




# The Predict and Invert Feedback Active Noise and Vibration Control Algorithm

Paulo A. C. Lopes<sup>1</sup> 

Received: 20 January 2023 / Revised: 18 July 2023 / Accepted: 20 July 2023 /

Published online: 4 August 2023

© The Author(s) 2023

## Abstract

This work proposes an algorithm for feedback ANC that does not require a prior secondary path model and usually remains stable after fast secondary path changes, as other algorithms proposed for feedforward ANC. This is achieved using a recursive least squares algorithm to model the secondary path and the primary noise with an autoregressive moving average model. The resulting model allows for predicting future values of the primary noise. Finally, the primary noise values predicted are filtered by a non-causal inverse of the secondary path model to generate the anti-noise signal. Simulation results attest to the validity of the algorithm in reducing narrowband noise.

**Keywords** Active noise and vibration control (ANVC) · Active noise control (ANC) · Feedback · Online secondary path modeling · Overall modeling · Model predictive control (MPC)

## 1 Introduction

Active noise and vibration control (ANVC) or simply active noise control (ANC) [11, 12, 15, 17, 23, 24, 33, 42] is a class of methods to reduce noise and vibration by generating an anti-noise signal with a phase opposite to the original noise so that both interfere destructively, reducing the noise level. ANC works best at low frequencies where passive techniques are less effective and can be used together with the last. It is finding applications in many fields like noise reduction in cars [40], homes [8], headphones [10, 25, 39], and even vibration of airplane tails [35].

ANC methods can be divided into two categories [23], feedforward ANC and feedback ANC. In feedforward ANC, a reference sensor captures a reference signal

---

✉ Paulo A. C. Lopes  
paulo.lopes@tecnico.ulisboa.pt

<sup>1</sup> Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento, Instituto Superior Técnico, Universidade de Lisboa, INESC-ID/IST/UL, Rua Alves Redol n. 9, 1000-029 Lisbon, Portugal

early in the noise path. Then a controller uses information from this sensor to generate an anti-noise signal that is fed to an actuator to reduce the noise. The residual noise level is measured by an error sensor and used by the controller when generating the anti-noise. In feedback ANC, there is no reference sensor, and the anti-noise signal is generated using only the information from the error sensor. This work focuses on feedback ANC since in many applications may be difficult to obtain a good reference. Feedback ANC has been extensively applied to ANC headphones, for instance, [22, 41, 43, 44] and many other systems [21, 35].

Most feedforward and feedback ANC algorithms (like the filtered-x least mean squares (FxLMS) algorithm [32]) require a model of the secondary path. This model may be obtained offline before the ANC controller starts to reduce noise or online during the operation of the ANC controller. Online secondary path modeling allows the ANC system to cope with slow changes in the secondary path. However, only a few algorithms can cope with fast changes. Eriksson [13] first proposed online secondary path modeling using a small auxiliary white noise signal as input to an adaptive filter. After this, it was proposed to improve the modeling by using an additional adaptive filter to remove the primary noise from the secondary path model error signal [5, 45]. The auxiliary noise power can be chosen so that residual noise to auxiliary noise at the error sensor is constant [9] and it can also be turned off after convergence has been reached [2, 26] and on after the secondary path changes. Finally, it can also be used to model the path from the anti-noise transducer to the reference sensor (feedback path) [1].

In feedforward ANC, the secondary path can also be modeled without the auxiliary noise by the simultaneous equations method [14] or the overall modeling algorithm (OMA) [23]. However, the secondary path model of the OMA algorithm may be incorrect [6, 23, 27]. Nevertheless, even with an incorrect secondary path model, the resulting overall model (primary and secondary path) can be used to determine the optimum controller as in the mirror-modified FxLMS (MMFxLMS) algorithm [27, 28].

There is very little work on online secondary path modeling for feedback ANC. However, when using auxiliary noise, the same algorithms as in feedforward ANC can be used [18, 43]. Unfortunately, to the authors' knowledge, there is no work on online secondary path modeling for feedback ANC without auxiliary noise. Moreover, the anti-noise signal and the (unknown) disturbance signal are usually highly correlated, making it difficult to use the first to obtain an accurate secondary path model.

This work proposes to use a variant of adaptive model predictive control (MPC) [30, 38] to determine the anti-noise signal. In MPC, the plant output (the ANC error signal) future values are predicted using a model and set close to a predetermined signal (zero in ANC) by carefully selecting the present and future values of the plant input signal (anti-noise). The predictions consider the effect that present inputs have on future outputs. The ANC's primary noise signal is the MPC disturbance. However, unlike standard model predictive control applications, in ANC, the plant plus disturbance model is unknown and needs to be estimated, making this an adaptive model predictive control application [30, 38] that has yet to receive much attention. Nevertheless, the model is linear, allowing for simpler solutions, as shown in this work. ANC also falls in the more general field of adaptive control [4] and stochastic adaptive control [7] that,

in general, requires complex solutions with caution and probing (a dual controller) [20]. Adaptive model predictive control applications have been presented in [34] and [31] for ANC. This work builds on these two papers but presents a different approach with lower computational complexity.

Regarding notation, vectors are boldface lowercase letters as  $\mathbf{u}$ . Given a signal  $u(n)$ ,  $\mathbf{u}(n)$  is the vector with its past samples  $\mathbf{u}(n) = [u(n) \dots u(n - N + 1)]^T$  where  $N$  is the vector length, which should be determined from context. The operator  $*$  stands for convolution.

## 2 Proposed Algorithm

In feedback ANC, the residual noise signal  $e(n)$  is given by the sum of the primary noise signal  $d(n)$  (disturbance) with the anti-noise signal  $u(n)$  after passing through the secondary path (or plant) with impulse response  $s(n)$

$$e(n) = d(n) + s(n) * u(n). \quad (1)$$

In the proposed algorithm,  $d(n)$  is modeled by an order  $N_d$  autoregressive (AR) model:

$$d(n) = \tilde{d}(n) - \mathbf{a}_{dx}^T(n)\mathbf{d}(n - 1) \quad (2)$$

or  $\mathbf{a}_d^T(n)\mathbf{d}(n) = \tilde{d}(n)$  where  $\tilde{d}(n)$  is a white noise process and  $\mathbf{a}_d = [1, \mathbf{a}_{dx}]$ . The signal  $\tilde{d}(n)$  is taken to have small power making  $d(n)$  predictable. This happens in narrowband systems, for instance. The secondary path is modeled by an autoregressive exogenous input (ARX) model so that the anti-noise signal  $y(n) = s(n) * u(n)$  is also given by

$$\mathbf{a}_s(n)\mathbf{y}(n) = \mathbf{b}_s(n)\mathbf{u}(n) \quad (3)$$

assuming the model coefficients are constant or slowly varying. Taking the z-transform [36] results in

$$a_d(z)a_s(z)e(z) = a_d(z)b_s(z)u(z) + a_s(z)\tilde{d}(z) \quad (4)$$

and taking the inverse z-transform result in

$$\mathbf{a}^T\mathbf{e}(n) = \mathbf{b}^T\mathbf{u}(n) + \mathbf{a}_s^T\tilde{\mathbf{d}}(n) \quad (5)$$

where  $\mathbf{a} = \mathbf{a}_s * \mathbf{a}_d$  and  $\mathbf{b} = \mathbf{b}_s * \mathbf{a}_d$ .

Estimates for  $\mathbf{a}$  and  $\mathbf{b}$ ,  $\hat{\mathbf{a}}$ , and  $\hat{\mathbf{b}}$  can be obtained by minimizing the residual noise estimation error square sum  $\xi(n) = \sum_{i=0}^{\infty} j^2(n - i)\lambda^i$  where

$$j(n) = e(n) - (\hat{\mathbf{b}}^T(n)\mathbf{u}(n) - \hat{\mathbf{a}}_x^T(n)\mathbf{e}(n - 1)). \quad (6)$$

These estimates can be obtained using the RLS algorithm [19, 31]. The vectors  $\hat{\mathbf{b}}$  and  $\hat{\mathbf{a}}_x$  have length  $N + 1$  and  $N$  resulting in a order  $N$  model. For small  $\tilde{\mathbf{d}}(n)$  (as in narrowband disturbances), persistent excitation, and  $N$  large enough, the minimization

results in  $\hat{\mathbf{a}} = \mathbf{a}$  and  $\hat{\mathbf{b}} = \mathbf{b}$ . The resulting model has canceling poles and zeros at the zeros of  $a_d(z)$  and uncontrollable states that are only excited by  $\tilde{d}(n)$  in the original model, which may be described by a zero input filter in the new model.

This model is the least-squares (LS) one-step-ahead predictor of  $e(n)$  given  $u(n)$ , and it can be used to predict future values of  $e(n)$ . Replacing the LS with the least mean squares (LMS) predictor (since they are equivalent for large  $M$  and stationary signals) and since the last is given by the expected value [3] results in the following. Using measures up to time  $n$ , the predicted residual noise at time  $n + 1$  is

$$\hat{e}(n + 1, n) = E[e(n + 1) | n] = \hat{\mathbf{b}}^T(n)\mathbf{u}(n + 1) - \hat{\mathbf{a}}_x^T(n)\mathbf{e}(n). \tag{7}$$

The predicted residual noise at time  $n + 2$  is

$$\begin{aligned} \hat{e}(n + 2, n) &= E[e(n + 2) | n] \\ &= \hat{\mathbf{b}}^T(n)\mathbf{u}(n + 2) - \hat{\mathbf{a}}_x^T(n)[E[e(n + 1) | n], \mathbf{e}(n)] \\ &= \hat{\mathbf{b}}^T(n)\mathbf{u}(n + 2) - \hat{\mathbf{a}}_x^T(n)[\hat{e}(n + 1), \mathbf{e}(n)], \end{aligned} \tag{8}$$

etc. So,  $\hat{e}(n + i)$  given  $u(n)$  can be obtained simply by iterating the difference equation of the estimated model. Namely,

$$\hat{e}(n + i + 1, n) = \hat{\mathbf{b}}^T(n)\mathbf{u}(n + i + 1) - \hat{\mathbf{a}}_x^T(n)\hat{e}(n + i, n), \tag{9}$$

where

$$\hat{e}(n + i, n) = E[e(n + i) | n] = \begin{cases} e(n + i) & \text{for } i \leq 0 \\ E[e(n + i) | n] & \text{otherwise} \end{cases}. \tag{10}$$

Note, however, that the model formed by  $\hat{\mathbf{a}}_x$  and  $\hat{\mathbf{b}}$  is the LS predictor of  $e(n)$  and may not be the best predictor of  $e(n + i)$  even for  $i = 1$ . It can be tuned to a set of  $u(n)$  components that change in  $u(n + i)$ , especially after fast changes. Regardless, this work will use this predictor in the remaining text.

Let  $u(n) = u_0(n) + u_1(n)$  where

$$u_0(n + i) = \begin{cases} u(n + i) & \text{for } i \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

and

$$u_1(n + i) = \begin{cases} u(n + i) & \text{for } i > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{12}$$

Then, since (9) describes a linear IIR filter [36], then

$$\hat{e}(n + i, n) = \hat{e}_0(n + i, n) + \hat{s}(i) * u_1(n + i) \tag{13}$$

where  $i > 0$ ,  $\hat{s}(i)$  is the impulse response of the IIR filter, and the signal  $\hat{e}_0(n + i)$  can be obtained from iterating the difference equation

$$\hat{e}_0(n + i, n) = \hat{\mathbf{b}}^T(n)\mathbf{u}_0(n + i) - \hat{\mathbf{a}}_x^T(n)\hat{e}_0(n + i - 1, n), \tag{14}$$

with  $\hat{e}(n+i) = e(n+i)$  for  $i \leq 0$ . Finally, let

$$e_0(n+i) = \begin{cases} e(n+i) & \text{for } i \leq 0 \\ 0 & \text{otherwise} \end{cases}. \quad (15)$$

Since the goal is to drive  $\hat{e}(n+i, n)$  to  $e_0(n+i)$ , keeping past values and driving future values to zero, then  $u_1(n)$  should be

$$u_1(n+i) = -\hat{s}^{-1}(n) * (\hat{e}_0(n+i, n) - e_0(n+i)) \quad (16)$$

where  $\hat{s}^{-1}(n)$  is the impulse response of the inverse of the IIR filter in (9). Let  $\hat{s}(z) = \hat{b}(z)/\hat{a}(z)$  and  $\hat{s}^{-1}(z) = 1/\hat{s}(z)$  be the z-transform of  $\hat{s}(n)$  and  $\hat{s}^{-1}(n)$ , then  $\hat{s}^{-1}(n)$  can be obtained using the inverse z-transform of  $\hat{s}^{-1}(z)$ . To obtain a finite energy signal (and a stable filter), the region of convergence (ROC) of the z-transform should be selected to include the unit circle resulting in general in a non-causal signal (and filter). This (non-causal) filter can be implemented since  $\hat{e}_0(n)$  is known for past and future. Finally,  $u(n+1)$  is set to  $u_1(n+1)$  and used as input to the plant at time  $n+1$ .

In the proposed algorithm, the non-causal filter  $\hat{s}^{-1}(z) = \hat{a}(z)/\hat{b}(z)$  is implemented by numerically calculating the roots of  $\hat{b}(z)$ ,  $z_i$ . Then forming  $\hat{b}_{st}(z)$  by the set of roots with  $|z_i| < \alpha$  with  $\alpha$  slightly greater than one (inside the unit circle or close), and  $\hat{b}_{ut}(z)$  by the set of roots with  $|z_i| \geq \alpha$  (outside the unit circle). Choosing  $\alpha > 1$  assures that canceling poles and zeros at the unit circle (that form oscillators to predict sinusoidal signals) stay in the same filtering operation. Then  $\hat{e}_0(n+i, n) - e_0(n+i)$  is filtered by  $1/\hat{b}_{ut}(z)$  by flipping (time reversing,  $x(-n)$ ) all the signals to make the filter stable; and then by  $\hat{a}(z)/\hat{b}_{st}(z)$ . All the signals extend from  $n-L+1$  to  $n+L$  (length  $2L$ ) with  $L$  large enough so that the transients of the filters are negligible at time  $n+1$ . The source code for the proposed algorithm is available in [29].

### 3 Computational Complexity

The computational complexity of the proposed algorithm is approximately (for large  $N$  and  $L$ )  $32N^2$  for the system identification component,  $4/3N^3$  for the calculation of the  $b(z)$  polynomial roots by calculating the eigenvalues of the companion matrix [16, 37] and  $16LN$  for filtering and prediction. This results in a total of  $32N^2 + 4/3N^3 + 16LN$ . As a comparison, Mohseni's algorithm [31] has a computational complexity of  $L_h N^2 + (8/3 + 6)L_h^3$ , where  $L_h$  is the horizon length. Using the same parameters as in the simulation section,  $f_s = 2000$  Hz,  $N = 16$ ,  $L = 64$ ,  $L_h = 64$  results in 60 MFLOPS for the proposed algorithm and 4577 MFLOPS for Mohseni's algorithm.

Note that it is still possible to reduce the computational complexity by noting that  $\hat{e}_0(z, n) - e_0(z) = g(z)/\hat{a}(z)$  where  $g(z)$  is an order  $N$  polynomial related to the initial conditions for  $\hat{e}_0(n+i, n)$  and that  $u_1(n+i) = -\hat{b}^{-1}(n) * g(z)$  but this will not be further discussed in this work.

**Table 1** Algorithm's parameters

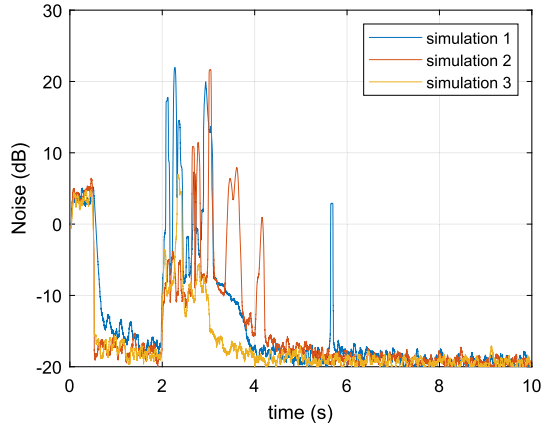
Algorithm	parameters
proposed	$N = 16, L = 64 \lambda = 0.999$ and $\alpha = 1.01$
Carini	$N = 32, M = 32, D = 16, R = 1, \lambda = 0.9, \hat{\lambda} = 0.7, \mu_{SC} = 0.1, \mu_{s, \min} = 0.1$
Lopes	$N_w = 32, N_s = 32, \lambda = 0.9, \mu = 0.1, \mu_s = 0.1, k_r = 0.1, \beta = 5, A = 10$
Mohseni	$n = 16, l = 64, \tau_n = 100, \tau_d = 200, \eta = 0.1, \lambda^{cc} = 0.1, E^{id} = 1, \bar{E}_y = 100\mathbf{I}, \bar{E}_u = \mathbf{I} C_t = 1$

## 4 Simulation Results

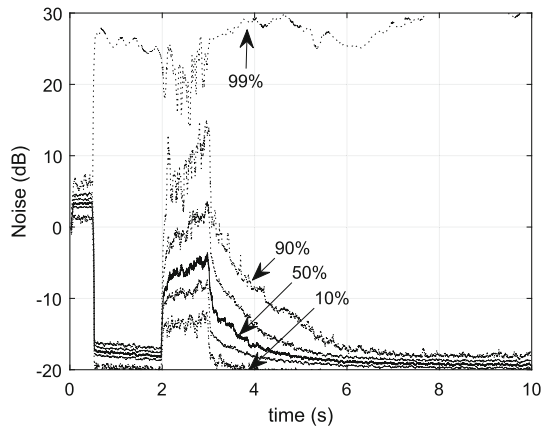
This section presents a set of simulation results. In the simulations, the sampling frequency is  $f_s = 2000$  Hz. The plant is formed by an order  $N_x = 6$  IIR filter. The plant parameters  $\mathbf{a}_s$  and  $\mathbf{b}_s$  coefficients are initially set Gaussian random values with zero mean and variance  $1/N_x^2$  and one, respectively, with  $\mathbf{a}_s$  first coefficient equal to one. The parameters change during the simulation according to random walk model ( $\mathbf{a}_s(n+1) = \mathbf{a}_s(n) + \mathbf{r}_a(n)$ ,  $\mathbf{b}_s(n+1) = \mathbf{b}_s(n) + \mathbf{r}_b(n)$ ) where the state noise  $\mathbf{r}_a$  and  $\mathbf{r}_b$  is formed by a set independent white noise signals with variance  $q_r(n)$ . The plant changes fast from 2 to 3 s with  $q_r = 1/f_s$  and slowly for the remaining time with  $q_r(n) = 1/(400f_s)$ . The filter poles and zeros are kept at a distance greater than  $\beta = 0.1$  of the unit circle at any time  $n$ . Three sinusoidal signals with frequencies of 200, 400, and 600 Hz and amplitudes of 0.5, 1.3, and 0.3 plus white background noise with variance  $q_v = 0.01$  are added at the plant's output. The charts plot the result for 100 simulations of several algorithms. The anti-noise signal saturates at 10 and  $-10$ . Before ANC is started, at 0.5 s, the input of the plant is set to a unit-power white noise signal, and the model identification component of the algorithm is left running. The percentile charts show the percentile 10%, 25%, 50%, 75%, 90%, and 99% residual noise level at each time, given some information about its probability density function (PDF). The algorithms parameters are presented in Table 1. These are based on the values presented in the corresponding papers and tuned by trial and error. In Carini's algorithm, the step size  $\mu$  was scaled by the factor  $\mu_{SC}$  to improve stability.

Figure 1 shows the residual noise level power versus time of three typical simulations of the proposed algorithm. As can be seen, the residual noise quickly settles to the minimum value after ANC is started, rises while the plant changes fast, and settles to the minimum again as desired. However, there are some narrow spikes and even divergence (not shown). Figure 2 shows the percentile chart for the 100 simulations. As can be seen, performance is not always as good as in Fig. 1. The residual noise power takes large values in less than 10% of the simulations. Narrow spikes contribute to a rise in the percentile curves, but they are not visible on the plots. The very high percentile 99% curve is primarily due to divergence in some simulations. Nevertheless, the residual noise takes low values (at each time instant) in more than 90% of the simulations. This is confirmed by the histogram of the residual noise power at the simulation end, as shown in Fig. 3. It shows that the final residual noise power in 10 out

**Fig. 1** Three typical simulations residual noise power versus time



**Fig. 2** Residual noise power percentile curves versus time. Curves for percentile 10%, 25%, 50% (solid), 75%, 90%, and 99% (gray)

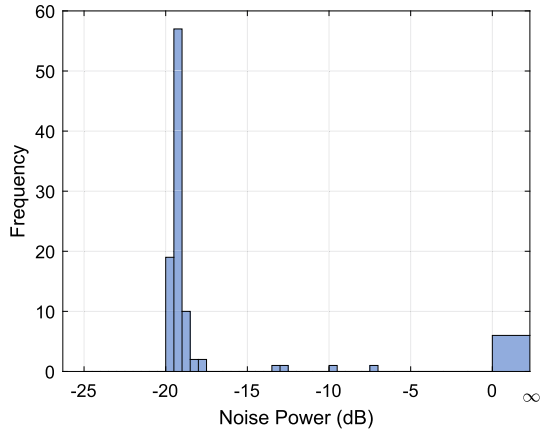


of 100 simulations is considerably larger than the minimum value (with 6 divergence cases), while in the remaining 90, it is close to the optimum.

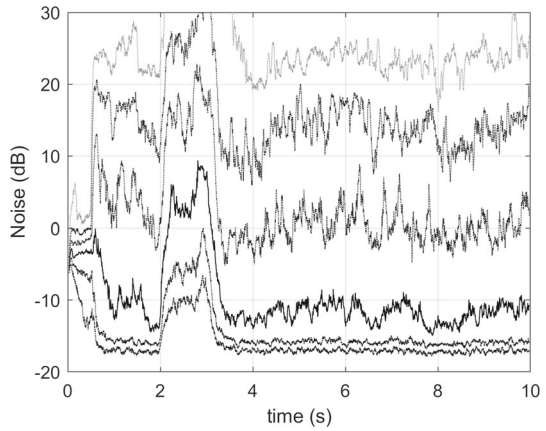
Figures 4 and 5 show percentile plots of the residual noise power versus time of Carini's [9] and Lopes' [26] algorithms simulation results when used in feedback ANC. The reference signal equals the estimated primary noise signal [23]. This signal equals the error signal minus the anti-noise signal filtered by the plant estimate. These algorithms have very low computational complexity. They deal well with slow plant changes and are intended for feedforward (not feedback) ANC systems. So, performance is not good and considerably worse than the proposed algorithm, as can be seen by comparing (for instance) the percentile 90% curve.

Figure 6 shows the same results for Mohseni's algorithm [31]. Performance is much better than Carini's and Lopes's algorithms, in fact also better than the proposed algorithm, but at the expense of higher computational complexity, much larger than the last. However, the residual noise is still high, at least in 1% of the simulations (and at each time  $n$ ), as seen by the percentile 99% curve. Also, there is a degradation of performance with time, as seen by the 90% curve.

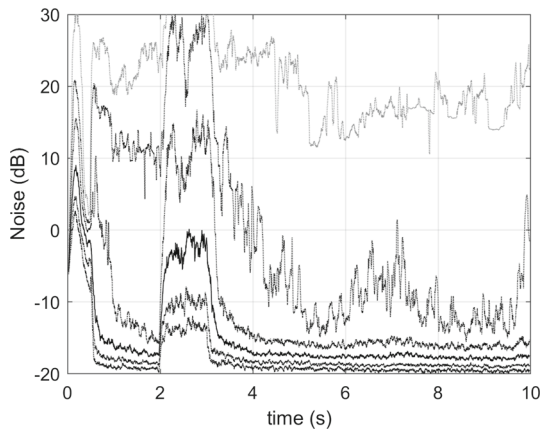
**Fig. 3** Final residual noise power histogram



**Fig. 4** Carini’s algorithm residual noise power percentile curves versus time. Curves for percentile 10%, 25%, 50% (solid), 75%, 90%, and 99% (gray)

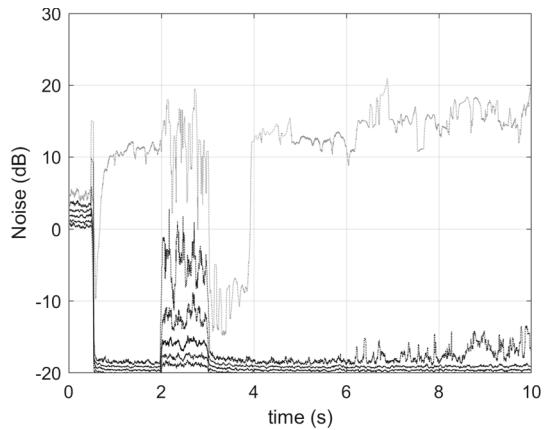


**Fig. 5** Lopes’s algorithm residual noise power percentile curves versus time. Curves for percentile 10%, 25%, 50% (solid), 75%, 90%, and 99% (gray)





**Fig. 6** Mohseni's algorithm residual noise power percentile curves versus time. Curves for percentile 10%, 25%, 50% (solid), 75%, 90%, and 99% (gray)



## 5 Conclusion

This work presents an algorithm for narrowband feedback ANC systems that improves the performance of state-of-the-art algorithms when dealing with fast secondary path changes while keeping a moderate computational complexity. However, performance is as good as desired in every case. This algorithm uses an estimated model to predict the primary noise and the same model to estimate a non-causal secondary path inverse that filters the predicted noise to calculate the anti-noise signal.

**Acknowledgements** This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020.

**Funding** Open access funding provided by FCTIFCCN (b-on).

**Data Availability** Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. S. Ahmed, M.T. Akhtar, Gain scheduling of auxiliary noise and variable step-size for online acoustic feedback cancellation in narrow-band active noise control systems. *IEEE/ACM Trans. Audio Speech Lang. Process.* **25**(2), 333–343 (2016)
2. S. Ahmed, M.T. Akhtar, X. Zhang, Robust auxiliary-noise-power scheduling in active noise control systems with online secondary path modeling. *IEEE Trans. Audio Speech Lang. Process.* **21**(4), 749–761 (2013)
3. B.D. Anderson, J.B. Moore, *Optimal filtering*. Courier Corporation (2012)

4. K.J. Åström, B. Wittenmark, Adaptive control. Courier Corporation (2013)
5. C. Bao, P. Sas, H.V. Brussel, Adaptive active control of noise in 3-d reverberant enclosures. *J. Sound Vib.* **161**, 501–514 (1993)
6. C. Bao, P. Sas, H.V. Brussel, Comparison of two on-line identification algorithms for active noise control. *Proc. Recent Advances in Active Control of Sound Vibration*, pp. 38–51 (1993)
7. Y. Bar-Shalom, Stochastic dynamic programming: caution and probing. *IEEE Trans. Autom. Control* **26**(5), 1184–1195 (1981)
8. L. Bhan, G. Woon-Seng, Active acoustic windows: towards a quieter home. *IEEE Potent.* **35**(1), 11–18 (2016)
9. A. Carini, S. Malatini, Optimal variable step-size NLMS algorithms with auxiliary noise power scheduling for feedforward active noise control. *IEEE Trans. Audio Speech Lang. Process.* **16**(8), 1383–1395 (2008)
10. C.Y. Chang, A. Siswanto, C.Y. Ho, T.K. Yeh, Y.R. Chen, S.M. Kuo, Listening in a noisy environment: integration of active noise control in audio products. *IEEE Consum. Electron. Mag.* **5**(4), 34–43 (2016)
11. S. Elliott, Down with noise. *IEEE Spectrum*, pp. 54–61 (1999)
12. S. Elliott, *Signal Processing for Active Control* (Academic Press, London, 2001)
13. L.J. Eriksson, M.C. Allie, Use of random noise for on-line transducer modeling in an adaptive active attenuation system. *J. Acoust. Soc. Am.* **85**(2), 797–802 (1989)
14. K. Fujii, J. Ohga, Method to update the coefficients of the secondary path filter under active noise control. *Signal Process.* **81**(2), 381–387 (2001)
15. C. Fuller, S. Elliott, P. Nelson, *Active Control of Vibration* (Academic Press, London, 1996)
16. G.H. Golub, C.F.V. Loan, *Matrix Computations*, 3rd edn. (Johns Hopkins University Press, Baltimore, 1996)
17. C.H. Hansen, *Active Control of Noise and Vibration* (CRC Press, Boca Raton, 2013)
18. H. Hassanpour, P. Davari, An efficient online secondary path estimation for feedback active noise control systems. *Digit. Signal Process.* **19**(2), 241–249 (2009)
19. S. Haykin, *Adaptive Filter Theory* (Prentice-Hall, London, 1996)
20. T.A.N. Heirung, B.E. Ydstie, B. Foss, Dual adaptive model predictive control. *Automatica* **80**, 340–348 (2017)
21. S.M. Kuo, X. Kong, W.S. Gan, Applications of adaptive feedback active noise control system. *IEEE Trans. Control Syst. Technol.* **11**(2), 216–220 (2003)
22. S.M. Kuo, S. Mitra, W.S. Gan, Active noise control system for headphone applications. *IEEE Trans. Control Syst. Technol.* **14**(2), 331–335 (2006)
23. S.M. Kuo, D.R. Morgan, *Active Noise Control Systems, Algorithms and DSP Implementations* (Wiley, Hoboken, 1996)
24. S.M. Kuo, D.R. Morgan, active noise control: a tutorial review **87**(6), 943–973 (1999)
25. S. Liebich, J. Fabry, P. Jax, P. Vary, Acoustic path database for anc in-ear headphone development, in *Proceedings of the 23th International Congress on Acoustics* (Universitätsbibliothek der RWTH Aachen, 2019), p. 4326–4333
26. P.A. Lopes, J.A. Gerald, Auxiliary noise power scheduling algorithm for active noise control with online secondary path modeling and sudden changes. *IEEE Signal Process. Lett.* **22**(10), 1590–1594 (2015)
27. P.A. Lopes, J.A. Gerald, Frequency domain analysis of the mirror-modified filtered-x least mean squares algorithm with low ambient noise. *Int. J. Adapt. Control Signal Process.* **35**(7), 1370–1387 (2021)
28. P.A. Lopes, J.A. Gerald, M.S. Piedade, The mmfxlms algorithm for active noise control with on-line secondary path modelling. *Digit. Signal Process.* **60**, 75–80 (2017)
29. P.A.C. Lopes, Source code for the predict and invert feedback ANVC algorithm. <https://github.com/paclopes/Predict-and-Invert-FANVC>
30. D.Q. Mayne, Model predictive control: recent developments and future promise. *Automatica* **50**(12), 2967–2986 (2014)
31. N. Mohseni, T.W. Nguyen, S.A.U. Islam, I.V. Kolmanovsky, D.S. Bernstein, Active noise control for harmonic and broadband disturbances using RLS-based model predictive control. *American Control Conference (ACC)*, (IEEE, 2020), p. 1393–1398
32. D. Morgan, An analysis of multiple correlation cancellation loops with a filter in the auxiliary path. *IEEE Trans. Acoust. Speech Signal Process.* **28**(4), 454–467 (1980)
33. P. Nelson, S. Elliott, *Active Control of Sound* (Academic Press, London, 1996)

34. T.W. Nguyen, S.A.U. Islam, A.L. Bruce, A. Goel, D.S. Bernstein, I.V. Kolmanovsky, Output-feedback RLS-based model predictive control, in *2020 American Control Conference (ACC)* (IEEE, 2020), p. 2395–2400
35. W. Niu, C. Zou, B. Li, W. Wang, Adaptive vibration suppression of time-varying structures with enhanced FxLMS algorithm. *Mech. Syst. Signal Process.* **118**, 93–107 (2019)
36. A.V. Oppenheim, R.W. Schaffer, *Discrete-Time Signal Processing* (Academic Press, London, 1999)
37. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 2007)
38. J.B. Rawlings, D.Q. Mayne, M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, vol. 2 (Nob Hill Publishing, Madison, 2017)
39. P. Rivera Benois, P. Nowak, E. Gerat, M. Salman, U. Zölzer, Improving the performance of an active noise cancelling headphones prototype, in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings, vol. 259* (Institute of Noise Control Engineering, 2019), p. 889–900
40. P.N. Samarasinghe, W. Zhang, T.D. Abhayapala, Recent advances in active noise control inside automobile cabins: toward quieter cars. *IEEE Signal Process. Mag.* **33**(6), 61–73 (2016)
41. T. Schumacher, H. Krüger, M. Jeub, P. Vary, C. Beaugeant, Active noise control in headsets: A new approach for broadband feedback anc, in *2011 IEEE International conference on acoustics, speech and signal processing (ICASSP)* (IEEE, 2011), p. 417–420
42. S.D. Snyder, *Active Noise Control Primer* (Springer, New York, 2012)
43. H.S. Vu, K.H. Chen, A high-performance feedback FxLMS active noise cancellation VLSI circuit design for in-ear headphones. *Circuits Syst. Signal Process.* **36**(7), 2767–2785 (2017)
44. L. Wu, X. Qiu, Y. Guo, A generalized leaky FxLMS algorithm for tuning the waterbed effect of feedback active noise control systems. *Mech. Syst. Signal Process.* **106**, 13–23 (2018)
45. M. Zhang, H. Lan, W. Ser, Cross-updated active noise control system with online secondary path modeling **9**(5), 598–602 (2001)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.