Quantum-behaved particle swarm optimization-based active noise control system with timing varying path

Ning Yu, Zhaoxia Li, Yinfeng Wu and Renjian Feng

Abstract
Active noise control systems can effectively suppress the impact of low-frequency noise and they have been applied in many fields. Recently, the evolutionary computation algorithm-based active noise control system has attracted considerable attention. To improve the noise reduction performance of the evolutionary computation algorithm-based active noise control system and solve the problem that the system cannot converge again when the path abruptly changes in steady state, we propose the path abruptly change-quantum-behaved particle swarm optimization algorithm. We apply quantum-behaved particle swarm optimization, a global optimization algorithm, to the active noise control system to improve noise reduction performance. In addition, the scheme of detecting the abrupt path change in steady state and performing re-convergence processing is designed to effectively address the problem that the system cannot regain convergence after a path change in steady state. The simulation study demonstrates that the proposed algorithm can efficiently improve noise reduction performance, accurately detect the path change, and re-converge to new global optimization.

Keywords
Active noise control, path change, quantum-behaved particle swarm optimization

Introduction
With increasingly serious noise pollution, active noise control (ANC) system has attracted increasingly more attention. It has been applied in aircraft cabins, submarine diesel engines, and vehicles. Compared with passive noise control, the ANC system can effectively suppress the impact of low-frequency noise with significant advantages in terms of size, weight, and cost. Based on the principle of destructive interference, the ANC system produces the necessary anti-noise (secondary noise) with the same magnitude but opposite phase as the original noise (primary noise) in the desired zone to counteract the influence of the unwanted noise.

The schematic diagram of a widely used single-channel feedforward ANC system is shown in Figure 1(a), which consists of four parts: a reference microphone, a secondary loudspeaker, an error microphone, and an adaptive controller. The reference microphone is responsible for picking up the reference noise and converting it into an electrical signal; the secondary loudspeaker generates the desired secondary noise to counteract the effect of the reference noise; the error microphone senses the residual noise, which is used to adjust the weights of the adaptive controller that generates the anti-noise.

The block diagram of the most commonly used Filtered-x least mean square (FxLMS) algorithm for the ANC system is shown in Figure 1(b). Benefiting from the simplicity of FxLMS, its prototypes and improved...
versions\textsuperscript{7,8} have been used in many ANC systems, but FxLMS has some limitations. First, the estimated model of the secondary path \( \hat{S}(z) \) is required; the model estimation error will reduce noise reduction performance and even cause instability.\textsuperscript{9–12} In addition, FxLMS is a gradient-based optimization method, which can easily fall into local minima.\textsuperscript{13}

The adaptability of the most commonly used off-line estimation \( S(z) \) method is poor. One solution is the on-line estimation of the secondary path.\textsuperscript{14–17} However, most of the on-line estimation methods require additional noise, which increases the residual noise level and computational complexity.\textsuperscript{18} Another method is evolutionary computation algorithm-based ANC system, which does not require secondary path estimation and is able to avoid the problem that the traditional gradient-based methods easily fall into local minima. Next, we introduce the evolutionary computation algorithm-based ANC systems in detail.

Genetic algorithm (GA) has been applied to ANC systems. Yim et al.\textsuperscript{13} proposed an ANC system, which uses the IIR filter as the controller and the GA to adjust the weights of the adaptive IIR filter. Russo and Sicuranza\textsuperscript{19,20} used the GA to adjust the weights of a nonlinear Volterra filter in an ANC system and demonstrates better noise reduction performance than an FxLMS-based ANC system. Chang and Chen\textsuperscript{21} proposed an adaptive GA-based linear ANC system using FIR filter and nonlinear ANC system using Volterra filter.

Compared with other evolutionary computation algorithms, particle swarm optimization (PSO) is effective in many optimization problems due to its fast and stable convergence performance.\textsuperscript{22} Motivated by the social behavior of animals, Kennedy and Eberhart\textsuperscript{23} proposed PSO in 1995. Collaboration between each particle helps the population find the global optimal solution. Recently, the PSO-based ANC system has been widely studied, and the performance of PSO-based ANC system is better than GA-based ANC system.\textsuperscript{24} Modares et al.\textsuperscript{25} used PSO to adaptively adjust the weights of the multilayer neural network in a nonlinear ANC system. Xia et al.\textsuperscript{26} used a three-layer neural network as an adaptive filter to design a PSO-based ANC system. George and Panda\textsuperscript{27} and Rout et al.\textsuperscript{24,28,29} have published a series of articles and made outstanding contributions to the online structure design, multi-channel design, path change processing, and nonlinear processing of the PSO-based ANC system. The PSO-based ANC system proposed by Rout et al.\textsuperscript{28} contains a training phase and testing phase. The training phase requires the model of the primary and secondary path to train a set of optimized filter coefficients; then, they designed an online PSO-based ANC system\textsuperscript{24} and used the modified CRPSO algorithm to address the abrupt change in primary and secondary paths. In addition, in Rout et al.,\textsuperscript{29} they used function-linked neural network as the controller to design a PSO-based nonlinear ANC system to solve the saturation nonlinearity problem of loudspeakers and microphones. George and Panda\textsuperscript{27} reached the conclusion that the Wilcoxon norm PSO-based ANC system performs better than the L2 norm PSO-based ANC system; they designed a multi-channel nonlinear ANC system,\textsuperscript{18} which uses a functionally linked artificial neural network as the controller and the PSO algorithm to train the weights of the network.

The above evolutionary computation algorithm-based ANC systems show great advantages over the ANC systems, which require on-line estimation of the secondary path. They do not require the estimation of the secondary path and prevent the local minima problem. Recently, quantum-behaved particle swarm optimization (QPSO) proposed by Sun et al.\textsuperscript{30,31} has been widely used. As a promising global optimization algorithm, QPSO has been applied to many fields such as spectral extinction measurements,\textsuperscript{32} economic dispatch,\textsuperscript{33} solving nonlinear equations,\textsuperscript{34} and processing medical image.\textsuperscript{35,36} For the application of medical image, Li et al.\textsuperscript{35} proposed a dynamic-context cooperative QPSO with enhanced search ability for processing medical images and Li et al.\textsuperscript{36}
proposed a SCQPSO algorithm to optimize the parameters for image segmentation of stomach CT images. The quantum theory is introduced into QPSO and the particles can search in the whole search space. In this paper, global optimization algorithm QPSO is applied to the ANC system to achieve better global convergence and noise reduction performance. This is one contribution of this article.

When the evolutionary computational algorithm enters steady state, the population converges to a satisfactory solution and loses the ability to adapt to environmental changes. In general, the ANC system based on the evolutionary computation algorithm cannot converge again after the primary or secondary path abruptly changes. Some literature presents methods to deal with the problem when the path is subject to change during control system operation and thus differs from the model. To solve this problem, we design a scheme for detecting the path change in steady state and performing re-convergence processing, which can effectively solve the problem that the system cannot re-converge after path change and can tolerate interference in the environment. This is another contribution of this article.

The above-mentioned method of improving the global convergence performance and dealing with path change in steady state is called path abruptly change-QPSO (PAC-QPSO). PAC-QPSO improves global convergence performance by introducing the QPSO method to the ANC system, which ensures that the particles can converge to a better optimal solution; PAC-QPSO can effectively address the path change and tolerate interference in the environment, which improves the robustness of the system. Simulation results show that compared with the existing methods, PAC-QPSO can improve noise reduction performance and effectively address the path change in steady state.

This article is organized as follows. The next section introduces the proposed PAC-QPSO algorithm in detail. The QPSO theory is introduced into the ANC system first, and then we solve the path change problem in steady state. In the “Simulation analysis” section, a series of simulations are carried out. The final section summarizes the investigation.

The proposed PAC-QPSO algorithm

Global optimization algorithm QPSO-based ANC system

As a global optimization algorithm, QPSO has attracted increasing attention because of its outstanding optimization ability. In this paper, we first introduce QPSO into the ANC system.

The particles in QPSO obey the quantum mechanics theory. According to the uncertainty principle, the position and velocity of a particle cannot be determined simultaneously. The state of the particle is determined by the wave function $\psi(x, t)$. Depending on the statistical significance of the wave function $\psi(x, t)$, the probability that a particle appears at a certain position can be obtained from the probability density function $|\psi(x, t)|^2$. The searching ability of QPSO is strong because the particle can appear anywhere during generation. In this paper, the QPSO algorithm is applied to the ANC to improve noise reduction performance.

The QPSO-based ANC system can transform into an optimization problem to minimize the mean square error (MSE) received by the error microphone. Then, the error signal is used to adjust the weights of the adaptive filter, generating secondary noise to counteract the effects of reference noise in the desired zone. Figure 2 shows the block diagram of the QPSO-based ANC system.

Collaboration among the populations contributes to find the global optimal solution. This process includes fitness evaluation, optimal position updating, and particle position updating. The control process of the QPSO-based ANC system specifically includes the following steps, and the corresponding flowchart is plotted in Figure 3.
**Step (1): Set the parameters.** We use $P$ adaptive filters and each represents a particle. The dimension of each particle (the order of the filter) is $D$. The whole population can be expressed as follows, where each row in $W$ represents the coefficient vector of an adaptive filter and $n$ represents the current sampling time

$$W(n) = \begin{bmatrix} w_1^1(n) & w_2^1(n) & \cdots & w_P^1(n) \\ w_1^2(n) & w_2^2(n) & \cdots & w_P^2(n) \\ \vdots & \vdots & \ddots & \vdots \\ w_1^P(n) & w_2^P(n) & \cdots & w_P^P(n) \end{bmatrix}$$  

**Step (2): Initialize the population.** Initialize the position $X_i(t)$ of all the particles in first iteration, where $i = 1, 2, \ldots, P$, $t = 1$.

**Step (3): Signal filtering.**

1. The reference signal $x(n)$ transforms to $d(n)$ after traveling through the primary path $P(z)$.
2. The reference signal $x(n)$ is fed to the controller and outputs $y(n)$. At any sampling time $n$, only one particle is used for filtering. The particle is selected rotationally after every $M$ samples.
3. The anti-noise signal $y_0(n)$ is generated from the control signal $y(n)$ by filter through the secondary path $S(z)$.

**Step (4): Signal counteraction.** In the desired zone, the secondary acoustic signal $y'(n)$ counteracts the single $d(n)$ and the residual noise $e_i(n)$ is given by

$$e_i(n) = d(n) + y'_i(n)$$  

**Step (5): Calculate fitness.** The fitness value $f_i$ of the $i$th particle or the fitness $f[X]$ of the position $X$ can be calculated by using $e_i(n)$. Generally, the $MSE$ is selected as the fitness function. The $f_i$ at the $t$th iteration is

$$f_i(t) = \frac{1}{M} \sum_{n=1}^{M} e_i^2(n)$$
$e_i(n)$ are the $M$ residual noises at $r$th iteration.

**Step (6): Update the optimal position.**

1. Update the personal best position $p_{best_i}(t)$: Compare the fitness of current iteration $f_i(t)$ with the fitness of personal best position in the last iteration $f[p_{best_i}(t-1)]$.

\[
\text{If } f_i(t) < f[p_{best_i}(t-1)] \\
\text{Then, } p_{best_i}(t) = X_i(t) \\
\text{Otherwise, } p_{best_i}(t) = p_{best_i}(t-1) \quad t > 1
\]

In general, the position $X_i(1)$ is selected as the respective $p_{best_i}(1)$ in the first iteration.

2. Update the global best position $g_{best}(t)$: Compare the smallest fitness $f[p_{best_{min}}(t)]$ with the fitness of global best position in the last iteration $f[g_{best}(t-1)]$, where $p_{best_{min}}(t) = p_{best_i}(t)$, $g = \arg \min_{1 \leq i \leq P} \{f[p_{best_i}(t)]\}$

\[
\text{If } f[p_{best_{min}}(t)] < f[g_{best}(t-1)] \\
\text{then } g_{best}(t) = p_{best_{min}}(t) \\
\text{otherwise } g_{best}(t) = g_{best}(t-1) \quad t > 1
\]

In the first iteration, the global best position $g_{best}(1)$ is the position $X_i(1)$ with the smallest fitness, where $g_1 = \arg \min_{1 \leq i \leq P} \{f[X_i(1)]\}$ or $g_1 = \arg \min_{1 \leq i \leq P} \{f_i(1)\}$.

**Step (7): Update the position of the particles.** The position of the particles is updated as follows by using the personal best position $p_{best_i}(t)$ and the global best position $g_{best}(t)$

\[
C(t) = \frac{1}{P} \sum_{i=1}^{P} p_{best_i}(t) \tag{10}
\]

\[
p_i(t) = \varphi p_{best_i}(t) + (1 - \varphi)g_{best}(t) \tag{11}
\]

\[
X_i(t+1) = p_i(t) + \xi_t[C(t) - X_i(t)] \ln \left(\frac{1}{u}\right) \tag{12}
\]

Equation (10) calculates the average optimal position $C(t)$ of the population by the personal best position $p_{best_i}(t)$; equation (11) calculates the attractor $p_i(t)$ at the $r$th iteration, based on the personal best position $p_{best_i}(t)$ and global best position $g_{best}(t)$, where $\varphi$ represents the random number between (0,1). Based on the $p_i(t)$, $C(t)$, and $X_i(t)$, the position $X_i(t+1)$ at the $t+1$ th iteration can be calculated, where $\xi$ is a random number between (0,1) and $x$ denotes contraction–expansion coefficient, which is the key parameter in the QPSO algorithm and should be less than 1.781 for the convergence.

**Step (8): Perform iterative operations.** Repeat steps (3)–(7) to update the particle’s position. The anti-noise constantly counteracts the effect of the reference noise, forming a quiet zone.

The QPSO algorithm continually updates the position of the particles according to equations (10), (11), and (12). After several iterations, all of the particles cluster near the global optimal position and the system enters into steady state. The particle in QPSO can search in the entire search space, which guarantees a better optimal solution. Hence, applying QPSO to the ANC system can improve the noise reduction performance. The
The computation complexity of the algorithm is as Table 1. The parameter $P$ represents the number of particles and $D$ represents the dimension of particles.

### Table 1. The computation complexity of the algorithm

|                      | Addition     | Multiplication | Logarithm |
|----------------------|--------------|----------------|-----------|
| QPSO-based ANC system| $3D + (P - 1)D$ | $5D$           | $D$       |

ANC: active noise control; QPSO: quantum-behaved particle swarm optimization.

computation complexity of the algorithm is as Table 1. The parameter $P$ represents the number of particles and $D$ represents the dimension of particles.

The solution of path change in steady state

The ANC system will lead to abrupt path change when the environment changes after it comes to steady state. Generally, the diversity of the ANC system based on the evolutionary computation method is single in steady state. If the path abruptly changes, the population cannot regain convergence. To solve this problem, we design a scheme to detect the path change in steady state and perform re-convergence processing.

When the path abruptly changes in steady state, path change detection and re-convergence processing are necessary.

**Path change detection.** Since the effect of the path change is long-term, intervention is required to achieve stabilization. The causes of these abrupt changes include sudden state changes of the door or window in indoor noise reduction, abrupt changes of the position of the secondary loudspeaker or the error microphone, etc. Chronic changes due to device aging or other factors are not taken into account.

**Path change detection.** The CRPSO proposed in Rout et al.\textsuperscript{24} uses the information of three continuous iterations to address the path change. The re-initialization operation is performed to restore the diversity of the population after the path change is confirmed. In CRPSO, the square error value of the global best ($g_{best}$) particle $E_{g_{best}}^2$ is selected as the criterion.

\begin{align}
\text{If } |E_{g_{best}}^2(t - 1) - E_{g_{best}}^2(t - 2)| &\leq k_1 \\
\text{and } |E_{g_{best}}^2(t) - E_{g_{best}}^2(t - 1)| &\geq k_2
\end{align}

![Figure 4. The convergence curve of the ANC system when the path changes. AMSE: accumulated mean square error.](image-url)
then reinitialize the particle, where \( k_2 > k_1 \), \( k_1 \) is a smaller number and \( k_2 \) can be experimentally determined. The results show that the algorithm can find the \( g_{best} \) again after re-initialization, but it will judge the fluctuations in the convergence process, as presented in Figure 4, as path changes, causing re-initialization; the selection of \( k_1 \) and \( k_2 \) is related to the input signal strength.

We choose the accumulated mean square error (AMSE) of all particles in each iteration as the criterion, which can better reflect noise reduction performance, where

\[
AMSE = \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{P} e_i^2(n).
\]

Our intention is to tolerate fluctuations in the convergence process, but accurately detect the abrupt path change in steady state.

We propose an online method to detect abrupt path changes. We select \( S \) data in each iteration by using the sliding window, as shown in Figure 5. After removing the maximum and minimum, calculate the mean and variance of the residual data. The data in the sliding window in the \((t + S)\)th iteration is

\[
SW(t + S) = AMSE[t + 1, t + 2, \ldots, t + S].
\]

We define the anomaly factor \( F \) based on the mean and standard deviation of the data \( SW(t + S) \). The anomaly factor \( F \) is given by

\[
F = \frac{AMSE(t + S) - \text{mean}[SW(t + S)]}{\text{std}[SW(t + S)]}
\]

When \( F \) exceeds a certain threshold, it is considered a path change.

In the initial stage of the algorithm, when the amount of AMSE is less than \( S \), we use 0 to enrich the rest. The algorithm starts from the second AMSE, because the anomaly factor \( F \) cannot be calculated when there is only one value.

As seen from equation (15), the anomaly factor \( F \) is independent of the input signal strength. When the strength of the input signal becomes \( a \) times the original, all the factors change into

\[
AMSE(t + S) \rightarrow a^2 \cdot AMSE(t + S)
\]

\[
\text{mean}[SW(t + S)] \rightarrow a^2 \cdot \text{mean}[SW(t + S)]
\]

\[
\text{std}[SW(t + S)] \rightarrow a^2 \cdot \text{std}[SW(t + S)]
\]

\[
F_{\text{new}} = \frac{a^2 \cdot AMSE(t + S) - a^2 \cdot \text{mean}[SW(t + S)]}{a^2 \cdot \text{std}[SW(t + S)]} = F
\]

From equations (16) to (19), we can see that the anomaly factor \( F \) is independent of the input signal strength. The method will not misjudge the fluctuation in the convergence process as paths change. Assuming that under the same changing conditions, that is fluctuation and path change have the same AMSE, as shown in Figure 4, the \( F_1 \) in \( t_1 \)th iteration is expressed as

\[
F_1 = \frac{AMSE - \text{mean}_1}{\text{std}_1}
\]
and the $F_2$ in $t_2$ th iteration is

$$F_2 = \frac{AMSE - \text{mean}_2}{\text{std}_2}$$

(21)

where subscript 1 and subscript 2 denote the related value in the fluctuation and the path change moment, respectively. Due to $\text{mean}_1 > \text{mean}_2$ and $\text{std}_1 \gg \text{std}_2$, $F_1 \ll F_2$. There is a peak when the path changes in steady state and this progress will not misjudge the fluctuations.

The path change detection method proposed can adapt to the input single strength, so it is not necessary to set the threshold value associated with the input signal strength. In addition, it will not judge the fluctuations in the convergence process as the path changes.

Occasional interference processing. In the actual environment, short-time interference commonly exists in ANC systems, such as the noise generated by passing-by automobiles in engine noise reduction. The inevitable interference at the error microphone will be misjudged as path change. If the impact of interference is ignored, the system will misjudge the interference as path change and perform unnecessary re-initialization. Therefore, it is necessary to consider the influence of the interference to ensure the noise reduction performance and the robustness of the system. Path change and disturbance are collectively called anomaly.

We assume that the longest duration of interference that the system can tolerate is $h$ iterations and the sampling frequency is $f$. Then, the maximum duration of tolerable interference is given by

$$T_c = h \cdot \frac{P \cdot M}{f}$$

(22)

We need to use $H = [h + 2]$ times-iteration of $AMSE$ to judge whether the anomaly is caused by interference or path change. The flowchart of the judgment process is shown in Figure 6. If $h = 1$ iteration and the sampling frequency $f = 8 \text{ kHz}$, $P = 100$, $M = 200$, then the maximum duration of interference that the system can tolerate is $T_c = 2.5 \text{ s}$. After judging an anomaly for three consecutive times, path change is proven to be the reason for the anomaly.

![Figure 6. Flowchart for detecting path change in an occasional interference-existing condition.](image-url)
When the anomaly factor $F$ exceeds the threshold, the algorithm judges that an anomaly occurs. If the anomaly occurs in the $(m+S)$ th iteration, the data in the sliding window are $SW(m+S) = AMSE[m+1, m+2, \ldots, m+S]$. Then, we need to further determine whether the anomaly is caused by interference or path change. In the $(m+S+1)$ th iteration $SW(m+S+1) = AMSE[m+1, m+2, \ldots, m+S-1, m+S+1]$ if the detection result of this iteration is normal, it indicates that the anomaly is caused by interference; otherwise, it requires further detection. In the $(m+S+2)$ th iteration $SW(m+S+2) = AMSE[m+1, m+2, \ldots, m+S-1, m+S+2]$, the detection result of anomaly indicates that the path is changed, hence it must perform re-initialization and empty the data in the sliding window; otherwise, it is interference in iteration $(m+S)$.

When the path change is detected in the presence of interference, if the longest duration of the tolerable interference is $h$ iterations, the system uses $H = [h + 2]$ iterations information to determine the cause of the anomaly. Because the system cannot determine whether there is interference in the environment in advance, a method that can be applied in both general conditions and interference-existing conditions is needed. We can set $h$ on the basis of the actual condition, adjust it according to the needs of the system, and improve the robustness of the system.

**Re-convergence processing.** The population has to be reinitialized to restore diversity after confirming path change. To further improve the global convergence performance of the algorithm, we propose to update the $pbes$ and $gbest$ by using the information of the change point in the first iteration after re-initialization to speed up the convergence rate and improve noise reduction.

In steady state, the particles gather near the $gbest$ as shown in Figure 7(a); after a path change, the $gbest$ is transferred. However, the population loses the ability to find the $gbest$ again and the particles continue to maintain the previous position as indicated in Figure 7(b); in order to regain convergence, the population must be reinitialized. Figure 7(c) shows the position of the particles after re-initialization. To further improve global convergence performance, we make full use of the information of the change point. In the first iteration after re-initialization, the particle itself is usually chosen as the $pbes_i(t)$. We compare the fitness of the particle with the fitness in the last iteration when an abrupt path change is confirmed as displayed in Figure 7(d) and choose the position with the smaller fitness as $pbes_i(t)$. The $gbest(t)$ is the position with the smallest fitness in $pbes_i(t)$.

Assuming that the path change is confirmed in the $T$th iteration, the fitness of each particle is $f_i(T)$ and the fitness after re-initialization is $f_i(T+1)$. In the first iteration after re-initialization, $f_i(T)$ and $f_i(T+1)$ are compared, just as the marked particle shown in Figure 7, and the position with smaller fitness is chosen as the $pbes_i(T+1)$. If $f_i(T) < f_i(T+1)$, then $pbes_i(T+1) = X_i(T)$, where $X_i(T)$ denotes the position when the path change is confirmed; otherwise the $pbes_i(T+1)$ is the reinitialized position of the particle $i$, that is $pbes_i(T+1) = X_i(T+1)$. The position with the smallest fitness among $pbes_i(T+1)$ is $gbest(T+1)$, setting $pbest_{\min}(T+1) = pbest_{\min}(T+1)$, $g = \arg\min_{1 \leq i \leq p} \{f[pbest(T+1)]\}$, so $gbest(T+1) = pbest_{\min}(T+1)$. Then, according to the method in the “Global optimization algorithm QPSO-based ANC system” section to update the position of the particles, Figure 8 shows the corresponding process.

We use the information of the change point to optimize the $pbes_i$ and $gbest$ in the first iteration after re-initialization. The benefits are that the convergence speed and noise reduction performance are improved because the search range of the optimal solution is reduced. The algorithm convergence can be referred to in Sun et al.\(^{38}\)

![Figure 7. Schematic diagram of particle’s position change when path changes in steady state. (a) Position of the particles in steady state, (b) position of the particles after path change, (c) position of the particles after re-initialization, and (d) fitness comparison between the position after path change confirmed and the position after re-initialization.](image-url)
In conclusion, the contributions of the proposed algorithm on the issue that an evolutionary computation algorithm-based ANC system cannot regain convergence when the path abruptly changes in steady state are summarized as follows: (1) It can accurately identify the abrupt path change in steady state. This process will not misjudge the fluctuations in the convergence process. (2) It can tolerate interference in the environment, thereby improving the robustness of the system. (3) It makes full use of the information of change point to improve the convergence rate and global convergence performance.

Simulation analysis

To verify the effectiveness of the proposed PAC-QPSO algorithm, a series of simulation experiments are carried out in this section. The noise signal used in the experiment is a Gaussian white noise with a power of 1 dBW. The pure time-delay system model is used to simulate the transfer function of the path. The transfer functions of the primary path and the secondary path are

\[ P(z) = z^{-5} - 0.3z^{-6} + 0.2z^{-7}, \ S(z) = z^{-2} + 1.5z^{-3} - z^{-4}, \]

respectively.\(^{27}\) The various parameters used in our studies are \( P = 100, D = 10, \) and \( M = 200. \) In all experiments, we mentioned the primary path change; this refers to the transfer function of primary path changes from \( P(z) = z^{-5} - 0.3z^{-6} + 0.2z^{-7} \) to \( P'(z) = z^{-5} - 1z^{-6} + 0.2z^{-7}. \) Similarly, the secondary path change means the transfer function of change from \( S(z) = z^{-2} + 1.5z^{-3} - z^{-4} \) to \( S'(z) = z^{-2} + 1.5z^{-3} - 3z^{-4}. \) All the path changes occur at the beginning of the 100th iteration.

The comparison of noise reduction and global best position fitness performance

Evolutionary computation algorithms applied to ANC systems primarily include the GA and PSO algorithm; performance of the PSO-based ANC system is superior to the GA-based ANC system.\(^{24}\) Therefore, this section compares the noise reduction performance of the PSO and QPSO-based ANC system. The PSO algorithm cannot converge to the global optimization with probability one.\(^{39}\) The QPSO algorithm is a global optimization algorithm and can guarantee a better optimal solution. To compare the performance of these two algorithms, the parameter settings in the PSO algorithm are the same as those of the QPSO algorithm. The mean of \( AMSE \) of 100

---

**Figure 8. Flowchart of convergence processing.**
independent runs is shown in Figure 9. In the simulation, the contraction–expansion coefficient of QPSO is \( \alpha = 0.5 \). The contraction–expansion coefficient \( \alpha \) controls the convergence speed of the algorithm. When it takes a large value, the algorithm has a strong global search ability and the convergence speed is fast, which is beneficial to step out of the local optimum. When it takes a small value, the algorithm has strong local search ability and the convergence speed is slow, which is conducive to obtain an exact solution. It should be ensured the contraction–expansion coefficient \( \alpha < 1.781 \) to guarantee the convergence of the algorithm.

For the ANC system, the merits of noise reduction performance are the most critical index. Figure 9(a) shows that the AMSEs of PSO and QPSO are 0.2880 and 0.0802, respectively, in the 200th iteration and the convergence accuracy of QPSO is about an order higher than that of PSO. Figure 9(b) shows the comparison of the global best position fitness of PSO and QPSO, namely \( MSE_{gbest} \). It can be seen that QPSO algorithm is superior to PSO algorithm in global optimization found in each iteration. In steady state, each particle in QPSO converges to a better global optimal position and its noise reduction performance is improved compared with PSO algorithm.

The performance comparison of path change processing

The phenomenon of path change. When the evolutionary computational algorithm is in steady state, the population converges to a satisfactory solution and loses the ability to adapt to environmental changes. The diversity of the evolutionary computation algorithm-based ANC system in steady state is single. In general, the ANC system based on the evolutionary computation algorithm cannot converge again when the primary or secondary path changes. If the path abruptly changes, the ANC system cannot find the \( g_{best} \) again. To analyze the effects of the
primary and secondary path change in steady state of the QPSO-based ANC system, the AMSE with respect to
the generation is plotted in Figure 10 for 10 independent runs.

Figure 10(a) indicates that the ANC system no longer has the ability to find the gbest after the primary path
abruptly changes in steady state; Figure 10(b) shows the same result when a secondary path change occurs in
steady state. Therefore, it is necessary to reinitialize the population when a path change occurs in steady state to
regain convergence.

The overall performance comparison of path change processing. The performance of path change processing of the CR-
QPSO and PAC-QPSO is compared in this section. CR-QPSO uses the same judgment method as CRPSO to
detect path change and the QPSO algorithm is used to update the coefficients of the filter; PAC-QPSO is the path
change processing method proposed in this paper. CR-QPSO uses the difference between two adjacent iterations
among three continuous iterations to determine whether the path changes or not, where the parameters are
\(k_1 = 0.03\), \(k_2 = 0.4\); PAC-QPSO uses the anomaly factor \(F\) to detect path change, where the threshold is 20. The
simulation results of each method with 10 independent runs are shown below.

Figure 11 illustrates that the performance of CR-QPSO is similar to the threshold method proposed in Rout
et al. They will both misjudge the fluctuations in the convergence process and cause unnecessary re-initialization;
Figure 12 demonstrates that the proposed PAC-QPSO can accurately detect path change in steady state and will
not cause misjudgment.

Figure 11. CRPSO. (a) Primary path change and (b) secondary path change. The changes happen at the beginning of the 100th
iteration. The changes happen at the beginning of the 100th iteration. AMSE: accumulated mean square error.

Figure 12. PAC-QPSO. (a) Primary path change and (b) secondary path change. The changes happen at the beginning of the 100th
iteration. AMSE: accumulated mean square error.
The analysis of the anomaly factor $F$. The PAC-QPSO algorithm can accurately detect path change in steady state by using the anomaly factor $F$ ("The solution of path change in steady state" section). In the ANC process, the first initialized population enters into steady state after several iterations and the primary and secondary path abruptly change in the 100th iteration. The size of the sliding window is $S = 7$. Figure 13(a) and (b) displays the mean of the anomaly factor $F$ with 10 independent runs when the transfer function of the primary and the secondary path abruptly changes.

The anomaly factor $F$ changes substantially when there is a path change in steady state. Figure 13(a) shows that $F$ approximates 500 when the primary path changes; similarly, $F$ exceeds 400 when the secondary path changes as shown in Figure 13(b). From the change of the $F$, we can clearly detect the path change. Within several iterations after the first initialization, $F$ remains below 10; therefore, using $F$ we can accurately detect the path change in steady state and will not misjudge the fluctuations in the gradual convergence process.

The analysis of occasional interference processing. To analyze the influence of interference on the detection of the path change, a simulation is executed. In the simulation, the interference is a sinusoidal signal with a frequency $f_d = 200$ Hz and amplitude of 10, which lasts $h = 1$ iteration. The performance of interference and secondary path change processing is plotted in Figure 14.

Figure 14(a) and (b) indicates the noise reduction performance of 10 independent runs when interference and secondary path change occur. In Figure 14(a), because of the short-time interference, the anomaly factor $F$ changes greatly but recovers to normal later. After the interference disappears, the system will maintain the

---

**Figure 13.** The anomaly factor $F$ with timing varying path. (a) Primary path change and (b) secondary path change. The changes happen at the beginning of the 100th iteration.

**Figure 14.** The performance of processing interference and secondary path change. (a) Interference and (b) secondary path change. The changes happen at the beginning of the 100th iteration. AMSE: accumulated mean square error.
previous convergence state; in Figure 14(b), since the effect of the secondary path change is long-term, the anomaly factor $F$ will maintain a large value if there is no intervention, and the system will regain convergence after intervention processing. The system can accurately detect the path change in steady state and tolerate interference in the environment.

The analysis of re-convergence performance. After detecting the path change, re-initialization is required to restore the diversity of the population. We make full use of the iteration information when the path change is confirmed. The $pbest_i(t)$ of the first iteration after re-initialization is chosen as the position with the smaller fitness from the current iteration and the last iteration; $gbest(t)$ is the position with the smallest fitness in $pbest_i(t)$. The performance of the system with change point information and without is presented in Figures 15 and 16. Figure 15 shows the mean value of the AMSE of 100 independent runs in both cases when the primary path abruptly changes from $P(z)$ to $P'(z)$. Figure 16 compares the results when the secondary path transfer function abruptly changes from $S(z)$ to $S'(z)$. It can be seen from Figure 15 that the number of iterations required to converge to the steady state reduces 5 times after using the change point information, and the convergence accuracy is changed.

Figure 15. The comparison of noise reduction performance when the primary path changes at the beginning of the 100th iteration with interference in the environment. AMSE: accumulated mean square error.

Figure 16. The comparison of noise reduction performance when the secondary path changes at the beginning of the 100th iteration with interference in the environment. AMSE: accumulated mean square error.
from 0.153 to 0.143 at the 200th iteration. In Figure 16, after using the change point information, the number of iterations required to converge to the steady state reduces 7 times, and the convergence accuracy is changed from 0.410 to 0.224 at the 200th iteration. The reason is that $pbes(t)$ is optimized by using the information of the change point, so the particles can find the $gbest(t)$ more quickly, and the narrowed range of the $gbest(t)$ improves convergence accuracy.

**Conclusion**

The PAC-QPSO algorithm proposed in this paper can effectively improve noise reduction performance of the ANC system. Generally, the evolutionary computation algorithm-based ANC system cannot regain convergence when the path abruptly changes in steady state. The algorithm we proposed can address path change in steady state and interference in the environment. Re-convergence processing after path changing further improves the convergence speed and accuracy of the system. A series of simulations illustrate that the noise reduction performance of the proposed method is superior to that of the PSO-based ANC system.

**Acknowledgement**

The authors are grateful to the anonymous reviewers for their industrious work and insightful comments.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by Defense Industrial Technology Development Program (JSJL2018205A002), the National Key R&D Program of China (2017YFF0204905), and National Natural Science Foundation of China (61671039).

**ORCID iD**

Ning Yu https://orcid.org/0000-0002-6532-3770

**References**

1. Haase T, Unruh O, Algermissen S, et al. Active control of counter-rotating open rotor interior noise in a Dornier 728 experimental aircraft. *J Sound Vib* 2016; 376: 18–32.
2. Sachau D, Jukkert S and Hövelmann N. Development and experimental verification of a robust active noise control system for a diesel engine in submarines. *J Sound Vib* 2016; 375: 1–18.
3. Cheer J and Elliott SJ. Multichannel control systems for the attenuation of interior road noise in vehicles. *Mech Syst Signal Process J* 2015; 60: 753–769.
4. Kuo SM. Adaptive active noise control systems: Algorithms and digital signal processing (DSP) implementations. In: *Proceedings of SPIE – Digital Signal Processing Technology: A Critical Review*, Orlando, United States, 17–21 April 1995, pp. 23–49. Bellingham: SPIE.
5. Sun G, Feng T and Li M. Convergence analysis of FxLMS-based active noise control for repetitive impulses. *Appl Acoust* 2015; 89: 178–187.
6. Ryu KW, Hong CS and Shin CJ. Active noise control of a closed rectangular cavity using FXLMS algorithms. *Trans Korean Soc Noise Vib Eng* 2011; 21: 983–990.
7. Akhtar MT and Mitsubishi W. A modified normalized FxLMS algorithm for active control of impulsive noise. In: European signal processing conference, Aalborg, Denmark, 23–27 August 2010, pp.1–5. Piscataway, NJ: IEEE.
8. Huang B, Xiao Y, Sun J, et al. A variable step-size FXLMS algorithm for narrowband active noise control. *IEEE Trans Audio Speech Lang Process* 2013; 21: 301–312.
9. Ardekani IT and Abdulla WH. Effects of imperfect secondary path modeling on adaptive active noise control systems. *IEEE Trans Control Syst Technol* 2012; 20: 1252–1262.
10. Tobias OJ, Bermudez JCM and Bershad NJ. Mean weight behavior of the filtered-X LMS algorithm. *IEEE Trans Signal Process* 2000; 48: 1061–1075.
11. Saito N and Sone T. Influence of modeling error on noise reduction performance of active noise control systems using filtered-X LMS algorithm. *J Acoust Soc Jpn* 1996; 17: 195–202.
12. Snyder SD and Hansen CH. The effect of transfer function estimation errors on the filtered-x LMS algorithm. *IEEE Trans Signal Process* 1994; 42: 950–953.
13. Yim KH, Kim JB, Lee TP, et al. Genetic adaptive IIR filtering algorithm for active noise control. In: Proceedings IEEE international conference fuzzy systems, Seoul, South Korea, South, 22–25 August 1999, pp.1723–1728. Piscataway, NJ: IEEE.

14. Lopes PAC and Gerald J. Auxiliary noise power scheduling algorithm for active noise control with online secondary path modeling and sudden changes. IEEE Signal Process Lett 2015; 22: 1590–1594.

15. Akhtar MT, Abe M, Kawamata M, et al. Online secondary path modeling in multichannel active noise control systems using variable step size. Signal Processing 2008; 88: 2019–2029.

16. Zhang M, Lan H and Ser W. A robust online secondary path modeling method with auxiliary noise power scheduling strategy and norm constraint manipulation. IEEE Trans Speech Audio Process 2003; 11: 45–53.

17. Lan H, Zhang M and Ser W. An active noise control system using online secondary path modeling with reduced auxiliary noise. IEEE Signal Process Lett 2002; 9: 16–18.

18. George NV and Panda G. A particle-swarm-optimization-based decentralized nonlinear active noise control system. IEEE Trans Instrum Meas 2012; 61: 3378–3386.

19. Russo F and Sicuranza GL. Accuracy and performance evaluation in the genetic optimization of nonlinear systems for active noise control. IEEE Trans Instrum Meas 2007; 56: 1443–1450.

20. Russo F and Sicuranza GL. Genetic optimization in nonlinear systems for active noise control: accuracy and performance evaluation. In: IEEE instrumentation and measurement technology conference proceedings, Sorrento, Italy, 24–27 April 2006, pp.1512–1517. Piscataway, NJ: IEEE.

21. Chang CY and Chen DR. Active noise cancellation without secondary path identification by using an adaptive genetic algorithm. IEEE Trans Instrum Meas 2010; 59: 2315–2327.

22. Kennedy J and Eberhart RC. Population structure and particle swarm performance. In: IEEE and congress on evolutionary computation, Honolulu, HI, USA, 12–17 May 2002, pp.1671–1676. Piscataway, NJ: IEEE.

23. Kennedy J and Eberhart RC. Particle swarm optimization. In: IEEE international conference on neural networks, Perth, WA, Australia, 27 November–1 December 1995, pp.1942–1948. Piscataway, NJ: IEEE.

24. Rout NK, Das DP and Panda G. Particle swarm optimization based active noise control algorithm without secondary path identification. IEEE Trans Instrum Meas 2012; 61: 554–563.

25. Modares H, Ahmadyfard A and Hadadzaria M. A PSO approach for non-linear active noise cancellation. In: The 6th WSEAS international conference simulation, modeling and optimization, Lisbon, Portugal, 22–24 September 2006, pp.492–497. Stevens Point, Wisconsin: WSEAS.

26. Xia L, Hui G and Jinfeng L. Adaptive noise canceller based on PSO algorithm. In: IEEE international conference on automation and logistics, Qingdao, China, 1–3 September 2008, pp.1759–1762. Piscataway, NJ: IEEE.

27. George NV and Panda G. A robust evolutionary feedforward active noise control system using Wilcoxon norm and particle swarm optimization algorithm. Expert Syst Appl 2012; 39: 7574–7580.

28. Rout NK, Das DP and Panda G. Performance evaluation of particle swarm optimization based active noise control algorithm. In: International conference on swarm, evolutionary, and memetic computing, Chennai, India, 16–18 December 2010, pp.531–536. Berlin, Germany: Springer Verlag.

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

30. Sun J, Feng B and Xu W. Particle swarm optimization with particles having quantum behavior. In: IEEE congress on evolutionary computation, Portland, OR, USA, 19–23 June 2004, pp.325–331. Piscataway, NJ: IEEE.

31. Sun J, Xu W and Feng B. A global search strategy of quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Inf Sci 2015; 294: 408–422.

32. Li Y, Bai X, Jiao L, et al. Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Appl Soft Comput 2017; 56: 345–356.

33. Kurczyk S and Pawelczyk M. Active noise control without secondary path modelling – varying-delay LMS approach. In: The 19th international conference on methods and models in automation and robotics, Miedzyzdroje, Poland, 2–5 September 2014, pp.134–139. Piscataway, NJ: IEEE.

34. Sun J, Wu XJ, Palade V, et al. Convergence analysis and improvements of quantum-behaved particle swarm optimization. Inf Sci 2012; 193: 81–103.

35. Frans VDB. An analysis of particle swarm optimizers. PhD Dissertation, University of Pretoria, South Africa, 2002.