COMPUTATIONALLY EFFICIENT ALGORITHM FOR NARROWBAND ACTIVE SOUND PROFILING

Jari Kataja, Marko Antila and Guangrong Zou

VTT Technical Research Centre of Finland, Tekniikankatu 1, 33720 Tampere, Finland
e-mail: jari.kataja@vtt.fi

In active sound profiling (ASP), the target is to obtain a certain sound field or profile using similar techniques as in active noise control (ANC). Active sound profiling can be applied to a passenger car, for example, to modify the engine sound in the cabin. A fundamental algorithm in active sound profiling is the command-FXLMS (C-FXLMS) algorithm, which is an extension of the famous FXLMS algorithm widely used in active noise control. The computational demand of the C-FXLMS algorithm is dominated by the reference signal filtering. In order to reduce the computational burden, an alternative method to modify periodic reference signals has been developed. Since the reference signals are periodic, a method delaying the signals and modifying their magnitude can be used. The magnitude and phase delay values are obtained via the system identification process and stored in lookup tables. The new method is computationally efficient but requires more memory than the conventional filtering. In this paper, the C-FXLMS algorithm has also been combined with the eigenvalue-equalized FXLMS (EE-FXLMS) algorithm. In the EE-FXLMS algorithm, the secondary-path model is modified so that the magnitude becomes flat. This leads to frequency-independent step sizes in the adaptation of the algorithm, increasing robustness and enabling easier tuning of the step sizes. The memory requirement is also decreased since the size of the lookup table for magnitude values is significantly reduced. The eigenvalue-equalized C-FXLMS algorithm using the lookup-table based reference signal compensation has been tested in a simulation model. The model is based on an experimental ASP system installed in a passenger car. Simulations have been carried out both in ASP and ANC. Results prove that the algorithm converges fast and, in ASP, is able to track the desired levels with sufficient accuracy. In ANC, the engine sound is significantly attenuated.

1. Introduction

Active noise control (ANC) is a technique to cancel unwanted sound using adjustable secondary sound.\(^1\) In active sound profiling (ASP), the target is to obtain a certain sound field or profile. The sound power over specific frequencies can be altered in a desired way, even by amplifying it. Active sound profiling can be used for increasing the sound quality in a passenger car by modifying the engine sound inside the car cabin.

One of the basic algorithms in active sound profiling is the command-FXLMS (C-FXLMS) algorithm, which is an extension of the famous FXLMS algorithm widely used in active noise control. The command-FXLMS (C-FXLMS) algorithm is relatively simple, provides fast convergence
and is stable against errors in the secondary-path model.\textsuperscript{2} A drawback of the C-FXLMS algorithm is that the control effort becomes excessive if the command signal is out of phase with the disturbance. More sophisticated algorithms have also been developed for active sound profiling. They include Internal Model FXLMS, Phase Scheduled Command-FXLMS and Automatic Phase Command-FXLMS.\textsuperscript{2,3} They are extensions to the command-FXLMS algorithm and provide a smaller control effort but, on the other hand, they are not as stable as the command-FXLMS. The extended algorithms are also more complicated and thus require more computation.

The adaptive noise equalizer (ANE) is also suitable for active sound profiling.\textsuperscript{4} It is, however, more sensitive to errors in the secondary-path model.\textsuperscript{3} It is also unable to produce a non-zero output if the disturbance signal is zero. ANE can also be combined with filtered-error LMS (FELMS) algorithm and normalized reference signal generator to obtain a more stable system with faster convergence.\textsuperscript{5} This NEX-LMS algorithm can be used for equalization of harmonic noise components.\textsuperscript{6,7} It is based on internal model control and reference signal normalization and provides an improved convergence rate with the cost of increased complexity.

2. Multiple-channel command-FXLMS algorithm

The command-FXLMS (C-FXLMS) algorithm forces the error signal to tend towards a given command signal having predetermined amplitude, instead of zero.\textsuperscript{3} This is done by subtracting the command signal from the error signal and creating a new pseudo-error signal which is minimized by the FXLMS algorithm. The block diagram of the multiple-channel C-FXLMS algorithm is shown in Fig. 1. The multiple-channel system consists of $J$ reference signals, $K$ secondary sources and $M$ error sensors. Adaptive notch filters are used for producing the output signals.\textsuperscript{8} The command signal is denoted as $e(n)$ and the pseudo-error is $e'(n)$.

Matrices $W_1(n)$ and $W_2(n)$ contain the weights of the adaptive notch filters for the reference sine and cosine waves, respectively. Two reference signal vectors, $x_1(n)$ and $x_2(n)$, contain the sine waves and the cosine waves for each reference. The output to the $k$th secondary source is

$$y_k(n) = \sum_{j=1}^{J} w_{kj}(n)x_j(n),$$

where

$$y_{kj}(n) = w_{kj,1}(n)x_{j,1}(n) + w_{kj,2}(n)x_{j,2}(n)$$

is the component of the $k$th output signal derived from the $j$th reference.

There are $M \times K$ secondary paths between the secondary sources and error sensors. They are modelled by FIR filters $\hat{S}_{mk}(z)$ and used for generating the $J \times M \times K$ filtered versions of the sine and cosine waves. The weights of the adaptive notch filters, $W_1(n)$ and $W_2(n)$, are adjusted by the following

\[
\begin{align*}
\Delta W_1(n) &= \mu d(n)x_1(n), \\
\Delta W_2(n) &= \mu d(n)x_2(n).
\end{align*}
\]
multiple-channel FXLMS algorithm. The update equations for the adaptive weights from \( j \)th reference to \( k \)th output are

\[
\mathbf{w}_{kj,i}(n+1) = \mathbf{w}_{kj,i}(n) - \mu \sum_{m=0}^{M-1} x'_{jkm,i}(n) e'_m(n), \quad i = 1, 2,
\]

where \( x'_{jkm,i}(n) \) is the filtered reference signal for the sine and cosine waves, respectively.

The pseudo-error signals \( e'_m(n) \) are computed as

\[
e'_m(n) = e_m(n) - c_m(n),
\]

where \( e_m(n) \) is the error signal from \( m \)th error sensor and \( c_m(n) \) is the corresponding command signal. There are \( J \) narrowband components to be controlled so \( c_m(n) \) is a sum of \( J \) sinusoids having the desired amplitudes at the \( m \)th error sensor.

Computationally, the most time-consuming part of the multiple-channel C-FXLMS algorithm is the filtering of the reference signals. The computational burden of the parts of the C-FXLMS algorithm is given in Table 1. Generation of the sine and cosine waves is not taken into consideration.

### Table 1. Power consumption of the multiple-channel C-FXLMS algorithm.

<table>
<thead>
<tr>
<th></th>
<th>Multiplications</th>
<th>Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter adaptation</td>
<td>( 2JK(M+1) )</td>
<td>( 2JKM+M )</td>
</tr>
<tr>
<td>Output computation</td>
<td>( 2JK )</td>
<td>( 2K(J-1) )</td>
</tr>
<tr>
<td>Reference signal filtering</td>
<td>( 2JKML_s )</td>
<td>( 2JKM(J_s-1) )</td>
</tr>
<tr>
<td>Error signal modification</td>
<td>( JM )</td>
<td>( M(J-1) )</td>
</tr>
</tbody>
</table>

In a typical multiple-channel ASP system in a car cabin, the number of references, \( J \), is 3–6, number of outputs, \( K \), is 2–8 and number of errors, \( M \), is 2–8. The length of the secondary-path model, \( L_s \), is typically quite large so that a sufficient estimate of the secondary-path response is obtained. In an ASP system consisting of 6 reference signals, 4 secondary sources and 4 error sensors and \( L_s \) chosen to be 100, almost 99% of the computational power is taken by the reference signal filtering. In order to reduce the computational burden of the multiple-channel C-FXLMS algorithm, other techniques for obtaining the filtered reference signals have to be considered.

### 3. Simplified methods for filtering periodic signals

In a narrowband system, the reference signals are sinusoidal and filtering such signals is equal to modification of the magnitude and phase of the sine waves. The filtered reference signals can be obtained by shifting the phase of each narrowband component. An adaptive delay filter (ADF) consisting of a variable delay and gain can be used for estimating the time delay and the gain difference between the primary and secondary paths at a given frequency. The performance of the ADF depends on the delay difference between the primary and secondary paths, and maximal performance is obtained when the delay is an integer. In the following, approximating the secondary-path model as a delay is studied.

#### 3.1 Delay compensation

In the simplest case, the secondary-path model is replaced by \( z^{-\Delta} \), where \( \Delta \) is the secondary-path delay. The delay is typically a function of frequency, however, and the frequency-dependent delay can be expressed as

\[
\Delta = \text{round} \left( \frac{\phi(\omega)}{\omega T} \right), \quad (5)
\]

where \( \phi(\omega) \) is the phase of the secondary-path model at frequency \( \omega \), \( T \) is the sampling period and round(\( * \)) is a function rounding the value to the nearest integer.
Since the delay is a function of frequency, a method for selecting the correct delay value is needed if the frequency changes. One solution is to use a lookup table containing the frequency-dependent delay values.\textsuperscript{13} In the lookup-table based delaying, a buffer is used to store the reference signal samples. The buffer can be denoted as

\[ X = [x(n) \ x(n-1) \ \ldots \ x(n-\Delta) \ \ldots], \tag{6} \]

where \( x(n) \) is the reference signal sample at time \( n \) and \( \Delta \) is the delay in samples. The delayed reference signal is obtained by selecting the value \( x(n-\Delta) \). The buffer is updated at each time step by shifting the samples rightwards and inserting the new sample in the leftmost position. If the frequency and hence the delay changes, the delayed reference signal sample can be read from the buffer by taking the value located in the position corresponding to the new delay.

In adaptive notch filters, two buffers are needed for each reference signal. The buffers for the sine and cosine waves are denoted as

\[ X_{ij} = [x_{ij}(n) \ \ldots \ x_{ij}(n-\Delta_{km}) \ \ldots \ x_{ij}(n-\Delta_{km,max})], \quad i = 1,2, \tag{7} \]

where \( x_{ij}(n) \) are the current sine and cosine wave samples for \( j \)th reference. \( \Delta_{km} \) is the delay of the secondary path from the \( k \)th secondary source to the \( n \)th error sensor and \( \Delta_{km,max} \) is the maximum delay of the secondary path. An estimate of the \( j \)th filtered reference signal for the secondary path from the \( k \)th secondary source to the \( m \)th error sensor is

\[ x'_{jkm,i} \approx X_{j,i}[\Delta_{km} + 1] = x_{ij}(n-\Delta_{km}), \quad i = 1,2, \tag{8} \]

with \( X[n] \) meaning that the \( n \)th element of buffer \( X \) is taken and \( x[1] \) corresponds to \( x[n] \).

The buffer pickup does not require computational operations such as multiplications or additions if the lookup tables containing the delay values are calculated offline. There are \( M \times K \) lookup tables for the secondary-path delays and \( 2J \) buffers so a considerable amount of memory is needed, however. The memory usage depends on the maximum delays of the secondary path and the frequency resolution of the delay values.

Rounding the delay value to the nearest integer causes an additional error in the secondary-path model. Writing the delay before rounding as \( I+F \), where \( I \) is an integer and \( F \) is a fractional number between \([-0.5, 0.5)\), the effect of the rounding error can be analysed. The error between the phase delay before and after rounding can then be expressed as

\[ \epsilon(\omega) = \varphi_1(\omega) - \varphi_2(\omega) = -(I + F)\omega T + I\omega T = -F\omega T, \tag{9} \]

where \( \varphi_1(\omega) \) and \( \varphi_2(\omega) \) are the phase delay values before and after rounding.

The maximum error occurs when \( F \) equals \(-0.5\), i.e. when the difference between the unrounded and rounded values is largest. The error also becomes larger as frequency increases.

### 3.2 Reference signal compensation using magnitude and delay lookup tables

The delay compensation does not take the magnitude of the secondary-path model into account. The lookup table method can be extended to include the magnitude values. Instead of calculating the frequency-dependant magnitude values, the eigenvalue-equalized FXLMS algorithm has been exploited.\textsuperscript{14} The EE-FXLMS algorithm is based on secondary-path models with flat magnitude responses. As a result of the equalization, the magnitudes become constant for each secondary-path model, only one magnitude value is needed for each secondary path. Using the delay buffers, estimates of the filtered reference signal components for the \( j \)th reference signal and secondary path from \( k \)th output to \( m \)th error can be obtained as

\[ x'_{jkm,i} \approx A_{km}X_{j,i}[\Delta_{km} + 1] = A_{km}x_{ij}(n-\Delta_{km}), \quad i = 1,2, \tag{10} \]

where \( A_{km} \) is the magnitude of the corresponding eigenvalue-equalized frequency response.
The magnitude-delay compensation requires \(2JMK\) multiplications as the reference signal values are multiplied by the amplitude values. By exploiting the EE-FXLMS algorithm, the magnitude lookup table is an \(M \times K\) matrix. Another benefit of eigenvalue equalization is that frequency-independent operation is achieved for the command-FXLMS algorithm and constant step sizes can be used.

### 3.3 Magnitude-delay compensation using delay interpolation

In the delay compensation, the rounding error becomes remarkable at higher frequencies. In order to overcome that, linear interpolation can be used to obtain samples between the integer delay values. Using interpolation, estimates of the filtered reference signal components for the \(j\)th reference signal and secondary path from \(k\)th output to \(m\)th error are expressed as

\[
x'_{ikm,i} \approx A_{km} \{ \delta_{km} \{ X_{j,i}[\Delta_{km} + 1] - X_{j,i}[\Delta_{km}] \} + X_{j,i}[\Delta_{km}] \}
\]

where \(\delta_{km}\) is the fractional part of the delay.

Magnitude-delay compensation using delay interpolation requires \(4JMK\) multiplications and \(6JMK\) additions. It is significantly less than the \(2JKML_s\) multiplications and \(2JKM(L_s - 1)\) additions taken by FIR filtering. The method also requires the integer and fractional parts of the delay so the delay values are stored in the lookup tables without rounding.

### 4. Simulation model of the ASP system in a passenger car

For evaluating the operation of the multiple-channel ASP algorithm using the lookup-table based reference signal compensation, a simulation model has been created. The model is based on an experimental ASP system installed in a Ford C-MAX passenger car. The system has 4 loudspeakers and 4 error microphones. The objective is to modify the engine sound at the front seats of the car by driving the dominant engine sound components to desired levels. The target levels depend on the engine RPM and load.

![Figure 2. Block diagram of the simulation model.](image)

The ASP system has been modelled in Simulink. The simulation model is illustrated in Fig. 2. The model consists of two main blocks, the control system block and the plant and excitation block. The plant and excitation block contains an engine model for producing the primary noise signals. The engine model also provides the engine RPM and load information for the control system. The primary paths from the noise source, i.e. engine, to each error microphone have been modelled as time delays. The plant and excitation block also contains the secondary-path models which have been measured from the experimental ASP system installed in the car.

The engine has been modelled as a source of multiple sinusoids and white noise. The engine sound contains the harmonics related to the RPM, called as engine orders. The engine orders to be
controlled (2nd, 3.5th, 4th, 5th, 5.5th and 6th) are generated with frequency-varying amplitudes. In the control system block, the output signals are computed using the C-FXLMS algorithm. The RPM information is used for generating the reference (sine and cosine) signals that are then modified using the plant model to produce the filtered reference signals. The error signals are modified based on the target profile in order to obtain the pseudo-error signals. The plant model is obtained via an offline identification process which is executed before the ASP algorithm is switched on.

Before the ASP system can be switched on, system identification is executed to obtain the lookup tables for the magnitudes and phase delays of the secondary-path models. Standard identification process is used for obtaining the secondary-path FIR filters. The magnitude and phase of each FIR filter are then calculated using FFT. The mean values of the magnitudes and the frequency-dependent delay values for each secondary path are stored in lookup tables. Calculation of the eigenvalue-equalized secondary-path models is thus not necessary since the magnitude response can be approximated by its mean value.

5. Performance of the ASP system with the simulation model

The operation of the ASP system is simulated in a test case with constant and varying engine RPM shown in Fig. 3. The test case simulates driving the car at constant speed and also during acceleration.

Figure 3. Simulated engine RPM.

![Figure 3. Simulated engine RPM.](image)

Figure 4. Simulation results with ASP at each engine order (black = without ASP, red = target, blue = LUT-based compensation, green = FIR-based filtering).

In the simulations, the reference signals of the C-FXLMS algorithm are generated using conventional FIR filters and the magnitude-delay compensation based on lookup tables. The perfor-
mance of the ASP algorithm is evaluated by calculating the amplitudes of the error signals at each engine order. Mean values of the error signal amplitudes are taken and shown as a function of time in Fig. 4. The ASP system is initially switched off and, at 2 seconds, the system is switched on.

The convergence curves of the ASP system indicate that the algorithm converges fast to the target levels at all engine orders, except the 3.5th engine order. The error between the target level and the actual amplitude is less than 3 dB at the 2nd and 4th engine orders. At the other engine orders, the tracking error is larger but the target levels have been tracked with adequate accuracy. The filtering method of the reference signals has a minimal effect on the convergence of the algorithm. At the 5.5th engine order, the error between the actual and target levels is clearly lower with FIR filtering than with LUT-based compensation.

The simulation model was also tested in active noise control so that the objective was to attenuate each engine order. Results of the ANC system are given in Fig. 5. All engine orders have been attenuated and the best performance has been obtained at the lowest engine orders. At the 2nd engine order, attenuation is 15–20 dB at constant RPM. At the 6th engine order, attenuation is about 6 dB. During acceleration when RPM is varied, performance has been decreased but all engine orders have still been attenuated. With FIR filtering, better performance has been obtained at the 5.5th engine order compared to the LUT-based compensation.

Simulation time taken by the LUT-based compensation and FIR filtering were also compared. Simulating the ASP system with the LUT-based compensation took only 4% of the time required by the system using FIR filtering. The computational burden of the LUT-based compensation is thus significantly lower.

6. Conclusions

The command-FXLMS (C-FXLMS) algorithm used in active sound profiling has been studied and its computational demand has been analysed. In multiple-channel systems with several reference signals, loudspeakers and microphones, the computational burden of the algorithm is clearly dominated by the reference signal filtering. In order to reduce the computational burden, an alternative method for modifying the periodic reference signals has been developed. The method is based
on delaying the signals and modifying their magnitude. In the developed method, magnitude and phase delay values are pre-calculated and stored in lookup tables. The C-FXLMS algorithm has also been combined with the eigenvalue-equalized FXLMS algorithm. This leads to constant magnitudes of the secondary-path models, reducing the size of the magnitude lookup table and enabling easier tuning of the step sizes.

The eigenvalue-equalized C-FXLMS algorithm using the lookup-table based reference signal compensation has been tested in a simulation model. The model is based on an experimental ASP system installed in a passenger car. The simulation model has been tested both in active sound profiling and active noise control. Performance with the LUT-based compensation and conventional FIR filtering has also been compared. The simulation results for ASP prove that the algorithm converges fast and is able to track the desired levels with sufficient accuracy. In ANC, the system is able to attenuate all engine orders. Difference between the LUT-based compensation and FIR filtering is minimal.

REFERENCES