hampel functions were validated in a previous study [2]. The effectiveness of the Fair algorithm in comparison to the other functions such as Huber and Hampel was verified by Wu et al. [6]. Thus, in this section, emphasis is put on the Fair algorithm. Computer simulations were carried out to verify the effectiveness of the proposed algorithms by taking the Fair algorithm as a reference. All computer simulations in this study were performed using a PC running a MATLAB software code implemented by the authors.

4.1. Noise model

One of the most famous noise models using probability density functions is the Gaussian mixture model (GMM):

$$f(x) = (1-\gamma)\, G(x) + \varepsilon\, I(x) \tag{4}$$

Here, γ is a small constant number and $G(x)$ and $I(x)$ are Gaussian probability functions. The variance of $I(x)$ should be much larger than $G(x)$.

4.2. Performance comparison

To compare the performance of algorithms, most studies have used average noise reduction (ANR), defined as follows:

$$ANR(n) = 20\log_{10}\frac{A_d(n)}{A_e(n)} \tag{5}$$

$$A_e(n) = \lambda A_e(n-1) + (1-\lambda)\,|e(n)| \quad A_d(n) = \lambda A_d(n-1) + (1-\lambda)\,|d(n)| \tag{6}$$

Here, λ is set to be $\lambda = 0.999$ and $d(n)$ and $e(n)$ are disturbance and error signals respectively, which are illustrated in Figure 1.

4.3. Parameter value

All the constant parameters are chosen to be the same values used by Wu and Qiu [6] in order to be able to compare the results. $S(z)$, $P(z)$, and $W(z)$ are modeled by FIR filters with 250, 800, and 350 taps, consequently. The γ in the GMM is chosen as 0.05.

5. Results and discussion

In this section, the effectiveness of RL methods compared with the Fair algorithm is investigated in three different GMMs, as are shown in Table 1. The distributions of $G(x)$ and $I(x)$ are chosen to be normal with averages equal to zero. When increasing the number of cases, the impulsiveness of generating noise increases (case3 > case2 > case1).

Table 1. Different conditions for the GMM.

Noise model	Case1	Case2	Case3
$\frac{\text{variance } I(x)}{\text{variance } G(x)}$	10	100	1000

To apply the RL method, it is first necessary to specify the number of actions, the number of iterations, and the action-selection method.

Although by increasing the number of actions the controller would become more similar to a continuous controller (the performance is expected to rise), the computational complexity increases rapidly. Increasing the number of iterations may decrease the effect of randomized numbers, but the computational complexity will grow.

Similar to the N-armed bandit problem [1], three action-selection methods are chosen:

Greedy (ε-greedy, $\varepsilon = 0$)

ε-greedy, $\varepsilon = 0.01$

ε-greedy, $\varepsilon = 0.1$

To analyze the effect of action number, iteration number, and action-selection method, the performances of the Fair and RL algorithms are compared when case2 of the GMM is selected with 50,000 plays, as illustrated in Figures 4 and 5.

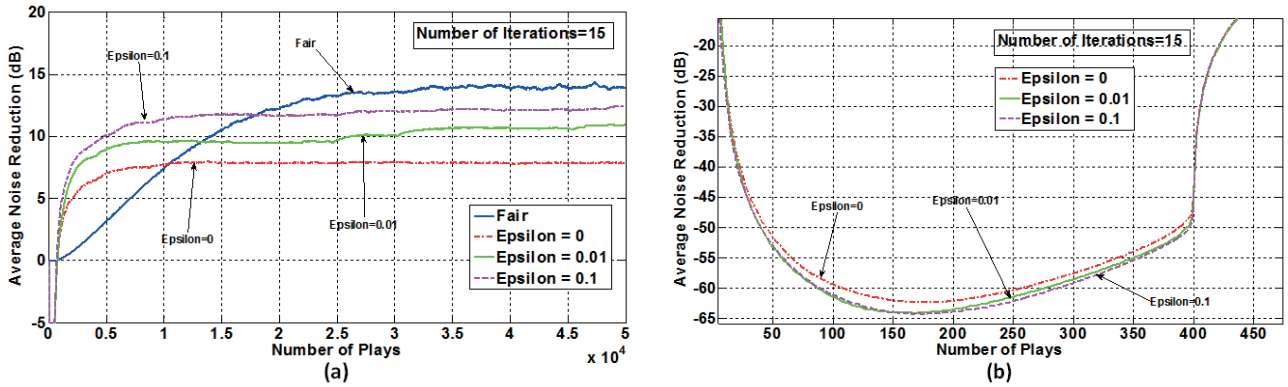

Figure 4. (a) Comparison of Fair, greedy, and ε-greedy in case2, number of actions = 11, number of Iterations = 15; (b) an arbitrary zooming of (a).

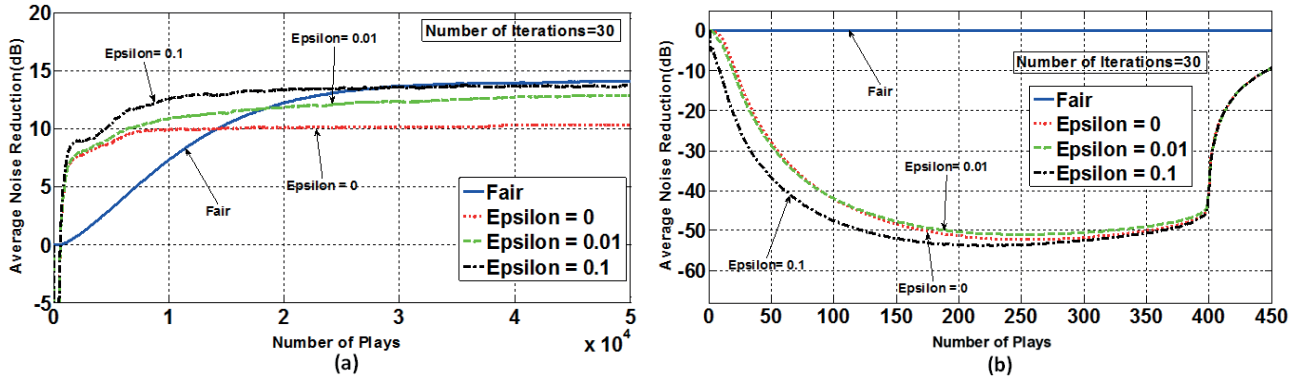

Figure 5. (a) Comparison of Fair, Greedy and ε-greedy in case-2, number of actions = 500, number of iterations = 30; (b) an arbitrary zooming of (a).

In Figure 4 the numbers of actions and iterations are set to be 11 and 15, respectively. All the FIR filters start with zero in taps value, so the absolute value of the error is growing at first. By assigning nonzero numbers in each tap, the value of $A_e(n)$ starts to decrease, so the value of ANR is rising rapidly. In Figure 5 the numbers of actions and iterations are increased from 11 to 500 and from 15 to 30, respectively. By increasing the number of actions, as it was expected, the final difference is declining for action-selection methods. In both figures,

the ε-greedy methods show better performance than the greedy method because they continue to explore for recognizing the optimal action.

The $\varepsilon = 0.1$ method explores more than $\varepsilon = 0.01$ and it is expected to find the optimal action quicker. The advantage of ε-greedy over greedy methods depends on the number of plays [1]. According to the number of plays and the results illustrated in Figures 4 and 5, the number of actions is set to be 500 and the action-selection method is chosen to be $\varepsilon = 0.1$ for the following simulation.

Figure 6 illustrates the distribution of action selection when the number of plays is 50,000. For example, Figure 6 shows that action number 238 (while the number of actions and action value index were equal to $m = 500$ and $p = 238$, respectively) was chosen 37,545 times during 50,000 plays (nearly 75% of all the action selections).

Figure 7 illustrates the error signal, $e(n)$, of the proposed method with the number of iterations and number of plays set to 1500 and 3000, consequently. The error curve can be categorized into three sections, as follows:

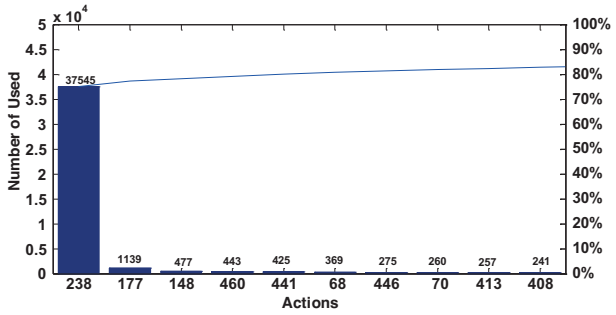

Figure 6. Distribution of action selection ε-greedy $\varepsilon = 0.1$.

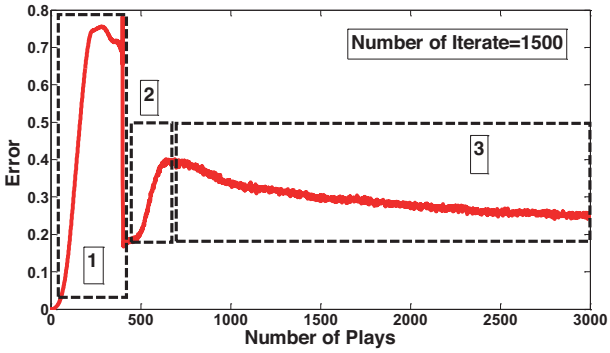

Figure 7. Error value in 1500 iterations and 3000 plays.

Section1: Increasing in error value until all the FIR taps get values

Section2: Exploring

Section3: Exploiting

As was mentioned before, in section1 taps values are changed to nonzero values. During the exploring section (section2), the value of the error is increasing rapidly because the optimal action has not yet been specified, but in the exploiting area, after finding the optimal action, the error is decreasing smoothly.

These findings coincide with the kind of error behavior mentioned in [1]. Figure 8 illustrates the performances of the Fair and RL algorithm with three different GMM noise model cases. Here the number of iterations, number of actions, and ε are equal to 20, 500, and 0.1, respectively. We see that RL shows faster initial convergence than Fair in all cases, as demonstrated in Table 2. In GMM cases, by increasing the impulsiveness of noise, the meeting point of Fair and RL curves shows a decrease from 27,785 in case1 to 25,840 in case2 and 17,770 in case3. Table 3 demonstrates the maximum performance difference between the RL method and Fair algorithm after the meeting point in GMM cases. The maximum of differences between algorithms is increased, as shown in Table 3, while the steady performances in all cases remain similar. Despite the fact that the proposed method can be categorized as feedback ANC type, because of RL qualities it shows an acceptable result in comparison to the feedforward type.

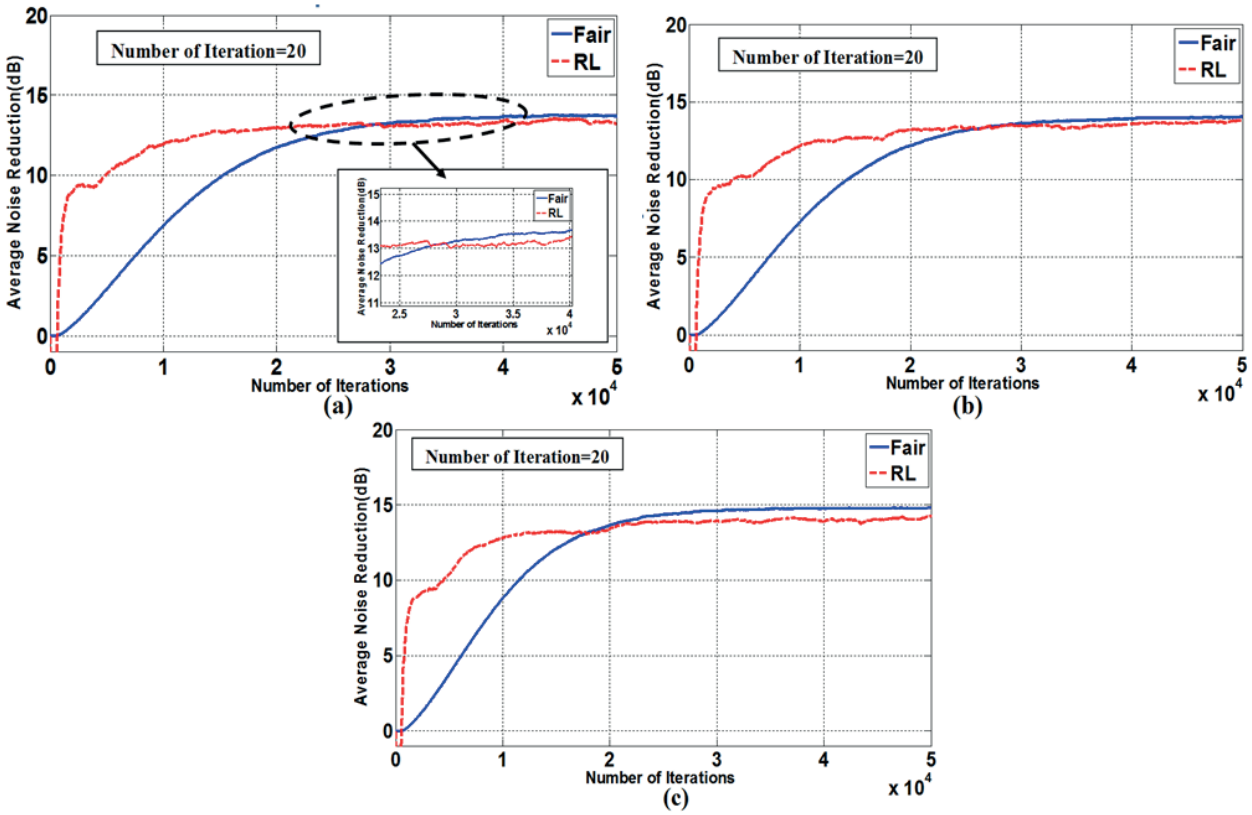

Figure 8. Comparison of Fair and RL algorithms in feedforward type: (a) case1, (b) case2, (c) case3.

Table 2. Comparison of Fair and RL convergences into 90% of their final values.

Noise model	Case1	Case2	Case3
Fair	22,515	20,721	18,305
RL	9067	11,178	11,067
Difference	13,448	9543	7238

Table 3. Maximum performance difference between RL method and Fair algorithm.

Noise model	Case1	Case2	Case3
Maximum difference value (dB)	0.42	0.48	1

6. Conclusion

In this paper, it is shown that using RL instead of the common FIR filters and M-estimators results in considerable improvement of the sensitivity with respect to the constant parameters. The computer simulations conducted proved that it is robust and the initial convergence rate is fast. By increasing noise impulses, the meeting point and the convergence differences show decreases from 27,785 and 12,448 in case1 to 17,770 and 7238 in case3, respectively, while the maximum performance differences increase from 0.42 in case1 to 1 in case3. Numerical simulations demonstrate that the proposed feedback method has faster initial convergence compared to the Fair algorithm.

Acknowledgments

The authors would like to thank Lifu Wu and Xiaojun Qiu for their responses and kind answers to our questions. We also thank Ferdowsi University of Mashhad for the support.

References

[1] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2011.

[2] El-Fakdi A, Carreras M. Two-step gradient-based reinforcement learning for underwater robotics behavior learning. Robot Auton Syst 2013; 3: 271-282.

[3] La HM, Lim R, Sheng W. Multirobot cooperative learning for predator avoidance. IEEE T Contr Syst T 2015; 23: 52-63.

[4] Fiore M, Clodic A, Alami R. On planning and task achievement modalities for human-robot collaboration. Spr Tra Adv Robot 2016; 4: 293-306.

[5] Kretzschmar H, Spies M, Sprunk C, Burgard W. Socially compliant mobile robot navigation via inverse reinforcement learning. Spr Tra Adv Robot 2016; 4: 71-83.

[6] Wu L, Qiu X. An M-estimator based algorithm for active impulse-like noise control. Appl Acoust 2013; 31: 407-12.

[7] Rout NK, Das DP, Panda G. Particle swarm optimization based nonlinear active noise control under saturation nonlinearity. Appl Soft Comput 2016; 41: 275-289.

[8] Molesworth BR, Burgess M, Chung A. Using active noise cancelling headphones to reduce the effects of masking in commercial aviation. Acta Acust United Ac 2013; 5: 822-827.

[9] Elliott SJ, Sutton TJ. Performance of feedforward and feedback systems for active control. IEEE T Speech Audi P 1996; 3: 214-223.

[10] Yang IH, Jeong JE, Jeong UC, Kim JS, Oh JE. Improvement of noise reduction performance for a high-speed elevator using modified active noise control. Appl Acoust 2014; 79: 58-68.

[11] Hart CR, Lau SK. Active noise control with linear control source and sensor arrays for a noise barrier. J Sound Vib 2012; 1: 15-26.

[12] Akhtar MT, Mitsuhashi W. Improving performance of FxLMS algorithm for active noise control of impulsive noise. J Sound Vib 2009; 3: 647-56.

[13] Thanigai P, Kuo SM, Yenduri R. Nonlinear active noise control for infant incubators in neo-natal intensive care units. In: IEEE 2007 Acoustics, Speech and Signal Processing Conference; 15–20 April 2007; Honolulu, HI, USA. New York, NY, USA: IEEE. pp. 103-109.

[14] Leahy R, Zhou Z, Hsu YC. Adaptive filtering of stable processes for active attenuation of impulsive noise. In: IEEE 1995 Acoustics, Speech and Signal Processing Conference; 9–12 May 1995; Detroit, MI, USA. New York, NY, USA: IEEE. pp. 2983-2986.

[15] Sun X, Kuo SM, Meng G. Adaptive algorithm for active control of impulsive noise. J Sound Vib 2006; 1: 516-522.

[16] Akhtar MT, Mitsuhashi W. Improving robustness of filtered-x least mean p-power algorithm for active attenuation of standard symmetric-α-stable impulsive noise. Appl Acoust 2011; 9: 688-694.

[17] Kuo SM, Morgan D. Active Noise Control Systems: Algorithms and DSP Implementations. New York, NY, USA: Wiley, 1995.

[18] Navarro-Guerrero N, Weber C, Schroeter P, Wermter S. Real-world reinforcement learning for autonomous humanoid robot docking. Robot Auton Syst 2012; 11: 1400-1407.