Journal of Sound and Vibration (2001) **240**(4), 647–665
doi:10.1006/jsvi.2000.3256, available online at http://www.idealibrary.com on IDEAL®

AN ALGORITHM FOR ACTIVE CONTROL OF TRANSFORMER NOISE WITH ON-LINE CANCELLATION PATH MODELLING BASED ON THE PERTURBATION METHOD

X. QIU AND C. H. HANSEN

Department of Mechanical Engineering, The University of Adelaide, South Australia 5005, *Australia.*
E-mail: *xjqiu@mecheng.adelaide.edu.au*

(*Received* 22 *March* 2000, *and in final form* 7 *July* 2000)

Previous work has demonstrated the potential for the active control of transformer noise using a combination of acoustic and vibration actuators and the filtered-x LMS algorithm (FXLMS), the latter being implemented to make the system adaptive. For a large electrical transformer, the number of actuators and error sensors needed to achieve a significant global noise reduction can be up to hundreds, and this makes the convergence of the FXLMS algorithm very slow. The memory requirement for the cancellation path transfer functions (CPTF) and the computation load required to pre-filter the reference signal by all the CPTFs are relatively large. On the other hand, not only the transformer noise but also the CPTF varies considerably from day to day, which makes on-line CPTF modelling very necessary. A new adaptive algorithm based on waveform synthesis is proposed, and the perturbation method is used to obtain the CPTF on-line. A comparison of the performance of the proposed algorithm with the FXLMS algorithm and the H-TAG algorithm shows the feasibility of the algorithm for the control of a slowly time-varying system with just a few fixed frequency components.

1. INTRODUCTION

Noise from large electrical transformers is characterized by single-frequency components at 2, 4, 6 and 8 times the AC line frequency. When transformers are located close to residential communities, the characteristic low-frequency humming noise is often a cause of widespread complaints. Previous work has demonstrated the potential of active control of transformer noise using a combination of sound sources and vibration actuators and the filtered-x LMS type of algorithm (FXLMS) [1–4].

The FXLMS algorithm is a simple extension of LMS algorithm for active noise and vibration control (ANVC) systems, which takes account of the presence of the cancellation path transfer functions (CPTF) between the output of the adaptive control filters and the error sensors. It has been shown both theoretically and experimentally that fast and precise estimation of the CPTF is one of the most important factors for fast convergence of an ANVC system using the FXLMS algorithm [5–9]. Usually, the CPTF are estimated on-line to track the time-varying parameters in the physical system.

A variety of methods have been tried for on-line CPTF modelling [7–11], which can be roughly divided into two distinct approaches. The first approach is based on the injection of auxiliary uncorrelated random noise into the system, and the level of the auxiliary noise can

0022-460X/01/090647 + 19 $35.00/0

be 20–30 dB lower than the unwanted disturbance. By contrast, the second approach, called the overall modelling algorithm, is based on the use of the very correlated control output to perform CPTF modelling by employing an extended least-squares approach. An extensive comparison of these two types of on-line CPTF modelling approach shows that the first approach is characterized by a faster convergence speed and a larger stability margin. The advantage of the second approach is that a very short adaptive filter can be used to model the CPTF for a disturbance consisting of multiple sinusoids.

Although the transformer noise field can be treated as a stationary sound field for a short time period, it varies with different loading and weather conditions. Furthermore, not only the transformer noise but also the CPTF varies considerably from day to day, which makes on-line CPTF modelling very necessary. Unfortunately, the injection of auxiliary uncorrelated random noise into the system for CPTF modelling is not suitable if vibration actuators are used in the system. For example, it is quite likely that, in a transfer function between an actuator and an error sensor, the amplitude at frequencies of interest is 20–30 dB lower than the nearby structural resonance frequencies. In this case, the modelling signal level of random noise has to be much lower than the primary disturbance, and this makes the modelling of the CPTF quite slow or impossible.

For disturbances like transformer noise, which consist of multiple sinusoids, there is usually no need to be concerned about the causality or delay associated with the cancellation path. Overall modelling might be suitable in this situation. However, overall modelling does not have a unique solution, and sometimes the model can be completely wrong, yet still have minimal total estimation error. The right CPTF are obtained provided that only one of the CPTF and primary sound field changes at one time and the starting estimate of the CPTF should be close to the correct value. For the transformer noise situation, it is likely that the CPTF and primary sound field change simultaneously, for example, as a result of changes in temperature.

For a large electrical transformer, the number of actuators and error sensors needed for a significant global noise reduction can be up to hundreds, and the cross-coupling among channels makes the on-line modelling of the CPTF even more difficult. The memory requirement for the CPTF and the computation load required to pre-filter the reference signal by all these CPTFs are also relatively large. All these considerations illustrate the difficulties of directly applying the time-domain FXLMS algorithm to the active control of power transformer noise. Although some fast algorithms have been proposed to increase the computation efficiency and reduce the memory requirement of the multi-channel FXLMS algorithm [12–14], yet they usually require a knowledge of the CPTFs *a priori*, so they will not be discussed here.

A number of other algorithms have been proposed especially for the active control of periodic disturbances. For example, the delayed-X harmonic synthesizer algorithm (DXHS) requires only the error signal and the information on the fundamental frequency of the harmonics of the primary noise and achieves control by adjusting the amplitude and phase for each harmonic component of a synthesized signal [15, 16]. The delayed-X LMS algorithm (DXLMS) is a simplification of the FXLMS algorithm for a long duct or narrowband noise cancellation application [17]. The algorithm reduces the computational load of the FXLMS algorithm based on the hypothesis that the CPTF modelling for the FXLMS algorithm does not have to be accurate and can be represented by a delay in such cases. Unfortunately, they both use an overall modelling type of algorithm for on-line delay estimation.

Another algorithm specially developed for attenuating tonal noise was demonstrated to be robust and to outperform the FXLMS algorithm in simplicity and convergence speed [18]. It adjusts the weights of the in-phase and the in-quadrature components of the reference signal respectively. For a disturbance consisting of multiple sinusoids, a bank of band-pass filters are needed to separate each tonal component. Unfortunately, the on-line cancellation path phase shift estimator belongs to the overall modelling algorithm type and this is associated with the problems discussed previously.

To avoid the inefficient, long adaptive FIR filter associated with the FXLMS algorithm, a linear combination of fixed stable IIR filters can be used to attenuate periodic noise [6, 9, 19]. A series of orthogonal IIR-based filter pairs are designed first, then the reference signal is filtered by these IIR filters into in-phase and in-quadrature components. The attenuation is achieved by adjusting the weights of these in-phase and in-quadrature components in a similar way to the DXHS algorithm [15, 16] and the algorithm proposed in reference [18]. The algorithm can save much computational power while avoiding instability and non-linearity problems that are usually associated with the IIR adaptive filters. However, no on-line CPTF modelling algorithm is provided with the algorithm.

There are also some adaptive ANVC algorithms that try to avoid CPTF modelling by using artificial neural networks, a genetic search method or Newton's method [9, 20, 21, 22]. For example, the Higher-harmonic Time-Average Gradient descent algorithm (H-TAG) does not require a measure of the CPTF for the system [20, 21]. If the system dynamics change during the control process, the H-TAG algorithm will simply adapt to the new optimal control configuration. The H-TAG algorithm uses a perturbation method to determine the gradient and the second derivative estimate of the performance surface. It has been implemented successfully on an actual laboratory model that has eight inputs and four outputs. However, it is very slow in adaptation and can only be used as an alternative for some slowly time-varying systems that only have a few parameters to be adjusted.

A new adaptive algorithm based on waveform synthesis, which is especially useful for the active control of transformer noise, is proposed here. The cost function of the proposed algorithm is the amplitude of the error signal rather than the instantaneous time-domain error signal, and the perturbation method is used to obtain the needed CPTF information on-line. Some initial results obtained with the algorithm were presented in reference [23].

2. DESCRIPTION OF THE ALGORITHM

2.1. SISO SYSTEM FOR A SINGLE SINUSOID

Figure 1 shows a block diagram of a single-channel active control system for a sinusoid using the proposed algorithm. The primary noise source is represented by $\sin(\omega n + \varphi_0)$, which has stable frequency and phase, as in the case of transformer noise. $P(Z)$ is the primary path transfer function (structural/acoustic system) between the primary noise $p(n)$ at the error sensor and the primary noise source. $s(n)$ is the actual control signal at the position of the error sensor, and is obtained by filtering the output of the controller, $y(n)$, with the physical cancellation path transfer function $S(Z)$. $v(n)$ represents other unwanted additional noise (such as wind noise) picked up by the error sensor. The error signal $e(n)$ comes from the sum of $p(n)$, $s(n)$ and $v(n)$.

As shown in Figure 1, a sine generator and a cosine generator are used to produce the control output $y(n)$:

$$y(n) = A_{cs}(n)\sin(\omega n) + A_{cc}(n)\cos(\omega n), \tag{1}$$

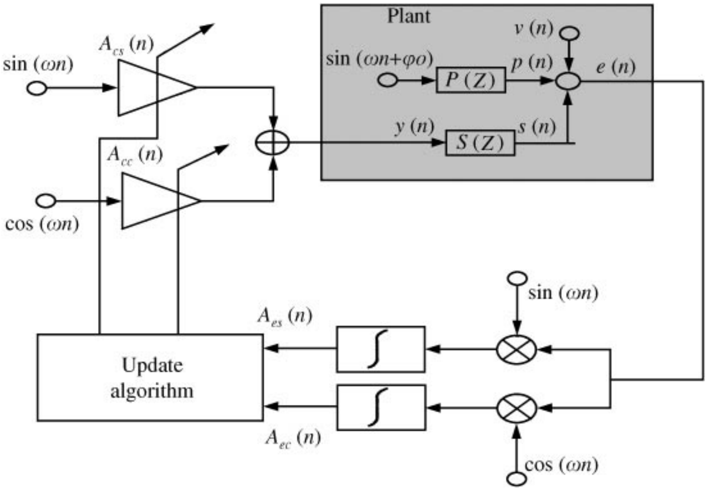

Figure 1. Block diagram of a SISO active control system using the proposed algorithm.

where $A_{cs}(n)$ is the weight of the sine component and $A_{cc}(n)$ is the weight of the cosine component for the control output. The same sine and cosine generators are also used to obtain the sine component $A_{es}(n)$ and cosine component $A_{ec}(n)$ of the error signal $e(n)$,

$$A_{es}(n) = \frac{2}{N}\sum_{i=1}^{N} \sin(\omega(n - N + i))e(n - N + i), \tag{2a}$$

$$A_{ec}(n) = \frac{2}{N}\sum_{i=1}^{N} \cos(\omega(n - N + i))e(n - N + i), \tag{2b}$$

where N is the number of samples within one or a certain number of disturbance periods. If the signal frequency is f_0 and the sampling frequency of the system is f_s, the angular frequency $\omega = 2\pi f_0/f_s$.

The cost function for the proposed algorithm is the expectation of the sum of the squared amplitudes of each component:

$$J = E(A_{es}^2(n) + A_{ec}^2(n)). \tag{3}$$

By using the orthogonality of the sine and cosine signals, and sine and cosine components of the error signal can be expressed as

$$A_{es}(n) = A_{ps}(n) + A_{cs}(n)C_{ss}(n) + A_{cc}(n)C_{cs}(n) + A_{vs}(n), \tag{4a}$$

$$A_{ec}(n) = A_{pc}(n) + A_{cs}(n)C_{sc}(n) + A_{cc}(n)C_{cc}(n) + A_{vc}(n), \tag{4b}$$

where A_{ps} and A_{pc} are the sine and cosine components of the primary noise $p(n)$, while A_{vs} and A_{vc} are the sine and cosine components of the unwanted additional noise $v(n)$. C_{ss}, C_{sc}, C_{cs} and C_{cc} are the corresponding components in the cancellation path transfer function $S(Z)$, which can be obtained by the perturbation method as will be described later.

By using the gradient search approach to minimize the cost function just in the LMS algorithm (there are no explicit filters involved here although the waveform synthesizer can be treated as a series of two tap FIR filters using two orthogonal reference signals), the recursive weight update equations so obtained are

$$A_{cs}(n+1) = A_{cs}(n) - 2\mu(A_{es}(n)C_{ss}(n) + A_{ec}(n)C_{sc}(n)), \tag{5a}$$

$$A_{cc}(n+1) = A_{cc}(n) - 2\mu(A_{es}(n)C_{cs}(n) + A_{ec}(n)C_{cc}(n)), \tag{5b}$$

where μ is the step size of the update.

2.2. MIMO SYSTEM FOR MULTIPLE SINUSOIDS

A MIMO system for a sinusoid is considered first where the sine and cosine components for all the error signals are obtained in the same way as in equation (2). The cost function for the system is defined as

$$J = E\left(\sum_{m=1}^{M} (A_{esm}^2(n) + A_{ecm}^2(n))\right), \tag{6}$$

where M is the number of error signals (the number of inputs to the system). Similarly, as in the SISO system, the mth error signal can be expressed as

$$A_{esm}(n) = A_{psm}(n) + A_{vsm}(n) + \sum_{l=1}^{L} (A_{csl}(n)C_{sslm}(n) + A_{ccl}(n)C_{cslm}(n)), \tag{7a}$$

$$A_{ecm}(n) = A_{pcm}(n) + A_{vcm}(n) + \sum_{l=1}^{L} (A_{csl}(n)C_{sclm}(n) + A_{ccl}(n)C_{cclm}(n)), \tag{7b}$$

where L is the number of control outputs. A_{psm} and A_{pcm} are the sine and cosine components of the primary noise $p(n)$ at the mth error sensor, while A_{vsm} and A_{vcm} are the sine and cosine components of the unwanted additional noise $v(n)$ at the mth error sensor. C_{sslm}, C_{sclm}, C_{cslm} and C_{cclm} are the corresponding components in the cancellation path transfer functions from the lth control output to the mth error sensor. For example, a sine signal can be fed into the cancellation path from the lth control output, and the input from the mth error sensor is decomposed into a sine component and a cosine component. The ratio of the amplitude of the sine component of the mth error sensor signal to the amplitude of the sine signal from the lth control output is defined as C_{sslm} while the ratio of the amplitude of the cosine component of the mth error sensor signal to the amplitude of the sine signal from the lth control output is defined as C_{sclm}. Using the gradient of the cost function in equation (6), the obtained recursive weight update equations are

$$A_{csl}(n+1) = A_{csl}(n) - 2\mu \sum_{m=1}^{M} (A_{esm}(n)C_{sslm}(n) + A_{ecm}(n)C_{sclm}(n)), \tag{8a}$$

$$A_{ccl}(n+1) = A_{ccl}(n) - 2\mu \sum_{m=1}^{M} (A_{esm}(n)C_{cslm}(n) + A_{ecm}(n)C_{cclm}(n)). \tag{8b}$$

When the disturbance consists of multiple sinusoids, each frequency component is processed in the same way as the others in parallel due to the orthogonal property of the sine and cosine signals at different frequencies. However, more sine and cosine signal generators with different frequencies are engaged to produce the control output and to obtain the sine and cosine components of the error signal.

2.3. ON-LINE CPTF MODELLING BY PERTURBATION METHOD

As the aim of the proposed algorithm is to control a few sinusoids, it is not necessary to obtain the complete cancellation path transfer functions. What the proposed algorithm needs for a proper update is just a matrix. For example, for a SISO system with a sinusoid, it is a 2 × 2 matrix:

$$C = \begin{bmatrix} C_{ss} & C_{sc} \\ C_{cs} & C_{cc} \end{bmatrix}.$$

If only one of the CPTF and primary sound field change at a time, the overall modelling approach, which employs an extended least-squares method, can be used to obtain the matrix. However, the overall modelling approach cannot work if both the primary disturbance and the CPTF are changing simultaneously, even for a very slowly varying system. The perturbation method is applied here to estimate the CPTF matrix for a very slowly varying system, independent of whether the primary disturbance and CPTF change simultaneously or not.

Because the system is assumed to be changing very slowly, a perturbation of a certain level can be added to the control output to obtain the transfer functions. Provided that the signal level of the perturbation is much larger than the self-change of the system at a certain time, a good estimation of the CPTF matrix is possible. For a SISO system, if a perturbation of ΔA_c is added on the amplitude of the sine component of the control output $A_{cs}(n)$ for a particular time, there will be a perturbation on the amplitude of the error signal. The obtained perturbation of the sine and cosine components of the error signal are

$$\Delta A_{es}(n) = \Delta A_c(n) C_{ss}(n) + \delta_1, \qquad \Delta A_{ec}(n) = \Delta A_c(n) C_{sc}(n) + \delta_1, \tag{9a, b}$$

where δ_1 is the perturbation generated by the self-change of the system at this particular period of time, and it should be much smaller than the perturbation generated by ΔA_c. From equation (9), the following elements of the cancellation path matrix can be obtained:

$$C_{ss}(n) \approx \frac{\Delta A_{es}(n)}{\Delta A_c(n)}, \qquad C_{sc}(n) \approx \frac{\Delta A_{ec}(n)}{\Delta A_c(n)}. \tag{10a, b}$$

Similarly, C_{cs} and C_{cc} can be obtained by imposing a perturbation of ΔA_c on the amplitude of the cosine component of the control output $A_{cc}(n)$ for a particular period of time. For a MIMO system with L control outputs and M error inputs to control a sinusoid, the required $2M \times 2L$ CPTF matrix is

$$C = \begin{bmatrix} C_{ss11} & C_{sc11} & \cdots & \cdots & C_{ssL1} & C_{scL1} \\ C_{cs11} & C_{cc11} & \cdots & \cdots & C_{csL1} & C_{ccL1} \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ C_{ss1M} & C_{sc1M} & \cdots & \cdots & C_{ssLM} & C_{scLM} \\ C_{cs1M} & C_{cc1M} & \cdots & \cdots & C_{csLM} & C_{ccLM} \end{bmatrix}.$$

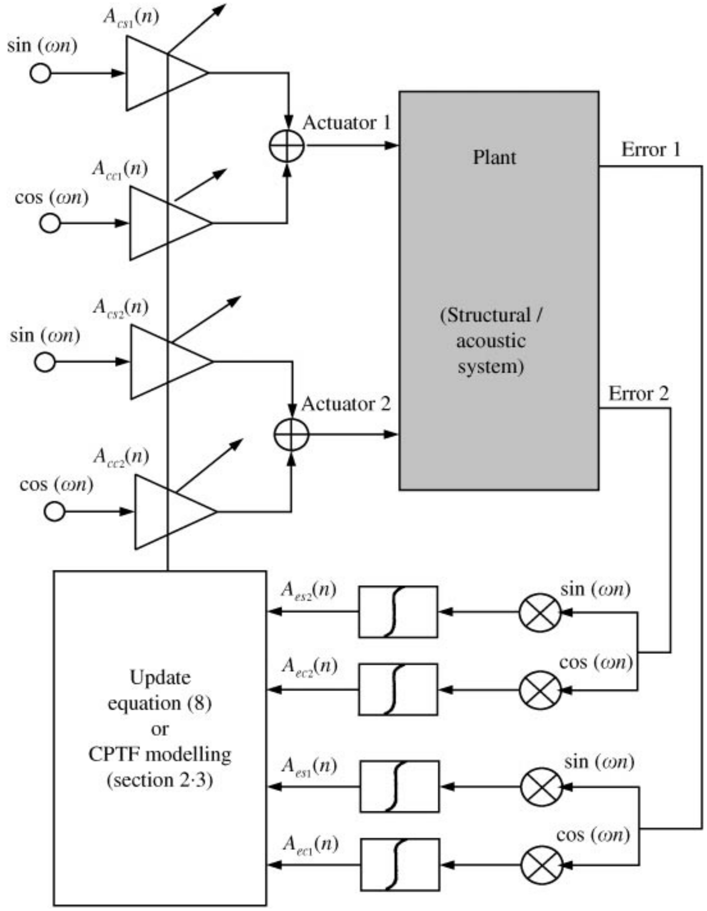

Figure 2. Block diagram of a MIMO active control system using the proposed algorithm (2 input, 2 output for a sinusoid).

To obtain all the elements of the CPTF matrix, the perturbation should be applied to all of the control output amplitudes in turn. When one control output is being perturbed, the elements of the CPTF matrix from one control output to all M error inputs can be obtained at the same time. Altogether $2L$ perturbations will be applied to the system to obtain the full matrix. For a MIMO system to control K sinusoids, there are K such $2M \times 2L$ CPTF matrices. Fortunately, due to the orthogonality property of the sine signal, the CPTF matrix for different frequencies can be estimated in parallel. Figure 2 shows the total structure of the proposed algorithm. Note that only one sinusoid is shown in the figure, as more sinusoids can be treated in parallel in the same way. The control weights of all control channels are updated at the same time. However, the control weight updating stops when perturbation is injected into the system to do the cancellation path modelling. In other words, the control weight update and the CPTF modelling run at separate times.

2.4. THE CONVERGENCE CONDITION FOR A SISO SYSTEM

The convergence condition of the proposed algorithm for a SISO system is derived below. For simplicity, A_{ps}, A_{pc}, A_{vs} and A_{vc} are not included in the following derivation.

Substituting equation (4) into equation (5) gives

$$A_{cs}(n+1) = A_{cs}(n) - 2\mu((A_{cs}(n)C_{ss} + A_{cc}(n)C_{cs})C_{ss} + (A_{cs}(n)C_{sc} + A_{cc}(n)C_{cc})C_{sc}), \quad (11a)$$

$$A_{cc}(n+1) = A_{cc}(n) - 2\mu((A_{cs}(n)C_{ss} + A_{cc}(n)C_{cs})C_{cs} + (A_{cs}(n)C_{sc} + A_{cc}(n)C_{cc})C_{cc}). \quad (11b)$$

For a cancellation path transfer function with a frequency response of the $A_{cp}e^{j\theta}$, the elements of the CPTF matrix are

$$A_{cp}\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \quad (12)$$

Therefore,

$$C_{sc} = -C_{cs}, \quad C_{ss} = C_{cc}, \qquad C_{ss}^2 + C_{sc}^2 = A_{cp}^2, \quad C_{cs}^2 + C_{cc}^2 = A_{cp}^2. \quad (13)$$

By using the properties of the CPTF matrix, equation (11) becomes

$$A_{cs}(n+1) = (1 - 2\mu A_{cp}^2)A_{cs}(n), \qquad A_{cc}(n+1) = (1 - 2\mu A_{cp}^2)A_{cc}(n). \quad (14a, b)$$

Thus, the convergence condition is

$$0 < \mu < \frac{1}{A_{cp}^2}. \quad (15)$$

As shown in the above equation, the step size is inversely proportional to the squared amplitude of the cancellation path transfer function at the frequency of interest and unlike a number of other algorithms, it has no explicit relationship to the phase of the cancellation path transfer function. However, the update algorithm does need the phase information for updating, as shown in the update of equations (5) and (8). This is the same as the FXLMS algorithm.

2.5. THE DIFFERENCE BETWEEN THE PROPOSED ALGORITHM AND THE OTHERS

Although a number of the algorithms reported previously have used the orthogonal property of the sine signal to control periodic disturbances [8, 9, 15, 16, 18, 20, 21], the orthogonal decomposition is often only carried out on the reference signal of the system. The algorithm proposed in this paper decomposes the error signal as well, which makes the update algorithm converge faster and in a more stable manner. For example, the cancellation path transfer function at a particular frequency from a control output to an error sensor becomes a simple 2 × 2 matrix, and the sine and cosine components of the error signal can be controlled separately.

One big difference between the proposed algorithm and the others is that the amplitudes of the error signal at the frequencies of interest are used as the cost function instead of the instantaneous time-domain error signal. The benefit of this is that the algorithm can be set-up to control only the frequencies of interest, so it has a high rejection of disturbances and system characteristics outside the individual frequencies being cancelled. In fact, the algorithm proposed in this paper has the same characteristics as a frequency domain adaptive control system [8, 9, 24]. Instead of using a FFT to obtain the whole spectrum, which is time consuming, the proposed algorithm just obtains the amplitudes of the sine and cosine components of the error signal at the frequencies of interest by integrating over a few disturbance periods.

Another difference between the proposed algorithm and some of the others is that the perturbation method is used to obtain the effect of the cancellation path. Just like a human being, the proposed algorithm changes the amplitude of the control output a little bit to observe the response of the system and then adjusts the control output. No other unwanted noise is injected into the system, and it is also not necessary to assume that only one of the cancellation path or primary disturbance can be changed at a time. However, the requirement of the proposed algorithm is that the system must not change rapidly.

3. COMPUTER SIMULATIONS

3.1. IMPLEMENTATION CONSIDERATIONS

The waveform synthesis method usually needs a synchronized signal so that the sampling frequency of the system is an integer multiple of the disturbance frequency. However, the condition cannot be met easily in some practical situations. When the sampling frequency of the system is not an integer multiple of the disturbance frequency, the average number N for the integration can be decided by

$$N = Int\left(\frac{f_s}{f_0} \times K\right), \tag{16}$$

where $Int(\ )$ means the nearest integer to a real number. K is the number of periods. The estimated amplitude is within $1/N$ percent of the true value. The estimation error generated in this way has a large effect on the system; actually, it sets the lowest limit of the signal level of perturbation for cancellation path modelling. The error of the estimation can be reduced by adding a moving average process below

$$A_{esa}(n+1) = (1-\lambda)A_{esa}(n) + \lambda A_{es}(n), \tag{17a}$$

$$A_{eca}(n+1) = (1-\lambda)A_{eca}(n) + \lambda A_{ec}(n), \tag{17b}$$

where λ is the forgetting factor, $A_{esa}(n)$ and $A_{eca}(n)$ are the averages of the $A_{es}(n)$ and $A_{ec}(n)$ respectively. $\lambda = 1$ means that no old data are included, with no averaging being involved. For the case described above, letting $N = 22$ and $\lambda = 1$, the maximum estimated error is 3·98% of the true value, while $\lambda = 0{\cdot}1$ gives a maximum estimated error of 0·74% and $\lambda = 0{\cdot}01$ gives an estimated error of 0·08%. The larger the K or the smaller the λ, the smaller the estimation error that can be obtained. However, if longer data samples are used to do the estimation, the tracking speed of the adaptive system becomes slower. It can also be seen from equation (16) that a higher sampling rate can reduce the estimated error, being equivalent to increasing N. However, the cost is an increase in the computation load.

Figure 3 shows the effects of the leakage from other frequency components on the estimation. It can be seen that the leakage has a large influence on the estimation. One way to reduce this effect is to increase the number of averages by increasing N or reducing λ. If the disturbance is dominated by a fundamental tone and its higher harmonics, N should be decided by the fundamental frequency. Fortunately, there is a minimum near each higher harmonic frequency in this case, as shown in Figure 3. For an adaptive control system, which is associated with a decreasing amplitude of the disturbance sinusoids, the relative influence of the nearby frequency components will increase as the controller adapts,

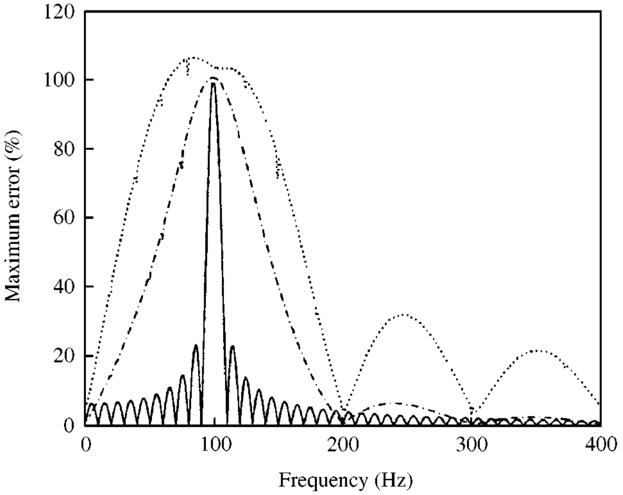

Figure 3. The maximum error, introduced by the leakage of other frequency components in the estimation of the amplitude of a sine wave by using the proposed method: ·····, $N = 22$, $\lambda = 1$; –·–·, $N = 22$, $\lambda = 0{\cdot}1$; ——, $N = 223$, $\lambda = 1$, $f_s = 2232{\cdot}1$ Hz, $f_0 = 100$ Hz.

therefore a longer N may be needed to increase the accuracy of the estimation when the residual error becomes small.

The next implementation issue to be addressed is the levels of the perturbations, ΔA_{es} and ΔA_{ec}. Because each perturbation presents a disturbance to the system, its level should be set as small as possible. However, it has a lower limit δ given by

$$\delta = \delta_1 + \delta_2 + \delta_3 + \delta_0, \qquad (18)$$

where δ_1 is the perturbation generated by the self-change of the system within the perturbation time, δ_2 is the estimation error due to the sampling frequency of the system not being an integer multiple of the disturbance frequency, δ_3 is the estimation error due to the leakage of other frequency components and δ_0 is all other unwanted background noise due to the hardware and physical system.

For a slowly varying system, δ_1 is small, and if other unwanted noise δ_0 is not considered, then the perturbation level is mainly decided by δ_2 and δ_3. Considering the case described above, if the disturbance is dominated by 100 Hz and its higher harmonics and the sampling frequency f_s is 2232·1 Hz, the maximum error of the estimation for $N = 22$ and $\lambda = 0{\cdot}1$ is 0·74% of the true value. So the perturbation level can be as large as 10% of the amplitude of the error signal when the system starts so that the δ_2 type of error can be neglected. However, it is also important to set a minimum perturbation to avoid the δ_3 type of error, especially when the error signal amplitude has been significantly reduced.

The background noise δ_0 mainly comes from the electric circuits and the environmental noise sensed by the sensors. Normally, a large background noise increases the perturbation level required to obtain the cancellation path information, resulting in a limited noise reduction of the ANVC system. Moreover, a large background noise level reduces the stability margin of the adaptive control algorithm and a smaller convergence coefficient might have to be used, resulting in a slow tracking system.

While the method described above provides a theoretical bound on the perturbation level, it is not practical, because the perturbation level put on the waveform synthesizer,

ΔA_c, cannot be determined without knowing the CPTF matrix as shown in equation (10). In practice, the ΔA_c can be determined experimentally by trial and error and then linked with the error signal amplitude. An example to determine ΔA_c will be given in the following section.

3.2. THE COMPUTATION LOAD AND MEMORY REQUIREMENT FOR IMPLEMENTING THE ALGORITHM

The computation load and memory requirement for implementing the algorithm will be considered for a 10 error input and 10 control output system, which uses the proposed algorithm to control four frequency components. The computation load of the algorithm is divided into two parts: the real-time part (the job must finish before the arrival of the next sample) and the background part. In the real-time part, the sine and cosine amplitudes of the 10 error signal for all the four frequency components were estimated first, which needs $4 \times 10 \times 4 = 160$ multiply/accumulate (MAC) operations. Another job in the real-time part is to synthesize the control output, which needs $2 \times 10 \times 4 = 80$ MAC. So the entire calculation load for the real-time part is 240 MAC. The control weights update and the cancellation path modelling were done in the background part, which executed every certain number of the samples. The number of the samples depends on the delay of the system (100 samples here) and the number of samples used to average the error amplitude. A normal DSP can handle all these processing.

For a 10 error input and 10 control output control system using the proposed algorithm, 400 floating point numbers are needed to express the cancellation path transfer function for one frequency component. For four frequency components, 1600 floating point numbers are needed to hold the cancellation path information. The other memory requirements are much less than this. However, for a controller using the FXLMS algorithm with 32 tap control filters and 32 tap FIR filters for cancellation path modelling, the memory is needs is at least $32 \times 10 \times 10 \times 2 = 6400$ floating point numbers, just for cancellation path modelling. So the proposed algorithm requires much less memory than a typical implementation of the FXLMS algorithm.

3.3. SIMULATION RESULTS

The proposed algorithm was simulated on a simply SISO system. The sampling frequency f_s was 2232·1 Hz, the disturbance frequency f_0 was 100 Hz, and the amplitude of the primary disturbance was 1·0 with an initial phase of 28·65°. The frequency response of the cancellation path transfer function was $2{\cdot}0e^{2{\cdot}0j}$, which is about a 114·6° phase shift. The level of the perturbation for CPTF modelling was determined by

$$\Delta A_c(n) = \eta \frac{A_{ea}(n)}{C_{max}} + \delta_{min}, \tag{19}$$

where η is a factor used to determine the level of the perturbation, and is set at 10% in the simulations. C_{max} (set at 4·0 in the simulations) is a number larger than the maximum value of the elements of the CPTF matrix and normally should be estimated before the algorithm is set running. $A_{ea}(n)$ is the squareroot of the sum of the $A^2_{esa}(n)$ and $A^2_{eca}(n)$. δ_{min} is the lowest limit for the perturbation level, which is set at 0·001 in the simulations.

Before the proposed algorithm starts, a delay time for estimating the sine and cosine components of the error signal should be determined. It depends on the delay of the

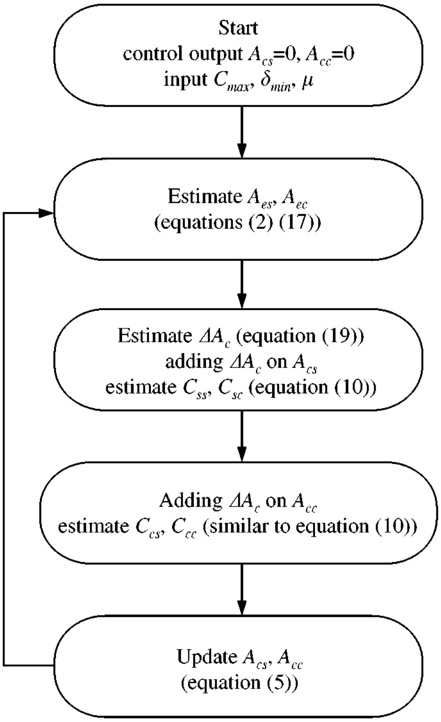
Start
control output A_{cs}=0, A_{cc}=0
input C_{max}, δ_{min}, μ
Estimate A_{es}, A_{ec}
(equations (2) (17))
Estimate ΔA_c (equation (19))
adding ΔA_c on A_{cs}
estimate C_{ss}, C_{sc} (equation (10))
Adding ΔA_c on A_{cc}
estimate C_{cs}, C_{cc} (similar to equation (10))
Update A_{cs}, A_{cc}
(equation (5))

Figure 4. The implementation procedure of the proposed algorithm.

cancellation path as well as N and λ. In this simulation, the time is set at 100 samples, which is enough for $N = 22$, $\lambda = 0{\cdot}1$ and a system delay of at least 70 samples.

Figure 4 shows the implementation procedure of the proposed algorithm. For example, for this simulation, in the first 100 samples, no control signal is injected into the system. $A_{esa}(n)$ and $A_{eca}(n)$ are obtained for later use. In the second 100 samples, the perturbation level ΔA_c is estimated, then a perturbation sine signal is injected into the system. At the end of this period, C_{ss} and C_{sc} can be obtained. ΔA_{es} and ΔA_{es} are obtained by subtracting the previously obtained $A_{esa}(n)$ and $A_{eca}(n)$ in the first 100 samples from the present ones. In the third 100 samples, just a perturbation cosine signal is injected into the system, C_{cs} and C_{cc} are obtained in the same way as C_{ss} and C_{sc}. From the 301th sample, the update algorithm starts. If the system changes very slowly, the update time can be made longer until the error achieves a minimum value. In this simulation, the update time is set at 200 samples. From the 501th sample, the control signal update stops and the system starts to measure the sine and cosine components of the error signal again just as in the first 100 samples. Then from 610th sample, the system starts to inject the perturbation sine signal, etc. and so on.

Figure 5 shows the first 2500 residual error amplitudes ($A_{ea}(n)$ in dB ref. $1{\cdot}0$) following the system starting. The results for $\lambda = 1$ are also shown in the figure for comparison. The update size μ in equation (6) was set at $0{\cdot}01$ in the simulations. The residual noise of the system is $0{\cdot}0001$, which corresponds to -80 dB in the figure. As shown in the figure, the residual error amplitude has a period of 500 samples. Within each 500 samples, the residual error amplitude remains unchanged in the first 100 samples and a perturbation is injected into the system at the second and third 100 samples; from the start of the fourth 100 samples, the control filter begins to update and stops updating when the 500th sample

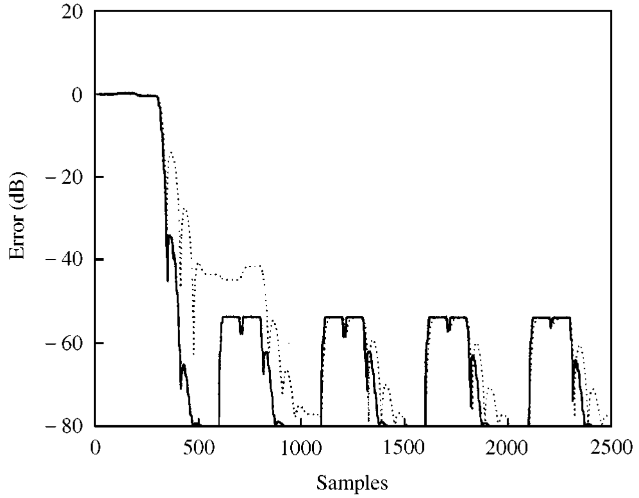

Figure 5. Time history of the residual error amplitude after applying the proposed algorithm: ·····, $N = 22$, $\lambda = 0{\cdot}1$; ——, $N = 22$, $\lambda = 1$, $\mu = 0{\cdot}01$, $\delta_{min} = 0{\cdot}001$.

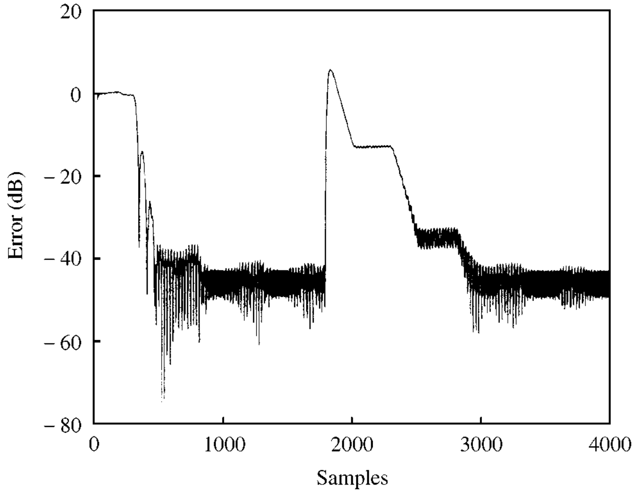

Figure 6. Time history of the residual error amplitude after applying the proposed algorithm with a second harmonic in the system: $N = 22$, $\lambda = 0{\cdot}1$, $\mu = 0{\cdot}01$. $\delta_{min} = 0{\cdot}001$, a sudden change occurs at sample 1800.

arrives. So the rate of CPTF modelling is once per 500 samples, about once every 0·224 s for the 2232·1 Hz sampling rate. In other words, if the cancellation path or primary path has a large change within 0·224 s, the control system will not be able to work properly.

As shown in Figure 5, the level of the perturbation reaches its minimum at the second 500 period for $\lambda = 1$ because the residual error amplitude reduces very quickly. It is −54 dB in the figure instead of −60 dB (0·001) due to the gain of the cancellation path. This gives the lowest limit for the final attenuation because the perturbation turns on periodically to track the slowly changing cancellation path.

Figure 6 shows the effects of a second harmonic on the convergence of the proposed algorithm. In this simulation, a sine signal of twice the fundamental frequency and an amplitude of 1·0 is added to the primary noise as an interference. There are two effects associated with this higher harmonic, due to its leakage, mentioned above. First, the maximum attenuation becomes less, and the residual error reaches only about -43 dB instead of -80 dB without the interference of the higher harmonic. This is coincident with the maximum error introduced by the higher harmonic as shown in Figure 3. Second, when the estimated error amplitude is dominated by the leakage error, the perturbation level obtained from equation (19) may not be large enough because the original δ_{min} used in the first simulation is estimated without taking into account the leakage of the higher harmonic. Thus, the obtained CPTF matrix may be completely wrong, and the following three matrices show the results:

$$C_{true} = \begin{bmatrix} -0{\cdot}83 & 1{\cdot}82 \\ -1{\cdot}82 & -0{\cdot}83 \end{bmatrix}, \quad C_1 = \begin{bmatrix} -1{\cdot}06 & 1{\cdot}57 \\ -1{\cdot}83 & -0{\cdot}88 \end{bmatrix}, \quad C_3 = \begin{bmatrix} -12{\cdot}05 & 8{\cdot}58 \\ -3{\cdot}12 & -0{\cdot}35 \end{bmatrix},$$

where C_{true} is the true value, while C_1 is the estimated value during the first 500 samples and C_3 is the estimated value during the third 500 samples. As can be seen, the CPTF matrix estimated during the third 500 samples is completely wrong, yet the residual error does not increase. This may be because the residual error at this stage is already not correlated with the control output so the update algorithm acts by just automatically turning itself off.

To further investigate the performance of the proposed algorithm, a sudden change is made on both the primary noise and the cancellation path. Note that the proposed algorithm is not supposed to track sudden change, the purpose of the simulation here is just to show what will happen if a sudden change happens. The sudden change happens at sample 1800, when the latest CPTF modelling just finishes. The initial phase of the primary noise change from 28·6 to 171·8°, and the amplitude increases two times. The cancellation path transfer function changes from $2{\cdot}0e^{2{\cdot}0j}$ to $1{\cdot}0e^{4{\cdot}0j}$, a dramatic phase shift change of 114·6°. As can be seen in Figure 6, the proposed algorithm has about 200 samples of unpredictable time and then begins to do CPTF modelling and then control. Provided that the step factor of the update equation is not very large, the proposed algorithm is capable of tracking the slow changing of the system as well as not to cause a disaster when a sudden change happens.

Figure 7 shows the results of applying the H-TAG algorithm on the same case as in Figure 5. The main difference between applying the H-TAG algorithm and the proposed algorithm is that the H-TAG algorithm does not further process the residual error and the mean squared error (MSE) is used directly to update control filter weights by using Newton's method [20, 21]. However, in the algorithm proposed in this paper, the residual MSE is further decomposed into sine and cosine components and the gradient search method is used to update the amplitude of control output. The same level perturbation decided by equation (19) is used for both cases. The H-TAG algorithm needs three steps (no perturbation, positive perturbation, negative perturbation) to update one control filter weight. There are 100 samples in each step, so the residual error amplitude in Figure 7 has a period of 300 samples.

Comparing Figures 6 and 7 with Figure 5, it can be seen that both algorithms can successfully cancel the primary noise; however, the H-TAG algorithm is more sensitive to the accuracy of the estimation of the residual MSE. The algorithm proposed in this paper can work successfully with an average number of 22 samples, while the H-TAG algorithm needs 67 samples (the period of disturbance is 22·321 samples). To increase the accuracy of the estimation, more cycles of error signal should be included in the average or a higher sampling rate should be applied to the system [20, 21]. This means either a slow down of the adaptive speed or a dramatic increase in the computational load.

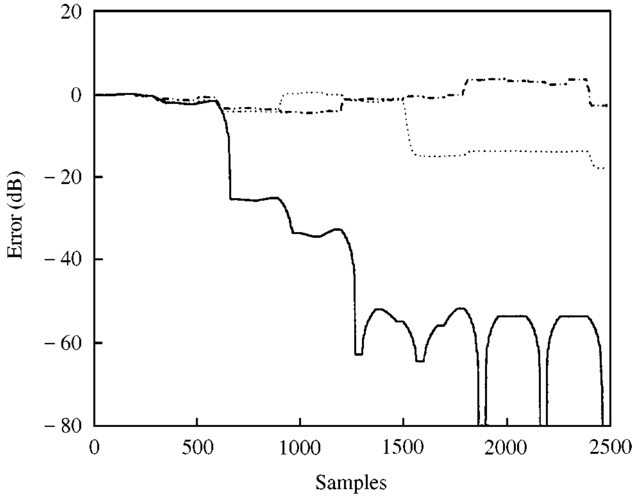

Figure 7. Time history of the residual error amplitude after applying the H-TAG algorithm: ······, $N = 22$, $\lambda = 0{\cdot}1$; –·–·, $N = 22$, $\lambda = 1$; ——, $N = 67$, $\delta_{min} = 0{\cdot}001$.

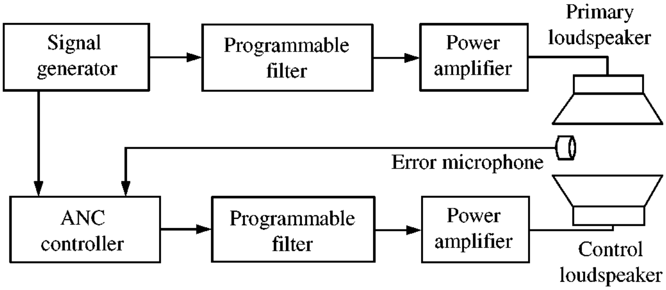

Figure 8. Block diagram of the experimental set-up.

4. EXPERIMENTAL WORK

4.1. EXPERIMENTAL SET-UP

The feasibility of the proposed algorithm was further verified in the experiments. Figure 8 shows the block diagram of the experimental set-up. Two loudspeakers were placed together, one was used to simulate the primary noise source, and the other was used as the control source. One signal form the signal generator was fed to a programmable filter, which was used to simulate the change of the primary sound field. The same signal from the signal generator was also fed to the ANC controller as the reference signal. The ANC controller processed the error signal from the error microphone located between the two loudspeakers, and then output the control signal into the control loudspeaker via a programmable filter, which was used to simulate the change of the cancellation path.

The proposed algorithm was realized on a SHARC EZ-KIT Lite board, which has an Analog Devices ADSP-21061 floating point DSP running at 40 MHz and An Analog Devices AD1847 16-bit Stereo SoundPort Codec providing 2 channel 16-bit A/D

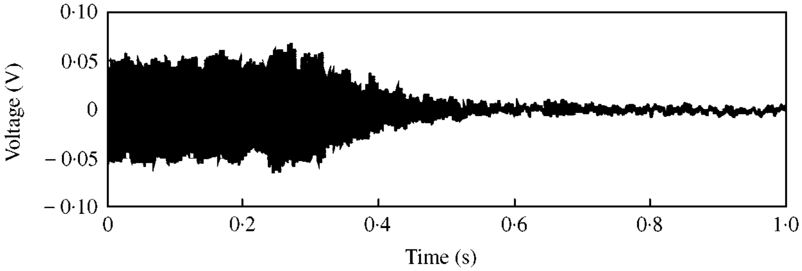

Figure 9. Time history of the residual error after applying the proposed algorithm.

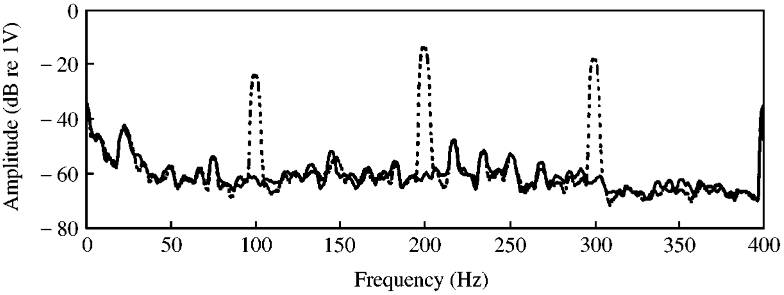

Figure 10. The spectrum of the residual error before (·····) and after (——) applying the proposed algorithm for the transformer noise.

converters and 2 channel 16-bit D/A converters. The Codec has on-chip low-pass filters for analog signal and programmable Gain control for the microphone input, so no low-pass filter or pre-amplifier are shown in Figure 8.

4.2. EXPERIMENTAL RESULTS

Figure 9 shows the time history of the residual error after applying the proposed algorithm to a sine wave of 200 Hz. The control starts at about 0·2 s. It can be seen that the proposed algorithm can successfully reduce the amplitude of the error signal within almost half second.

Figure 10 shows the result for the transformer noise. Note that the primary noise in the figure was not exactly the same as the noise recorded around the transformer because the loudspeaker used to reproduce the sound was not ideal. As can be seen from the figure, the proposed algorithm can successfully control the first three harmonics simultaneously. There is no change in the 400 Hz component because it was not considered in the current programming.

The proposed algorithm was compared with the FXLMS algorithm, which is realized on a commercial active controller, EZ-ANC from Causal Systems. It was found that the FXLMS algorithm can also successfully reduce the above transformer noise to the same level. However, if both the primary transfer function and cancellation path transfer function were changed simultaneously by adjusting the programmable filters in Figure 8, the FXLMS algorithm had a large residual error, resulting in an unstable system. Figure 11 shows the results. Figure 11(a) is the time history of the residual error after applying the

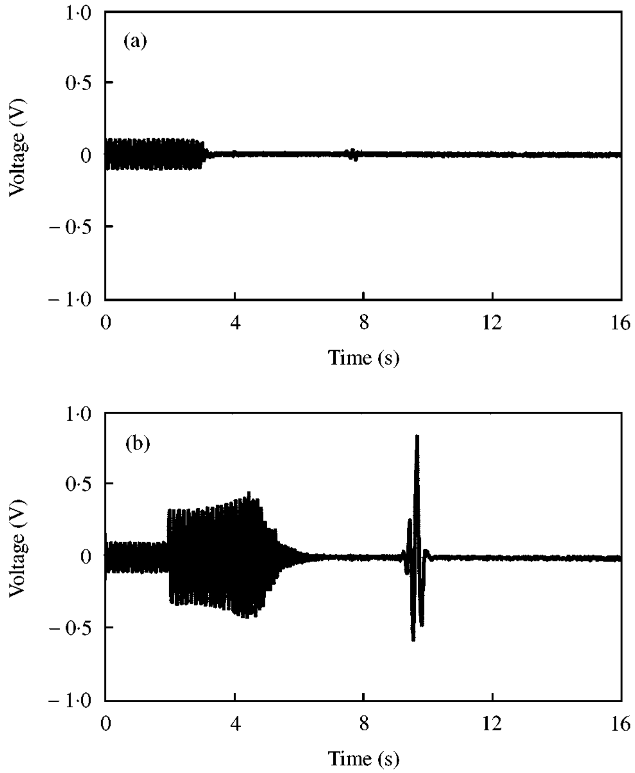

Figure 11. Time history of the residual error after applying (a) the proposed algorithm and (b) the FXLMS algorithm with changes occur both on the primary sound field and cancellation path.

proposed algorithm. The original signal was about 0·1 Vpp and the control started at about the third second. The residual error reduced very quickly and just had a small rip when the sudden change occurred at about seventh second. Figure 11(b) is the time history of the residual error after applying the FXLMS algorithm. The original signal was about 0·1 Vpp and the control started at about the 2nd second. It took about 4 s to reduce the noise to the bottom noise level. The residual error had a very large jump (almost 1 V) when the sudden change occurred at about the 9th second, which indicates that the controller became unstable during that time. (However, the controller eventually converged again due to the special stability enhancing features of the EZ-ANC, which are not, however, associated with the FXLMS algorithm.)

5. CONCLUSIONS

A new adaptive algorithm based on waveform synthesis has been described for the control of a slowly time-varying system with a few fixed frequency components, using a perturbation method to obtain the CPTF on-line. The convergence condition of the update algorithm and some implementation considerations such as system sampling rate, average number and the influence of unwanted noise were also discussed. A comparison of the proposed algorithm with the H-TAG algorithm and the FXLMS algorithm showed its feasibility and superior performance.

ACKNOWLEDGMENTS

The research described in this publication was funded by a grant from the Electricity Supply Association of Australia Limited and the Australian Research Council.

REFERENCES

1. S. E. Craig and O. L. Angevine 1993 *Proceedings of the Second Recent Advances in Active Control of Sound and Vibration*, 279–290. Active control of hum from large power transformer—the real world.
2. M. McLoughlin, S. Hidebrand and Z. Hu 1994 *Proceedings of the Inter-Noise* **94**, 1323–1326. A novel active transformer quieting system.
3. S. M. Kuo and X. H. Jiang 1995 *Proceedings of Inter-Noise* **95**, 505–508. Secondary path modelling technique for transformer active noise control.
4. K. Brungardt, J. Vierengel and K. Weissman 1997 *Proceedings of Noise-Con* **97**, 173–182. Active structural acoustic control of noise from the power transformers.
5. B. Widrow and S. D. Stearns 1985 *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall.
6. C. C. Boucher, S. J. Elliott and P. A. Nelson 1991 *Proceedings of the Recent Advances in Active Control of Sound and Vibration*, 290–301. The effects of modelling errors on the performance and stability of active noise control systems.
7. C. Bao, P. Sas and H. V. Brussel 1993 *Proceedings of the Second Recent Advances in Active Control of Sound and Vibration*, 38–51. Comparison of two on line identification algorithms for active noise control.
8. S. M. Kuo and D. R. Morgan 1996 *Active Noise Control Systems—Algorithms and DSP Implementations*. New York: John Wiley & Son Inc.
9. C. H. Hansen and S. D. Snyder 1997 *Active Control of Noise and Vibration*. London: E&FN SPON.
10. L. J. Eriksson 1991 *Journal of the Acoustical Society of America* **89**, 257–265. Development of the filtered-U algorithm for active noise control.
11. S. D. Sommerfeldt 1991 *Noise control Engineering Journal* **37**, 77–89. Multi-channel adaptive control of structural vibration.
12. Scott C. Douglas 1997 *Proceedings of Noise-CON* '97. Reducing the computational and memory requirements of the multichannel filtered-x LMS adaptive controller.
13. G. Nijsse, M. Verhaegen, B. De Schutter, D. Westwick and N. Doelman 1999 *Proceedings of Active* '99, 909–920. State space modelling in multichannel active control system.
14. R. H. Cabell and C. R. Fuller 1999 *Journal of Sound and Vibration* **227**, 159–181. A principal component algorithm for feedforward active noise and vibration control.
15. T. Usagawa, Y. Shimada, Y. Nishimura and M. Ebata 1997 *Proceedings of Active*, **97**, 837–848. Adaptive algorithm for active control of harmonic signal with online error path modelling.
16. T. Usagawa, Y. Shimada and M. Ebata 1998 *Proceedings of Inter-noise* '98. Adaptive algorithm for active control of harmonic signal with online error path modelling.
17. H. S. Kim and Y. Park 1998 *Journal of Sound and Vibration* **212**, 875–887. Delayed-x LMS algorithm: an efficient ANC algorithm utilizing robustness of cancellation path model.
18. S. M. Lee, H. J. Lee, C. H. Yoo, D. H. Youn and I. W. Cha 1998 *Journal of the Acoustical Society of America* **104**, 248–254. An active noise control algorithm for controlling multiple sinusoids.
19. S. H. Oh and Y. Park 1998 *Proceedings of Inter-Noise* '98. Active noise control algorithm using IIR based filter.
20. G. P. Gibbs and R. L. Clark 1993 *Proceedings of Noise-CON* '93. Feedforward higher harmonic control using the H-TAG algorithm.
21. D. L. Kewley, R. L. Clark and S. C. Southward 1995 *Journal of the Acoustical Society of America* **97**, 2892–2905. Feedforward control using the higher-harmonic, time-averaged gradient descent algorithm.
22. Y. Maeda 1999 *Proceedings of Active* '99, 985–994. An active noise control without estimation of secondary path—ANC using simultaneous perturbation.

23. X. Qiu and C. H. Hansen 1999 *Proceedings of Active* '99, 1057–1068. An adaptive waveform synthesis algorithm for active control of transformer noise.

24. X. Qiu and C. H. Hansen 1997 *Proceedings of the Fifth International Conference on Sound and Vibration*. An adaptive sound intensity control algorithm for active control of transformer noise.