



Assessment of stability of distributed FxLMS active noise control systems[☆]

Miguel Ferrer^a, Víctor M. García-Mollá^b, Antonio M. Vidal-Maciá^b, Maria de Diego^a, Alberto Gonzalez^{a,1,*}

^a Institute of Telecommunications and Multimedia Applications (iTEAM), Universitat Politècnica de València (UPV), Valencia, 46022 SPAIN

^b Department of Information Systems and Computing (DSIC), Universitat Politècnica de València (UPV), Valencia, 46022 SPAIN



ARTICLE INFO

Article history:

Received 13 January 2023

Revised 5 April 2023

Accepted 27 April 2023

Available online 2 May 2023

Keywords:

Acoustic sensor networks

Distributed adaptive filters

Active noise control

Eigenvalue calculation

stability analysis

ABSTRACT

This paper addresses the assessment of the stability of distributed active noise control (ANC) systems, which are designed to cancel acoustic noise at given points in space. These systems distribute the control task across several simple acoustic nodes that generate the control signals by filtering a noise reference signal. The coefficients of each node filter are iteratively calculated by the filtered-X LMS algorithm. The nodes remain stable when the adaptive filters computed in each node converge to finite values. However, the acoustic coupling among nodes can cause instability (i.e., divergence). Collaboration among nodes is required to avoid this phenomenon. It is shown that the properties of the system are summarized in a system matrix and that the system remains stable when the real parts of all of the eigenvalues of this system matrix are positive. However, computation of all of the eigenvalues is computationally expensive. In this paper, we propose a fast method for checking the positiveness of the real parts of the system matrix eigenvalues. It is shown that the proposed method is faster than the direct calculation of eigenvalues and it assesses the stability/instability of the ANC systems without any false stability outcomes and more accurately than existing alternatives.

© 2023 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

1. Introduction

Multichannel active control noise (ANC) applications [1–4] reproduce specific sound signals (anti-noise signals) to cancel out a given noise within an area of the space. These systems generate the control sound signals from a version of the actual noise, called reference signal, by filtering it through a set of filters, which are usually finite impulse response (FIR) filters [5]. The filters are iteratively adjusted making use of measured signals in the control (quiet) area and multichannel adaptive signal processing algorithms. The filtering of the reference noise signal and the multichannel algorithms [2] to calculate the filters' coefficients could be

executed by a central processor that has enough computational capacity. This processor must be capable of handling many input and output signals in a general case. In [6], a floating-point architecture to achieve a high-efficiency implementation of a centralized multichannel adaptive algorithm for ANC applications is introduced and evaluated in a real-time setup.

Alternatively, distributed systems [7–10] have been proposed based on the use of several units of lower capacity equipment (both computational and signal handling), which we call acoustic nodes, to avoid the need to use high-performance equipment. When working in parallel, these nodes provide the system with the ability to handle a large number of input and output signals and with sufficient versatility, scalability, and robustness (redundancy) to deal with possible failures in a single processing unit or the need for system performance improvement. The simplest version of an acoustic node is composed of one speaker (acoustic actuator), one microphone (acoustic sensor), a simple computer unit where a controller (usually managed by an adaptive filter) is executed, and communication capability (network connection). This node carries out the cancellation of noise at its microphone by generating an anti-noise signal that destructively interferes with

[☆] This work has been partially supported by RTI2018-098085-BC41 and PID2021-125736OB-I00 (MCIN/AEI/10.13039/501100011033/, ERDF A way of making Europe) and GVA through PROMETEO/2019/109.

* Corresponding author.

E-mail addresses: mferrer@dcom.upv.es (M. Ferrer), vmgarcia@dsic.upv.es (V.M. García-Mollá), avidal1972@gmail.com (A.M. Vidal-Maciá), mdediego@dcom.upv.es (M.d. Diego), agonzal@dcom.upv.es (A. Gonzalez).

¹ Eurasip member.

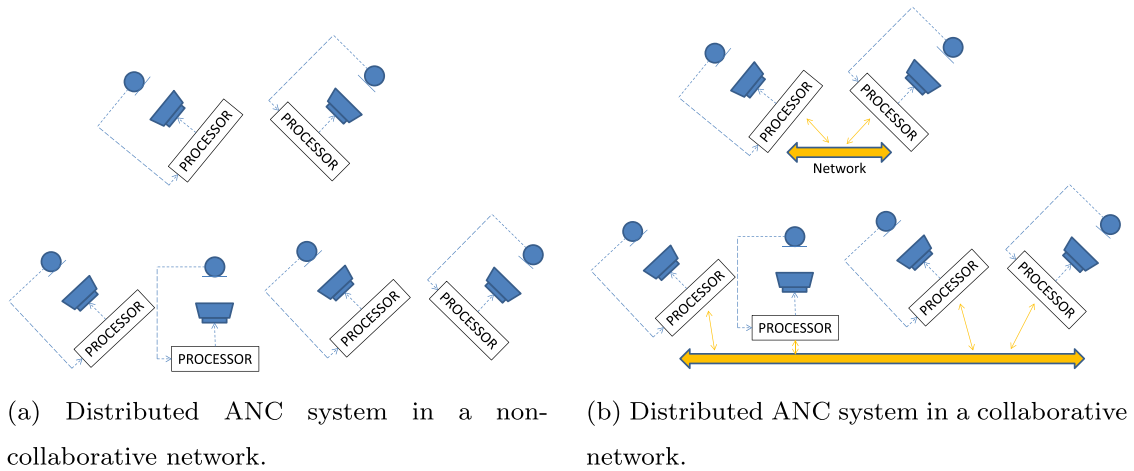


Fig. 1. Examples of different configuration networks for ANC. The network allows processors to share data in collaborative mode.

the actual noise. The anti-noise is generated by filtering a reference signal that is correlated with the actual noise through an adaptive FIR filter, whose coefficients can be adjusted using a version of the filtered-x LMS algorithm (FxLMS) [11,12]. A feedforward structure without feedback between the secondary sources and the reference signals is considered.

It can be shown that the coefficients of the adaptive filter of the controller of each acoustic node achieve a stable steady state under common adaptive filtering and signal conditions, when working in isolation. This means that the coefficients of the adaptive filter converge in mean to finite values. When this convergence happens, we say that the (single-node) system is stable. However, a network is comprised of several nodes working simultaneously within the same acoustic space; thus, the signal generated by each node can interfere with the signals from other nodes. This interference from other nodes can lead the system to instability, which means in this context that the adaptive filter coefficients of some of the nodes diverge, unless the nodes collaborate to a certain extent. In this paper, we are mainly concerned with the study and computation of the stability conditions of multiple-node networks. Note that there are several internal feedback loops between the multiple secondary sources and the multiple error sensors during the coefficient adaptation process. This fact could lead the system to diverge, which motivates the proposed stability analysis.

The distributed algorithms (running on distributed nodes) for ANC can work in collaborative [10,13,14] or non-collaborative [15,16] modes. The collaborative mode involves data exchange among either all of the nodes or a subset of them, whereas the non-collaborative mode does not allow any exchange. Recently, several decentralized solutions have been proposed based on the eigenvalue shaping approach, such as [17]. However, they require a preprocessing of reference signals and only address the two-channel ANC problem. The non-collaborative mode is more sensitive to the acoustic system characteristics, which can affect the convergence of the coefficients of the adaptive filters of the controllers. Thus the rules of collaboration among nodes, and which nodes should collaborate, are vital in order to achieve good performance of the whole system. Although collaboration can be understood as the actions that nodes must perform to share given data, some of these strategies exhibit an outcome equivalent to that of multichannel centralized algorithms. The main purpose of this work is to assess if a set of nodes formed by one sensor and one loudspeaker (single-channel) can work independently within an acoustically coupled environment without stability problems (as

it is illustrated by Fig. 1-(a)), or it needs to be grouped into multi-channel systems using the network (e.g. Fig. 1-(b)).

We will show that the stability of a (collaborative or non-collaborative) distributed system depends mainly on the eigenvalues of a system matrix [18,19], which is formed using the correlation between the reference signal filtered through acoustic channels linking loudspeakers and microphones. The dimensions of this matrix are proportional to the number of nodes and to the length of the adaptive filters, and it can be large when the network grows. Consequently, the computation of the eigenvalues of this matrix can be costly computationally speaking. Furthermore, there are several meaningful practical applications that would require assessing the stability of many ANC systems (i.e., if we search for a stable network setup with minimal collaboration among nodes). Therefore, it is important to efficiently determine the stability of an acoustic network setup of active noise controllers.

In this paper, we propose a fast and approximate method that assesses whether a distributed system is stable for given collaboration rules among nodes. The proposed method can provide incorrect stability diagnoses (by marking the system as unstable when it is stable) in a few cases (false negative, FN). However, it can be shown that it never provides false positives (FP), which is the case that must be avoided in practice. This method outperforms other proposed methods in terms of FN and FP ratios and also outperforms the direct calculation of the eigenvalues in terms of computation time. Therefore, it becomes a practical choice for assessing the stability of distributed ANC controllers. The performance in terms of noise reduction that a stable system can achieve depends mainly on the acoustical characteristics of the system (i.e., acoustical paths among the noise sources and the microphones and among the loudspeakers and the microphones), but its study is out of the scope of this work. Several studies have analyzed statistically the performance of multichannel ANC systems based on the FxLMS in terms of convergence and steady-state behavior. In this regard, a centralized multichannel narrowband ANC system is introduced in [20]. Furthermore, [21] and [22] investigate several diffusion strategies to address the distributed narrowband ANC problem.

The paper is structured as follows. Section 2 is devoted to the study of a network of N single-channel nodes. This analysis is carried out from the point of view of the convergence of the adaptive filters to finite values at the nodes, which implies system stability in practice. In Section 3, we introduce the above-mentioned computationally efficient method to assess the stability of a network of

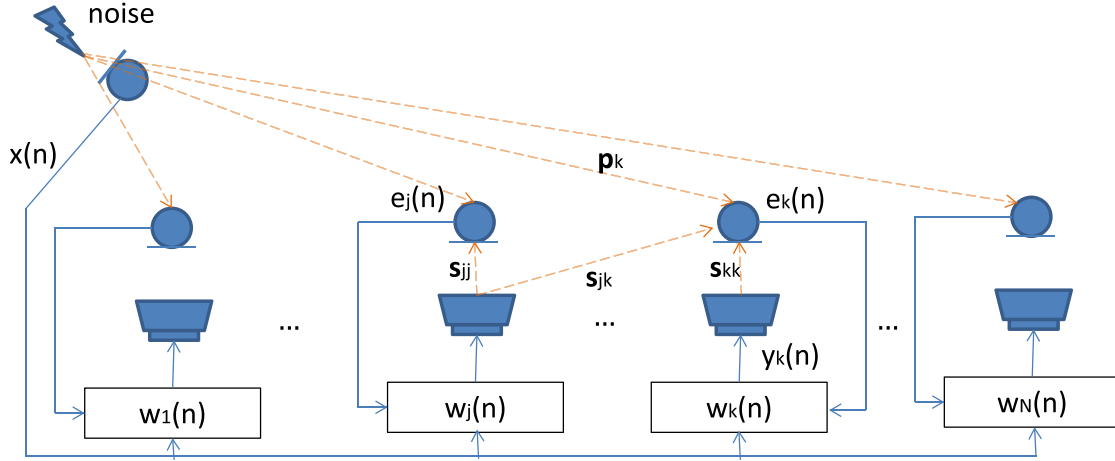


Fig. 2. Single channel network and node signal details (electrical paths in solid lines, acoustic paths in dashed lines).

N single-channel nodes, which are running the FxLMS. The use of this practical and meaningful method represents the main novelty of the present work. The proposed method is validated through extensive simulations in Section 4. Finally, the discussion and conclusions of the work are presented in Sections 5 and 6.

Notation: For the sake of clarity, the following notation has been used throughout this work: italics denote scalars (e.g., x), boldface lower-case letters denote vectors (e.g., \mathbf{x}) and boldface upper-case letters denote matrices (e.g., \mathbf{X}). Boldface subindexes and superindexes are part of the vector or matrix name (e.g., \mathbf{x}_f and \mathbf{R}_B^S) and do not take numerical values. Finally, the expectation operator is denoted by $E\{\cdot\}$.

2. Stability of a network of N single-channel nodes

Given a network of N single-channel nodes with an adaptive filter of L coefficients at each node, we will use the following signal definitions and nomenclature (see Fig. 2):

- $x(n)$: reference noise signal, which is correlated with the noise signal at the nodes' sensors (error microphones).
- $\mathbf{x}(n)$: vector with the last L samples of $x(n)$.
- $\mathbf{w}_k(n)$: vector with the L coefficients of the k -th node FIR filter at time n .
- $\mathbf{w}(n)$: vector with the coefficients of the N FIR filters, $\mathbf{w}_k(n)$ for $k \in \{1, \dots, N\}$, at time n .
- $e_k(n)$: signal picked up by the sensor at k -th node (k -th error signal).
- $\mathbf{e}(n)$: vector with the most recent samples of $e_k(n)$ for $k \in \{1, \dots, N\}$.
- $y_k(n)$: signal rendered by the k -th node, which is generated from the filtering of the reference signal through its corresponding adaptive filter. It can be calculated as: $y_k(n) = \mathbf{x}^T(n)\mathbf{w}_k(n)$.
- \mathbf{s}_{jk} : acoustic path between the j -th node actuator and the sensor of the k -th node. It is modeled as a FIR filter of M order, which has the following coefficients: $\mathbf{s}_{jk} = [s_{jk}(0), s_{jk}(1), \dots, s_{jk}(M)]^T$.
- $\mathbf{x}_{fjk}(n)$: vector comprising the last L samples of the reference signal filtered through the acoustic path between the j -th node actuator and the sensor of the k -th node.
- \mathbf{p}_k : acoustic path between the noise source and the sensor of k -th node.

The system is usually acoustically coupled. This means that the error signal in each node depends on the interference from the rest of the nodes (apart from the signal generated by that node and the noise signal to be canceled). Therefore, we can write this dependence in matrix notation as:

$$\mathbf{e}(n) = \mathbf{d}(n) + \mathbf{x}_f^T(n)\mathbf{w}(n), \quad (1)$$

with $\mathbf{x}_f(n)$ being the following $(LN \times N)$ matrix:

$$\mathbf{x}_f(n) = \begin{bmatrix} \mathbf{x}_{f11}(n) & \mathbf{x}_{f12}(n) & \cdots & \mathbf{x}_{f1N}(n) \\ \mathbf{x}_{f21}(n) & \mathbf{x}_{f22}(n) & \cdots & \mathbf{x}_{f2N}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{fN1}(n) & \mathbf{x}_{fN2}(n) & \cdots & \mathbf{x}_{fNN}(n) \end{bmatrix}, \quad (2)$$

where $\mathbf{x}_{fjk}(n)$ is defined above, and $\mathbf{d}(n) = [d_1(n), d_2(n), \dots, d_N(n)]^T$ is an $(N \times 1)$ vector with the last samples of the noise signal at the N sensors at time n .

We consider that each node is running the FxLMS adaptive algorithm [11,23], with an effort parameter given by β_k [24]. Therefore, when collaboration between nodes is not allowed, the k -th node is updating its coefficients $\mathbf{w}_k(n)$ as follows:

$$\mathbf{w}_k(n+1) = (1 - \mu\beta_k)\mathbf{w}_k(n) - \mu\mathbf{x}_{fkk}(n)e_k(n), \quad (3)$$

where $\mathbf{x}_{fkk}(n)$ is a column vector comprising the last L samples of the reference signal filtered through the acoustic path that joins the k -th node actuator with its sensor, and μ denotes the step-size convergence parameter. Eq. (3) arises from the minimization of the instantaneous squared error plus the system control effort, $J_k(n) = e_k^2(n) + \beta_k\mathbf{w}_k^T(n)\mathbf{w}_k(n)$, by a gradient descent strategy [23] and it can be also derived as a version of the leaky LMS algorithm [25]. For the sake of simplicity, it is assumed that each node has perfect knowledge of the real acoustic paths (perfect estimation) that it needs to update its coefficients. It should be noted that a single node running the FxLMS without collaboration can remain stable as long as the step-size (μ) and effort (β_k) parameters are suitably chosen [18].

When some of the nodes of the network are allowed to update their coefficients using the information handled by other (one or more) nodes, the network is running a distributed collaborative algorithm. Then, let us consider the k -th node. This node tries to cancel the noise signal at its microphone minimizing $J_k(n)$. However, by doing so, the k -th node is creating an interference such that a neighbor node l becomes unstable. To avoid this effect, the cost function of the k -th node should be modified to simultaneously consider (and thus minimize) the signal at its microphone

and at the microphone of the l -th node, $J_k(n) + J_l(n)$. Therefore, the coefficients of the k -th node would be updated through the following adaptive equation [10]:

$$\mathbf{w}_k(n+1) = (1 - \mu\beta_k)\mathbf{w}_k(n) - \mu(\mathbf{x}_{fkk}(n)e_k(n) + \mathbf{x}_{fkl}(n)e_l(n)), \quad (4)$$

which can be generalized (allowing collaboration) as:

$$\mathbf{w}_k(n+1) = (1 - \mu\beta_k)\mathbf{w}_k(n) - \mu\mathbf{x}_{fkk}(n)e_k(n) - \mu \sum_{l \neq k} c_{kl} \mathbf{x}_{fkl}(n)e_l(n), \quad (5)$$

where c_{kl} constants indicate if there is collaboration between nodes k and l , so that $c_{kl} = 1$ if the node k uses information from node l for updating its coefficients, and $c_{kl} = 0$ otherwise. Although the k -th node would apparently need to have access to the error signal of the l -th node to collaborate when $c_{kl} = 1$, distributed algorithms can carry out the collaboration avoiding sharing signals. An example is reported in [10] where an incremental communication of the coefficients of the adaptive filters is implemented and no signals are shared.

Using vector notation, all of the nodes in the network are updating their coefficients as follows:

$$\mathbf{w}(n+1) = (\mathbf{I} - \mu\mathbf{B})\mathbf{w}(n) - \mu\mathbf{X}_{\mathbf{f}k}(n)\mathbf{e}(n), \quad (6)$$

where $\mathbf{w}(n)$ is an $(LN \times 1)$ vector that holds the L coefficients of the N adaptive filters, $\mathbf{e}(n)$ is an $(N \times 1)$ vector with the N error signals at time n , \mathbf{B} is a diagonal matrix where the diagonal elements are repeated in blocks of L elements keeping the effort parameter of each node, from β_1 to β_N , and, finally, $\mathbf{X}_{\mathbf{f}k}(n)$ is an $(LN \times N)$ matrix with the following structure:

$$\mathbf{X}_{\mathbf{f}k}(n) = \begin{bmatrix} \mathbf{x}_{f11}(n) & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{x}_{f22}(n) & \cdots & \cdots & \cdots & \mathbf{0} \\ & \vdots & & & & \\ \mathbf{0} & \cdots & \mathbf{x}_{fkk}(n) & \cdots & \mathbf{x}_{fkl}(n) & \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{x}_{fNN}(n) \end{bmatrix}, \quad (7)$$

where $\mathbf{0}$ is a column vector with L zeros and the vector $\mathbf{x}_{fkl}(n)$ appears as long as $c_{kl} = 1$ ($\forall k, \forall l, l \neq k$).

In the case where all of the nodes collaborate with each other (equivalent to the centralized algorithm), then $\mathbf{X}_{\mathbf{f}k}(n) = \mathbf{x}_f(n)$ and the network would be running a version of the multiple error LMS algorithm [11]. The step-size μ could be different in each node. However, for the sake of simplicity, the same step-size is used for the entire network throughout this paper, which may set stiffer limits in the step-size that could slow down the convergence of some nodes.

Eq. (6) can be rewritten using (1) as:

$$\mathbf{w}(n+1) = (\mathbf{I} - \mu\mathbf{B})\mathbf{w}(n) - \mu\mathbf{X}_{\mathbf{f}k}(n)(\mathbf{d}(n) + \mathbf{x}_f^T(n)\mathbf{w}(n)). \quad (8)$$

The aim is to know if the vector coefficients of the network given by $\mathbf{w}(n)$ will keep finite values after the iterative use of Eq. (8). This means that the network is stable within the context of this work. It should be noted that a given matrix $\mathbf{X}_{\mathbf{f}k}(n)$ encloses implicitly the collaboration coefficients between nodes (c_{kl} , $\forall k, \forall l$) and the effort parameters (β_k , $\forall k$). Thus an alternative collaboration scheme between the nodes must be proposed when Eq. (8) does not converge to finite values for any convergence parameter μ .

Let us assume that we are dealing with stationary zero-mean signals and, when $n \rightarrow \infty$, the filter coefficients of the nodes of the network will converge to $\mathbf{w}(\infty) = \lim_{n \rightarrow \infty} E\{\mathbf{w}(n)\}$, thus the following expression can be derived:

$$\mathbf{w}(\infty) = (\mathbf{I} - \mu\mathbf{B})\mathbf{w}(\infty) - \mu E\{\mathbf{X}_{\mathbf{f}k}(n)\mathbf{d}(n)\} - \mu E\{\mathbf{X}_{\mathbf{f}k}(n)\mathbf{x}_f^T(n)\}$$

$$\mathbf{w}(\infty), \quad (9)$$

and consequently:

$$\begin{aligned} \mathbf{w}(\infty) &= -(E\{\mathbf{X}_{\mathbf{f}k}(n)\mathbf{x}_f^T(n)\} + \mathbf{B})^{-1}E\{\mathbf{X}_{\mathbf{f}k}(n)\mathbf{d}(n)\} \\ &= -(\mathbf{R} + \mathbf{B})^{-1}E\{\mathbf{X}_{\mathbf{f}k}(n)\mathbf{d}(n)\}. \end{aligned} \quad (10)$$

where \mathbf{R} is an $(LN \times LN)$ matrix that contains the cross-correlation matrices of the reference signal filtered by the different acoustic paths and is defined as:

$$\mathbf{R} = E\{\mathbf{X}_{\mathbf{f}k}(n)\mathbf{x}_f^T(n)\} = \begin{bmatrix} \tilde{\mathbf{R}}_{\mathbf{x}f_{11}} & \tilde{\mathbf{R}}_{\mathbf{x}f_{12}} & \cdots & \tilde{\mathbf{R}}_{\mathbf{x}f_{1N}} \\ \tilde{\mathbf{R}}_{\mathbf{x}f_{21}} & \tilde{\mathbf{R}}_{\mathbf{x}f_{22}} & \cdots & \tilde{\mathbf{R}}_{\mathbf{x}f_{2N}} \\ & \vdots & & \\ \tilde{\mathbf{R}}_{\mathbf{x}f_{N1}} & \tilde{\mathbf{R}}_{\mathbf{x}f_{N2}} & \cdots & \tilde{\mathbf{R}}_{\mathbf{x}f_{NN}} \end{bmatrix}, \quad (11)$$

where each new $(L \times L)$ submatrix $\tilde{\mathbf{R}}_{\mathbf{x}f_{rc}}$ that appears in \mathbf{R} is computed as:

$$\tilde{\mathbf{R}}_{\mathbf{x}f_{rc}} = \mathbf{R}_{\mathbf{x}f_{rc}} + \sum_{m \neq r} c_{rm} \mathbf{R}_{\mathbf{x}f_{rm,cm}}, \quad (12)$$

with $\mathbf{R}_{\mathbf{x}f_{rm,cm}}$ being a correlation matrix defined in Eq. (13). The c_{rm} constants in Eq. (12) indicate if there is collaboration between nodes r and m so that $c_{rm} = 1$ if the node r uses information from node m for updating its coefficients, and $c_{rm} = 0$ otherwise.

Each submatrix $\mathbf{R}_{\mathbf{x}f_{rm,cm}}$ in Eq. (12) is $(L \times L)$ with Toeplitz structure:

$$\mathbf{R}_{\mathbf{x}f_{rm,cm}} = \begin{bmatrix} R_{\mathbf{x}f_{rm,cm}}(0) & R_{\mathbf{x}f_{rm,cm}}(1) & \cdots & R_{\mathbf{x}f_{rm,cm}}(L-1) \\ R_{\mathbf{x}f_{rm,cm}}(1) & R_{\mathbf{x}f_{rm,cm}}(0) & \cdots & R_{\mathbf{x}f_{rm,cm}}(L-2) \\ & \vdots & & \\ R_{\mathbf{x}f_{rm,cm}}(L-1) & R_{\mathbf{x}f_{rm,cm}}(L-2) & \cdots & R_{\mathbf{x}f_{rm,cm}}(0) \end{bmatrix}, \quad (13)$$

where $R_{\mathbf{x}f_{rm,cm}}(l) = R_{xx}(l) * s_{rm}(l) * s_{cm}(-l)$ (the operator $*$ represents linear convolution), and $R_{xx}(l) = E\{x(n+l)x(n)\}$, the self-correlation of the reference signal $x(n)$, (see Appendix A).

If $\mathbf{x}_f(n) \approx \mathbf{X}_{\mathbf{f}k}(n)$ and $\mathbf{B} = \mathbf{0}$, then the solution given by Eq. (10) would provide the same solution of the equivalent centralized (or fully collaborative) system. The matrix of effort factors \mathbf{B} adds a bias to the optimal solution that achieves the maximum noise cancellation [11,25,26], consequently $\mathbf{B} = \mathbf{0}$ is preferred in most practical cases.

2.1. Convergence conditions

The translated weight vector $\mathbf{v}(n)$ is defined as: $\mathbf{v}(n) = E\{\mathbf{w}(n)\} - \mathbf{w}(\infty)$. When the filter coefficients of the nodes of the network converge, it holds that $\mathbf{v}(n) \rightarrow 0$ if $n \rightarrow \infty$. Using this definition, expression (8) can be written as:

$$\mathbf{v}(n+1) = (\mathbf{I} - \mu(\mathbf{B} + \mathbf{R}))\mathbf{v}(n), \quad (14)$$

It can be shown that the convergence of the system described by Eq. (14) when n increases depends on the eigenvalues of the matrix $\mathbf{R}_{\mathbf{B}} = \mathbf{B} + \mathbf{R}$. If $\mathbf{R}_{\mathbf{B}}$ is a nondefective matrix [27], then the matrix $\mathbf{R}_{\mathbf{B}}$ fulfills that

$$\mathbf{R}_{\mathbf{B}}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}, \quad (15)$$

where \mathbf{Q} is an $(LN \times LN)$ matrix such that its columns are the eigenvectors of $\mathbf{R}_{\mathbf{B}}$, and $\mathbf{\Lambda}$ is a diagonal matrix with the LN eigenvalues (λ_i) of $\mathbf{R}_{\mathbf{B}}$ along its diagonal. Eq. (14) can be uncoupled considering that the vector $\mathbf{v}(n)$ is a linear transformation of another vector $\mathbf{v}'(n)$ as:

$$\mathbf{v}(n) = \mathbf{Q}\mathbf{v}'(n). \quad (16)$$

Therefore (14) becomes

$$\mathbf{v}'(n+1) = (\mathbf{I} - \mu\mathbf{\Lambda})\mathbf{v}'(n), \quad (17)$$

which can be decomposed in LN uncoupled equations as follows:

$$v'_l(n+1) = (1 - \mu\lambda_l)v'_l(n) = (1 - \mu\lambda_l)^{(n+1)}v'_l(0). \quad (18)$$

Eq. (18) reveals the following necessary condition for the convergence of the filter coefficients of the nodes of the network to the optimal solution:

$$|1 - \mu\lambda_l| < 1, \quad 1 \leq l \leq LN. \quad (19)$$

This means that the filter coefficients of the nodes will converge, which provides system stability in practice, only if all of the eigenvalues of the matrix \mathbf{R}_B , λ_l , fulfill that

$$(1 - \mu\operatorname{Re}(\lambda_l))^2 + (\mu\operatorname{Im}(\lambda_l))^2 < 1. \quad (20)$$

The condition given by (20) will not hold if any $\operatorname{Re}(\lambda_l) < 0$, independently of the value of μ . Therefore, if any eigenvalue of the matrix \mathbf{R}_B has a negative real part, the system will not be stable. If all of the eigenvalues of \mathbf{R}_B have non-negative real parts, it does not automatically imply that the system will converge since the step-size μ , which must be a positive constant, must simultaneously fulfill the following constraints for all of the eigenvalues:

$$0 < \mu < \frac{2}{\operatorname{Re}(\lambda_l)}, \quad (21)$$

$$0 < \mu < \frac{1}{|\operatorname{Im}(\lambda_l)|}, \quad (22)$$

$$\mu < \frac{2\operatorname{Re}(\lambda_l)}{|\lambda_l|^2}. \quad (23)$$

The above conditions depend on many factors, such as algorithm parameters (value of constant μ , effort factors in \mathbf{B}) and the features of the room and the signals (included in the values of the matrix \mathbf{R}). If unit-variance white noise is used as the reference noise signal, the values of the matrix \mathbf{R} would depend only on the physical configuration of the system, (i.e., on the properties of the acoustic paths), because $R_{xx}(l) = \delta(l)$ and hence $R_{xf_{rm,cm}}(l) = s_{rm}(l) * s_{cm}(-l)$.

Similarly, if the input signal follows an MA (Moving Average) mode, i.e., $x(n) = n_o(n) * f(n)$, with $n_o(n)$ being unit-variance white noise and $f(n)$ the impulse response of a filter, then $R_{xx}(l) = f(l) * f(-l)$. Then, the values in the matrix \mathbf{R} can be easily computed as: $R_{xf_{rm,cm}}(l) = f(l) * f(-l) * s_{rm}(l) * s_{cm}(-l) = s_{rm}^f(l) * s_{cm}^f(-l)$, where $s_{rm}^f(l) = f(l) * s_{rm}(l)$. These values depend only on the acoustic system (and the MA model) and represent an equivalent system whose input is unit-variance white noise, and all of the acoustic paths of the system have been replaced by the result of their convolutions with the filter $f(n)$.

On the other hand, when $\mathbf{R}_B = \mathbf{R}$ ($\beta_l = 0, \forall l$), finding an eigenvalue such that $\operatorname{Re}(\lambda_l) < 0$ implies that the algorithm would never converge independently from the choice of step-size μ . However, this circumstance can be avoided using appropriate values of the effort factor $\beta_l > 0$ together with an appropriate choice of the step-size parameter, at the expense of decreasing the final noise cancellation levels. To illustrate this fact, let us consider $\beta_l = \beta > 0, \forall l$, simplifying the Eq. (14) to:

$$\mathbf{v}(n+1) = ((1 - \mu\beta)\mathbf{I} - \mu\mathbf{R})\mathbf{v}(n). \quad (24)$$

From here, new conditions can be derived for the eigenvalues of the new matrix \mathbf{R}_B . If the eigenvalues of \mathbf{R} are named as λ_l^R , we would have:

$$|1 - \mu\beta - \mu\lambda_l^R| < 1 \Rightarrow |1 - \mu(\beta + \lambda_l^R)| < 1. \quad (25)$$

This would allow that if any eigenvalue λ_l^R has a negative real part, its influence on convergence can be countered by using a suitable value of β . However, even choosing appropriate values for

β such that negative eigenvalues are avoided, the following conditions on μ should also hold:

$$\mu < \frac{2}{\beta + \operatorname{Re}(\lambda_l^R)}, \quad (26)$$

$$\mu < \frac{1}{|\operatorname{Im}(\lambda_l^R)|}, \quad (27)$$

$$\mu < \frac{2(\beta + \operatorname{Re}(\lambda_l^R))}{\beta^2 + 2\beta\operatorname{Re}(\lambda_l^R) + |\lambda_l^R|^2}. \quad (28)$$

In a non-collaborative network ($c_{kj} = 0, \forall k \neq j$) and the hypothetical case where all of the acoustic paths are identical, which means that all of the acoustic paths s_{rr} are equal $\forall r$, and all of the crossed acoustic paths are also equal, $s_{rc} = s_{cr}$ for $r \neq c$, matrix \mathbf{R} will be symmetric and positive definite. Therefore, all of its eigenvalues are real and nonnegative. This means that it is always possible to find a value of μ such that the non-collaborative distributed ANC system is theoretically stable. Eigenvalues close to zero might appear, which (in practice) could deteriorate the robustness and convergence of the adaptive algorithm. However, in this case, a small value of β could be easily tuned to improve the algorithm's convergence. Unfortunately this hypothetical case is not given in practice and the stability of non-collaborative networks needs to be checked. The above results apply when the signals and acoustic paths of the ANC systems match the stationary model considered to build the matrix \mathbf{R} or \mathbf{R}_B . They do not apply in transient state.

3. Criteria for assessing network stability

The stability of an ANC network depends (among others) on factors like the delay between signals arriving at each sensor or the relative energies of these signals [28,29]. However, these factors cannot independently determine the stability of the whole network. Therefore, the study of the stability must be addressed from the eigenvalues of the \mathbf{R}_B matrix [15], which gathers all of the acoustic system information and the rules of collaboration if they exist. The properties of these eigenvalues are used to determine the convergence of the adaptive algorithm (FxLMS), which is running at each node, and consequently to assess ANC network convergence. The computation of all of the eigenvalues can be performed using a state-of-the-art function, like the *eig* function of Matlab [30], which uses an optimized version of the LAPACK library [31] and is able to take advantage of the multiple cores of a modern computer. However, the computation of all of the eigenvalues of a matrix using functions like *eig* is computationally very demanding. The main computational cost for obtaining the eigenvalues (without eigenvectors) using the *eig* function is due to the reduction of the matrix to upper Hessenberg form [27], which is $O(\frac{10M^3}{3})$ for a matrix in $\mathbb{R}^{M \times M}$.

The computational cost is especially relevant when the stability of several collaboration set-ups of a given network needs to be tested. Depending on the size of the network (number of nodes), the number of different configurations may be very large. This happens when we are searching for a collaboration scheme so that an unstable non-collaborative network becomes stable by collaboration among nodes. For each possible collaboration set-up, a different matrix \mathbf{R}_B will arise. If the eigenvalues of many different matrices \mathbf{R}_B have to be computed, the computational cost will become unaffordable. Hence, other methods for checking the properties of the eigenvalues of \mathbf{R}_B related to the stability of the ANC system must be tried and tested.

3.1. Stability condition based on the Gershgorin Circle Theorem

There are well-known theorems that provide information about the location of the eigenvalues of a matrix on the complex plane without computing them. The Gershgorin Circle Theorem [27] is possibly the best known of these theorems. A sufficient criterion to guarantee that the eigenvalues of a matrix have positive real parts without needing to compute them can be derived from this theorem. Thus, it can be shown that the eigenvalues will have positive real parts if the moduli of the diagonal elements are higher than the sum of the moduli of the rest of the elements in the same row. This means, given $Rb_{xf_{ij}}$ as the (i, j) element of the matrix \mathbf{R}_B , that:

$$|Rb_{xf_{ii}}| > \sum_{j \neq i} |Rb_{xf_{ij}}|, \quad \forall i. \quad (29)$$

The computational cost of checking the condition given by (29) is only $M \cdot (M - 2)$ additions for a matrix in $\mathbb{R}^{M \times M}$. However, the condition given by Eq. (29) turns out to be too restrictive in practice [13,15]. As will be shown in Section 4, this condition can lead to having to consider most networks as unstable (even if they are stable), which means most system matrices will not fulfill the condition (29), thus providing many False Negatives (FN).

However, it must be stressed that if Eq. 29 is verified, this ensures system stability in practice (i.e., it cannot give FP). Furthermore, it is always possible to lead any system to fulfill Eq. (29) by choosing a suitable effort factor ($\beta_i \geq 0$) for each node. Since \mathbf{R}_B differs from \mathbf{R} only in the diagonal elements and all of the elements of the diagonal are positive, it holds that:

$$|Rb_{xf_{ii}}| = \beta_i + |R_{xf_{ii}}|, \quad (30)$$

and, therefore, condition (29) holds if:

$$\beta_i > \sum_{j \neq i} |R_{xf_{ij}}| - |R_{xf_{ii}}|, \quad \forall i. \quad (31)$$

Although these values of β_i can provide stable systems, it is at the expense of degradation in noise cancellation performance and give ANC systems without practical usefulness in most cases.

The condition given by Eq. (29) can be alternatively obtained from Eq. (14) considering that

$$\lim_{n \rightarrow \infty} (\mathbf{I} - \mu \mathbf{R}_B)^n = 0, \quad (32)$$

for an ANC system to converge to the optimum filter coefficients. Expression (32) holds if

$$\|\mathbf{I} - \mu \mathbf{R}_B\| < 1, \quad (33)$$

for any consistent matrix norm (denoted by $\|\cdot\|$). Specifically, the use of the infinity norm, $\|\mathbf{R}_B\| = \max_i \sum_j |Rb_{xf_{ij}}|$ leads to Eq. (29). To prove this, let us consider that i -th row has the maximum sum of the modules of the row elements. In this case Eq. (33) holds if:

$$|1 - \mu Rb_{xf_{ii}}| + \mu \sum_{j \neq i} |Rb_{xf_{ij}}| < 1, \quad (34)$$

which holds when:

$$|Rb_{xf_{ii}}| > \sum_{j \neq i} |Rb_{xf_{ij}}|. \quad (35)$$

3.2. Stability condition based on the computation of the inertia

The stability criterion given by Eq. (29) is computationally very efficient. However, this criterion is far too restrictive and marks most network configurations as likely unstable, even if they are stable. This means a high number of FN outcomes. Thus, a criterion that minimizes the FN outcomes but that is computationally

less expensive than the computation of all of the eigenvalues of \mathbf{R}_B is needed to be useful in practice. It is important to take advantage of the structure of the matrix \mathbf{R}_B , which is a block matrix with Toeplitz blocks. These matrices are usually referred to as Toeplitz-block matrices.

As mentioned above, a distributed ANC network running the FxLMS algorithm will converge to the optimum controllers when all of the eigenvalues of \mathbf{R}_B have positive real parts and the step-size μ is suitably selected. This implies in practice that the system will perform properly and remain stable. Therefore, our goal is to identify the systems where all of the eigenvalues of \mathbf{R}_B have positive real parts, because these systems can be made stable (by selecting an appropriate μ).

The so-called inertia theorems can be used to check whether the real parts of all of the eigenvalues of the system matrix are positive. The inertia of a matrix \mathbf{A} , denoted by $\mathbf{In}(\mathbf{A})$, is defined as the triplet: $(\pi(\mathbf{A}), \nu(\mathbf{A}), \delta(\mathbf{A}))$ [32], where $\pi(\mathbf{A})$, $\nu(\mathbf{A})$ and $\delta(\mathbf{A})$ are, respectively, the number of eigenvalues of \mathbf{A} with positive, negative, and zero real parts, counting multiplicities. There are methods for computing the inertia (without computing the eigenvalues) for symmetric and non-symmetric matrices. However, the methods described in [32] have a computational cost $O(M^3)$ for matrices of size $(M \times M)$, like the computation of all the eigenvalues.

Since we are interested in faster methods, this requires taking advantage of the Toeplitz-block structure of the matrix \mathbf{R}_B . Toeplitz-block matrices belong to a wider class of matrices that is described as the set of matrices having **displacement structure**. These matrices are defined in [33] as matrices $\mathbf{A} \in \mathbb{C}^{M \times M}$ that can be written as:

$$\mathbf{AZ}_M - \mathbf{Z}_M \mathbf{A} = \mathbf{G}_M (\mathbf{H}_M)^T, \quad (36)$$

where \mathbf{Z}_M is denoted as

$$\mathbf{Z}_M = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad (37)$$

and \mathbf{G}_M and $\mathbf{H}_M \in \mathbb{C}^{M \times \alpha}$ are called the generator matrices. α is called the displacement rank of \mathbf{A} and is small compared to M . A Toeplitz-block matrix with α blocks has a displacement rank of at most 2α [33].

A fast method ($O(\alpha M^2)$) for computing the inertia of a Hermitian Toeplitz-block matrix (hence, with real eigenvalues) is described in [33] (Algorithm 2.3 in [33]). This algorithm allows computing the number of eigenvalues of a Hermitian Toeplitz-block matrix \mathbf{A} that are greater than, equal to, or less than a given real number κ . Algorithm in Appendix B gives a pseudo-code description of the inertia algorithm for $\kappa = 0$, which incorporates an improvement described below. A demo version of the code can be downloaded from http://personales.upv.es/vmgarcia/Toeplitz_inertia.zip.

Unfortunately, matrix \mathbf{R}_B is not Hermitian in general. However, the interval containing the real parts of the eigenvalues of a matrix can be approximated using Bromwich's inequality [34], which states that the real parts of the eigenvalues of a non-symmetric matrix \mathbf{A} are contained in the interval formed by the largest and smallest eigenvalue of the symmetric part of \mathbf{A} , which is defined as: $\mathbf{A}^S = 0.5 * (\mathbf{A} + \mathbf{A}^T)$. If the matrix \mathbf{R}_B is Toeplitz-block, its symmetric part (\mathbf{R}_B^S) will also be.

The Bromwich inequality can be combined with the inertia algorithm proposed in [33] (for $\kappa = 0$) to develop an approximate fast method for assessing the sign of the eigenvalues of \mathbf{R}_B^S . This method can be used to approximately test the sign of the real parts

of the eigenvalues of \mathbf{R}_B . If all of the eigenvalues of \mathbf{R}_B^S are positive, then the real parts of all of the eigenvalues of \mathbf{R}_B are also positive. This result is assured thanks to the Bromwich inequality and therefore this method will not provide false positives in the case of ANC networks. On the other hand, if any of the eigenvalues of \mathbf{R}_B^S were negative, then it could not be stated that the system was unstable since it might still happen that the real parts of all of the eigenvalues of \mathbf{R}_B were positive. However, it will be shown in Section 4 that the rate of successful assessment of stability/instability provided by this method is good enough for practical use.

This method requires the computation of the symmetric part matrix (\mathbf{R}_B^S) and the computation of the generator matrices. The algorithm for computing the generator matrices is included in a downloadable version from http://personales.upv.es/vmgarcia/Toeplitz_inertia.zip. Algorithm was implemented as a mex file to speed up its execution in Matlab [30]. Furthermore, the algorithm can be accelerated by taking advantage of the special features of the problem. In the description in [33], when Algorithm is applied to a ($M \times M$) matrix \mathbf{R}_B , it computes a vector \mathbf{q} such that the number of negative components in \mathbf{q} indicates the number of negative eigenvalues of \mathbf{R}_B . The vector \mathbf{q} is generated within a loop with M iterations, where a component of \mathbf{q} is generated at each iteration. In our application, a single negative value of \mathbf{q} is enough to classify the ANC system as unstable. Therefore, Algorithm has been modified so that it stops as soon as a negative value of \mathbf{q} arises. This modification does not have any effect when the system is stable (because the vector \mathbf{q} has to be completely computed), but the number of iterations of the loop (and the computing time) can be greatly reduced in many cases where the system is being classified as unstable. The computational cost of Algorithm depends on the number of iterations of its main loop (line 8 of Algorithm) that are carried out. Each iteration adds $O(\alpha M)$ flops to the computational cost. The number of iterations depends, among other factors, on whether the system is stable or not. If the system is stable, which means that all real parts of the eigenvalues of the system matrix are positive, the loop will iterate M times, therefore the cost will be the one of the original algorithm proposed in [33], i.e. $O(\alpha M^2)$. If the system is unstable, the cost can be as small as $O(\alpha M)$ (if a negative value in the \mathbf{q} vector arises in the first iteration).

3.3. Discussion about alternative methods

Iterative methods (such as Arnoldi, Lanczos,...) can alternatively be used to compute the eigenvalues of interest without computing all of the eigenvalues of \mathbf{R}_B [27]. There are several well-known libraries, of which Arpack [35] is the best known. The *eigs* function of Matlab [30] makes internally use of the *znaupd* Arpack function. Iterative methods are used to compute only a few eigenvalues. The *eigs* function can be configured to look for the eigenvalues with the smallest real parts (which are the eigenvalues that we are interested in). Generally speaking, these methods can be very fast (for large matrices) and the Toeplitz-block structure can be used to accelerate their computation. Unfortunately, the *eigs* function (and the underlying Arpack function *znaupd*) often fails to converge when it is applied to some of the \mathbf{R}_B matrices arising in ANC networks. These failures occur when the *eigs* function is configured to find the eigenvalues with the smallest real parts, and, at the same time, the \mathbf{R}_B matrix has a cluster of several (many) eigenvalues very close to zero. It is possible to obtain convergence by modifying some *eigs* input parameters, i.e., increasing the number of eigenvalues to be found or increasing the dimension of the subspace used in inner computations. However, when either of these two approaches is applied, the computational cost of *eigs* becomes higher than the computational cost of the *eig* function. Therefore, the use of *eigs* was discarded.

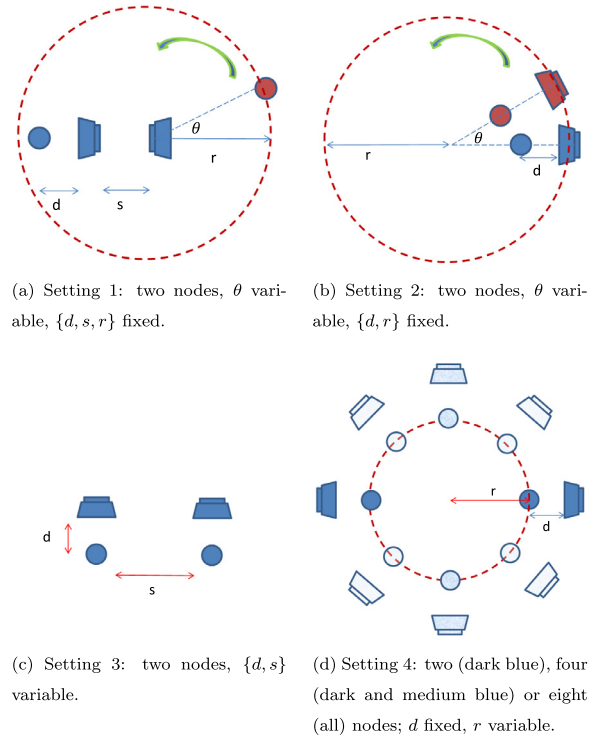


Fig. 3. Different ANC network scenarios working in non-collaborative mode. Each of the settings provides several scenarios. The blue elements are fixed and the red elements change. Setting 4 displays scenarios with different number of nodes.

4. Results

In this section, different methods that are available to assess the stability of a distributed ANC are evaluated, these are: the computation of all of the eigenvalues, the criterion based on Gershgorin's theorem given by (29), and the inertia method proposed in 3.2. These methods have been tested for ANC networks using synthetic room impulse responses of a room with size $9.13 \text{ m} \times 4.48 \text{ m} \times 2.64 \text{ m}$ and different reverberation values. A total of 166320 scenarios were studied. These were obtained by generating room impulse responses using the images method [36] with the software reported in [37] and available in [38]. The acoustic systems were ANC networks of 2, 4, and 8 nodes located at different positions inside the room. White noise signal filtered through different filters (low-pass, band-pass, and high-pass) is considered. The stability analysis only requires knowledge of the estimated secondary paths and perfect estimation has been assumed. The number of coefficients of the generated room impulse responses of the secondary paths was 250.

The stability of each scenario was assessed by computing all of the eigenvalues of the system matrix using the Matlab *eig* function. As mentioned above, if the real part of all of the eigenvalues is positive, then the system is considered stable; thus this assessment is considered to be the ground truth throughout this section. The system stability was also estimated using the condition given by (29) and using the proposed inertia method. As previously mentioned, both methods can not give false positives, which means an FP rate of 0%. However, both methods can fail by marking systems that are stable as unstable (based on exact eigenvalues). As a matter of fact, in practice, the method derived from the Gershgorin theorem becomes useless for assessing the stability of wide-band ANC systems, because it nearly always marks the system as unstable in all the simulated scenarios, which means an unacceptable

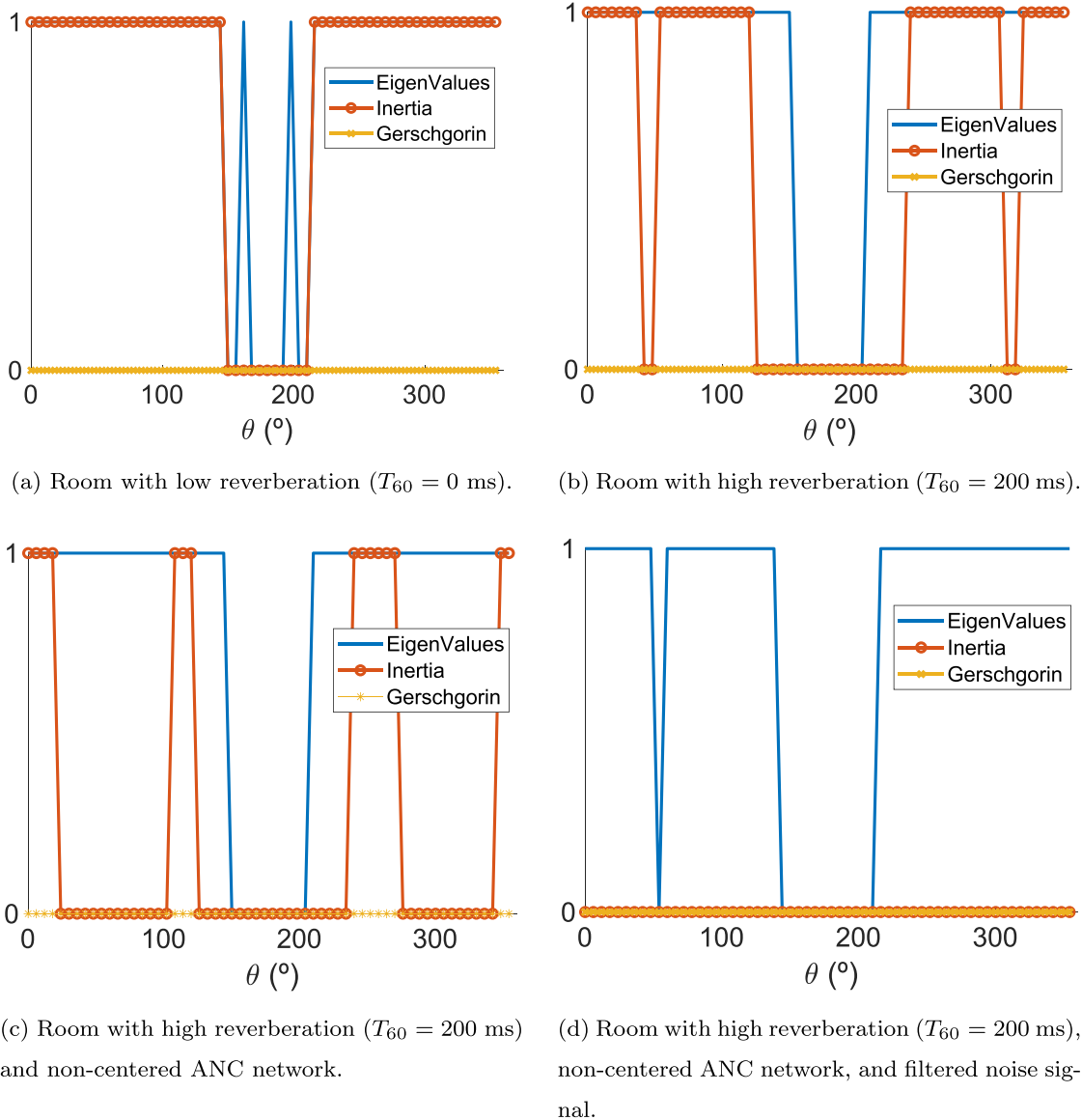


Fig. 4. Stability assessment from Inertia and Gerschgorin's criterion for different configurations of an ANC network composed of two nodes (Setting 1 with $\{d, s, r\} = \{20, 40, 50\}$ cm) working in non-collaborative mode.

FN ratio. Its outcomes coincide with the direct calculation of the eigenvalues only in 1.51% of the simulated scenarios (almost only those with negative real part of any eigenvalue), while the inertia method coincides in 57.88% of them. This means that the FN ratio is $\approx 100\%$ for the Gershgorin criterion and $\approx 43.8\%$ for the inertia criterion, for the simulated scenarios.

Fig. 3 illustrates some of the scenarios, which can be generated selecting different values of the parameters: d , s , and r . Furthermore, Fig. 3(a) can provide different scenarios by modifying the angle of one of the microphones (θ), and Fig. 3(b) generates scenarios by modifying the angle of one node (speaker and microphone). Fig. 3(d) considers configurations of 2, 4, and 8 nodes with constant angular distribution. The radius, r , of separation between nodes is the single parameter that is changing in this scenario.

The sampling frequency was fixed to $f_s = 1000$ Hz in all cases. The stability assessment of the ANC networks was also calculated for different values of L , the number of coefficients of the adaptive filter at each node. This way, e.g., the assessment of

stability for the network given by Fig. 3(a) ($d = 20$ cm, $s = 40$ cm and $r = 50$ cm) can be displayed as a function of the angle of the mobile microphone for the particular case of $L = 180$. Note that we are selecting values of d , r and s that provide stable and unstable systems depending on θ . The assessments of the three methods are displayed in Fig. 4(a) (1 indicates a stable system and 0 indicates an unstable system) for a white noise signal in a low reverberation room ($T_{60} = 0$ ms, i.e, similar conditions to free space propagation). It can be seen that the proposed inertia method gives incorrect assessments for only two scenarios ($\theta = 162^\circ$ and $\theta = 198^\circ$), while the condition derived from the Gerschgorin theorem marks all of the configurations as unstable.

In the case of rooms with higher reverberation time, the system stability depends on the relative positions of the nodes and also on their placements within the room. However, the ability of the methods to evaluate the stability of the network remains similar. The results for the same room with a reverberation time of $T_{60} = 200$ ms are shown in Fig. 4(b). Again, the inertia method

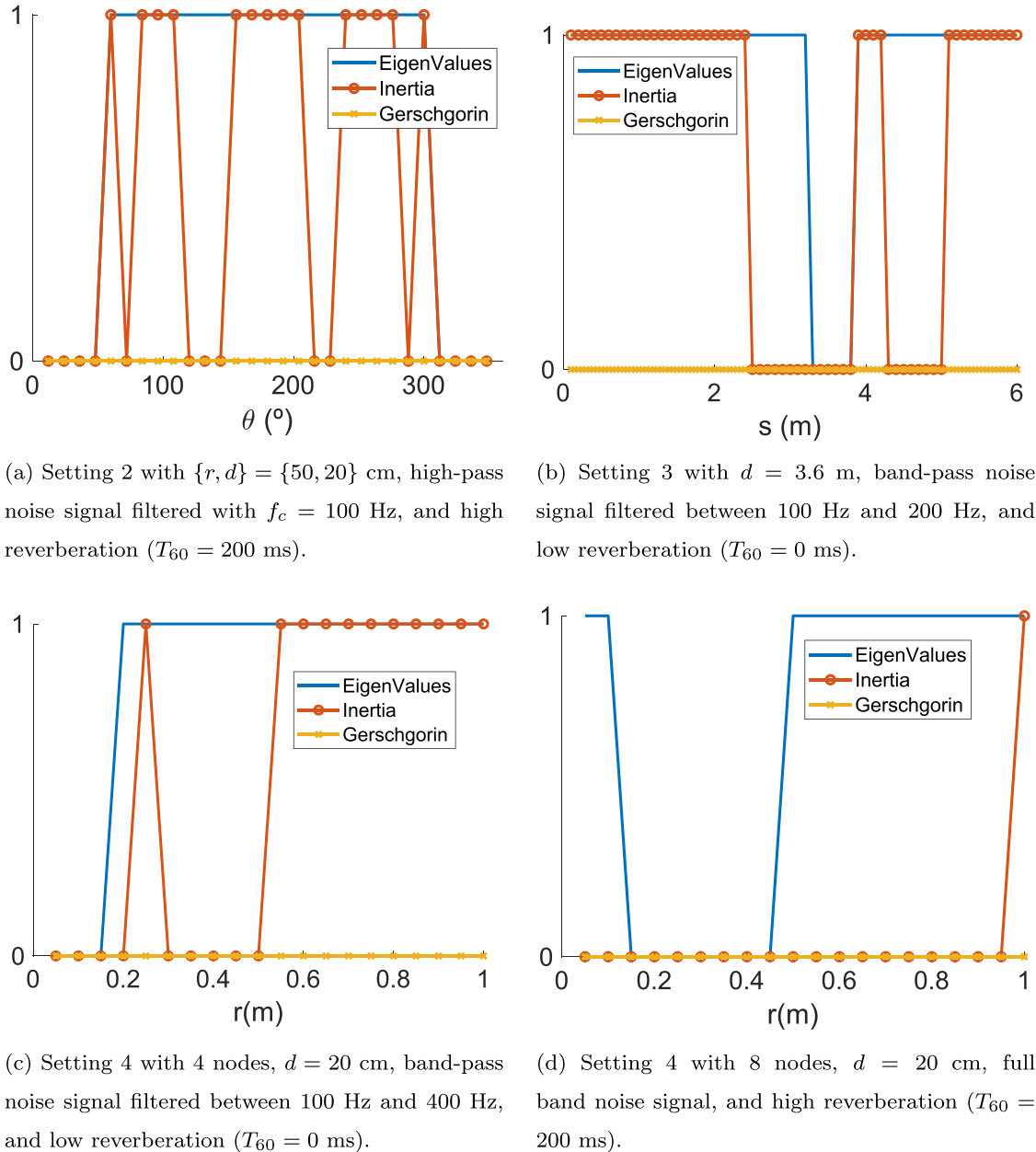


Fig. 5. Stability assessment from Inertia and Gerschgorin's criterion for different configurations of an ANC network working in non-collaborative mode.

outcomes are closer to the direct calculation of eigenvalues than Gerschgorin's criterion. In this example, the network is located in the center of the room. As a consequence, symmetries can be observed in the results. Fig. 4(c) shows the results of the same scenario but displacing the network away from the center of the room. The inertia method does not provide symmetrical results in this case and exhibits worse results than Fig. 4(b). However, this method is still much more accurate than Gerschgorin's criterion.

The inertia method shows different accuracy in different scenarios when it is compared with the direct calculation of the eigenvalues, depending on the features of the acoustical system and on the reference signal. However, there is not a clear pattern. In a few cases, the inertia method can exhibit a behavior that is as restrictive as Gerschgorin's criterion, i.e., the configuration shown in Fig. 4(d). In this case, the reference signal has been high-pass-filtered with a cutoff frequency of $f_c = 400$ Hz. However, the

Table 1

Theoretical computational cost of the considered methods to check the positiveness of the real part of the eigenvalues in flops. $M = L\alpha$, where α is the number of nodes and L is the length of the adaptive filters.

Method	Flops
<i>eig</i> function	$\frac{10M^2}{3} + \text{lower order terms}$
Inertia method (proposed)	$\alpha M^2 + \text{lower order terms (worst case)}$
Gerschgorin's criterion	$M(M - 2)$

inertia method always shows better performance than Gerschgorin's criterion. Similar results for the rest of the configurations (also with $L = 180$) are shown in Fig. 5.

As mentioned above, the computational costs of Gerschgorin's criterion and the inertia method are small when compared with the computational cost of computing all of the eigenvalues. The theoretical costs of the three methods are shown in Table 1. These

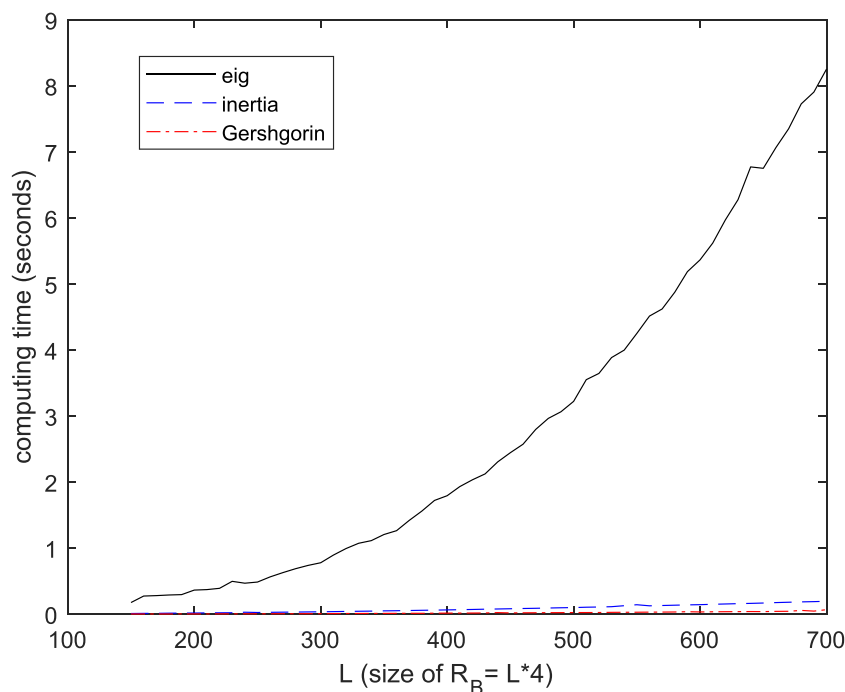


Fig. 6. Computing times of: *eig*, inertia method, and Gershgorin's criterion. A computer equipped with an Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz with 6 cores and 32 GB has been used.

theoretical computational costs must be carefully considered when practical measurements are being carried out because the *eig* function is a parallelized state-of-the-art routine, which takes profit of all the available computing cores. On the other hand, the main loop of the proposed inertia algorithm (line 8 of Algorithm) builds each new iteration using data from the former iteration, therefore this loop cannot be effectively parallelized. Furthermore, the expression given in the Table 1 for the inertia method corresponds to the worst case, when the internal loop is completed. Fig. 6 shows the computing time of the three methods (Matlab's routine *eig* was used to compute all of the eigenvalues) using a computer equipped with an Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz with 6 cores and 32 GB. In this figure, a system with 4 nodes (increasing L from 150 to 700) was chosen.

The inertia method gives a FN ratio as small as 33% in the simulated low reverberation scenarios, which becomes 50% in the high reverberation scenarios, without any FP outcome (FP=0%). Furthermore, it requires an affordable computing time for most cases as compared with the direct calculation of the eigenvalues. Thus, the proposed inertia method provides a good balance between accuracy and computing time. This makes the inertia method suitable for the exhaustive assessment of different ANC network configurations. Furthermore, it can be used to search for the best rules of collaboration (values of c_{km} in Eq. (5)) that keep the network stable. It can be shown that if all c_{km} coefficients are set to 1, the system will remain stable. In the other case, if the only c_{km} values set to 1 are those where $k = m$, then the system has no collaborations and it becomes unstable in many cases. Thus, it is advisable to perform a search for the rules of collaboration with the smallest number of collaborations (smallest number of nonzero c_{km} values) among nodes. The proposed inertia method allows this search to be performed within an affordable computing time.

The inertia method can also be used to find the largest adaptive filter length (L) so that the system remains stable. As an example

of this use, Fig. 7 shows the stability assessment of the different methods varying L up to 500.

5. Discussion

The proposed method assesses the stability of the distributed ANC systems, but it does not provide information about the achieved noise cancellation level. The performance of a stable ANC system in terms of noise cancellation level depends on the relation between the reference and noise signals, the step size and the spatial placement of the transducers. On the other hand, stability only depends on the reference signal and the spatial placement of the transducers. The best performance of a distributed system is achieved for a fully collaborative scheme, which is stable since the system matrix is definite positive and coincides with the performance of the equivalent centralized scheme [10]. However, this is at the expense of the highest communication needs. Therefore, looking for a configuration that achieves a given minimum desired performance in terms of cancellation while minimizing collaboration (communications in distributed systems) seems reasonable in practice. The inertia method proposed in this paper allows to efficiently search for a minimum collaboration setup that assures system stability. This collaboration setup gives excellent cancellation results in practice (close to the fully collaborative network) because the nodes that do not collaborate should have very little influence on the final residual noise levels. Moreover, the fully minimization of collaboration is not always achieved since the FN rate is not zero in the proposed algorithm. However, the obtained stable systems exhibit slightly better performance in terms of residual noise levels than the corresponding stable system with minimum collaboration because collaboration cannot worsen the performance in terms of cancellation [10].

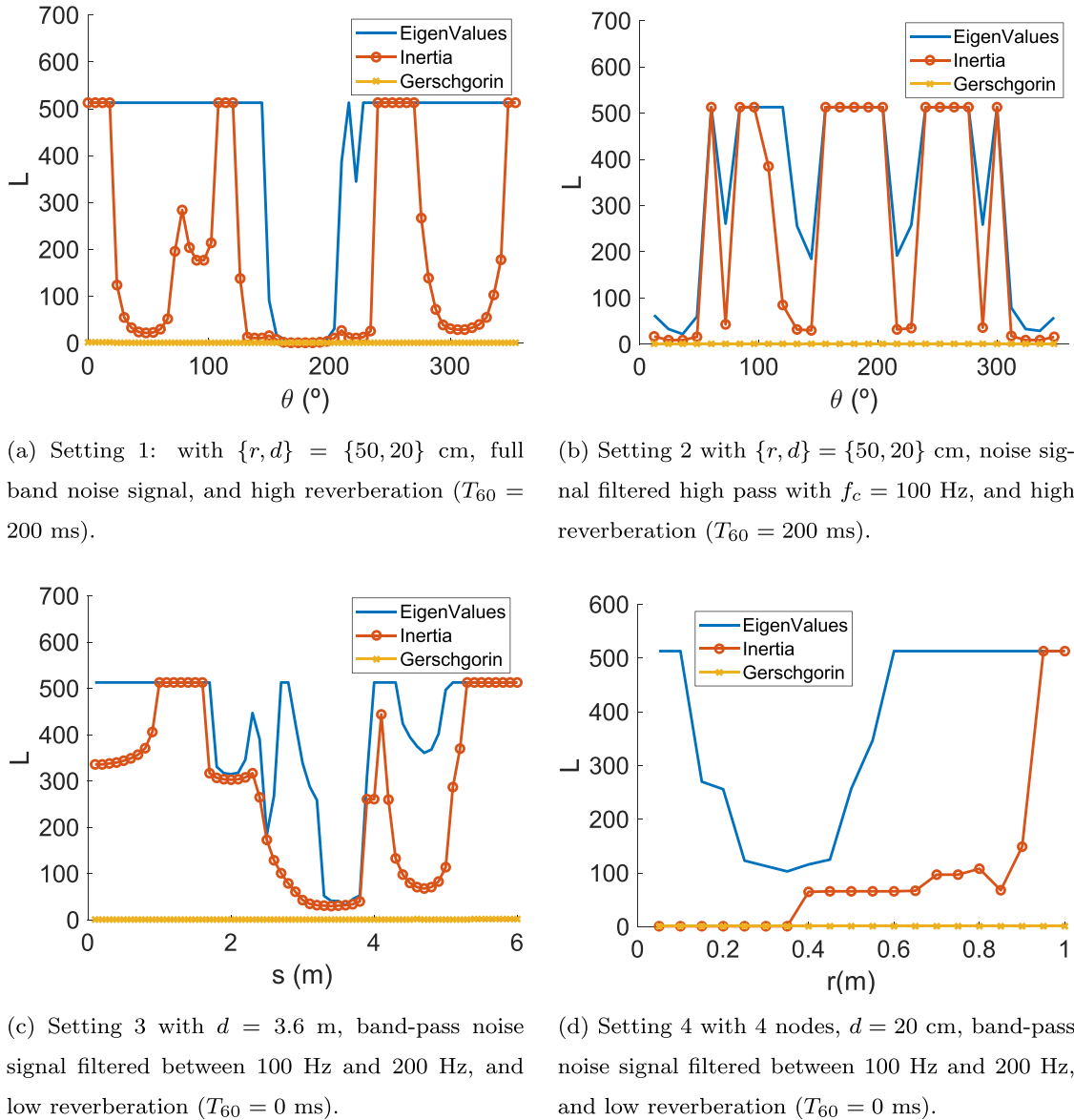


Fig. 7. Maximum adaptive filter length (L) that guarantees stability for different settings of an ANC network working in non-collaborative mode.

6. Conclusions

This paper has addressed the stability assessment of an ANC network of N single-channel nodes working in non fully-collaborative mode. In this kind of ANC network, each node tries to minimize noise (which is a wide-band signal) by executing the FxLMS adaptive algorithm. Each node could reach a stable steady state (independently of the other nodes) by choosing an appropriate convergence step. However, since the network is acoustically coupled, the stability of the network depends on the interaction among nodes.

The convergence analysis of the multichannel FxLMS running on the network reveals that the stability of the system can be assessed from the eigenvalues of the system matrix, which is denoted by \mathbf{R}_B . The size of this matrix is proportional to the number of nodes of the network and the length of the adaptive filters (L). This matrix is generated using the data of the correlation of the noise at the node microphones with the reference signal

noise filtered through the different network acoustic paths. Matrix \mathbf{R}_B of a given network can be thought of as the assembly of $(N \times N)$ submatrices, each one of $(L \times L)$ size. The blocks on the diagonal of the system matrix represent the self-coupling between the speaker of each node with its microphone. It should be noted that the proposed algorithm needs the \mathbf{R} matrices as inputs, which, in practice, must be approximated by estimators such as time averages.

The network stability assessment of an ANC network is simpler in case of tonal noise. In this case, the size of the system matrix is just given by the number of nodes, and the coefficient (i, j) of \mathbf{R}_B represents the influence of the j -th node on the i -th node [15]. In such case, the moderate size of the system matrix makes easier the computation of all of its eigenvalues and, consequently, makes easier the stability assessment. However in the case of wide-band noise signals, the computation becomes far more complex due to the larger size of \mathbf{R}_B and the diversity of scenarios and noise signals.

The stability of ANC networks can be assessed using standard routines for the computation of all of the eigenvalues of \mathbf{R}_B , but the computational cost of this procedure (for large L and N) is too high for practical use. On the other hand, there are theorems that provide clues or boundaries about the location of the eigenvalues, such as the Gershgorin Theorem. Conditions on the system matrix derived from them can assure that its eigenvalues remain within a given region on the complex plane and thus can provide network stability. Therefore the condition given by Eq. (29) can be derived from the Gershgorin Theorem and can be computed very fast. However, the results show that this condition is very restrictive for most of the system matrices, (excepting those belonging to ANC systems working at a single frequency [11]) and, consequently, there are many stable scenarios of ANC networks that do not fulfill this condition in practice.

In this paper, we have proposed an alternative inertia method that can be used to estimate if the real part of the eigenvalues of \mathbf{R}_B are positive and to assess the stability of ANC networks. The computation of this method has been optimized and takes advantage of the Toeplitz-block structure of the \mathbf{R}_B matrix. This method is fast enough for practical use and is far more accurate than the condition derived from the Gershgorin Theorem. The results show that the proposed inertia method cannot give FP assessments and provides much less FN assessments than the Gershgorin condition. Therefore, this method can also be used to efficiently find stable ANC network configurations and collaboration rules.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Miguel Ferrer: Conceptualization, Methodology, Formal analysis, Software, Writing – review & editing. **Víctor M. García-Mollá:** Formal analysis, Software, Conceptualization, Methodology, Writing – review & editing. **Antonio M. Vidal-Maciá:** Formal analysis, Conceptualization, Methodology, Writing – review & editing. **Maria de Diego:** Conceptualization, Methodology, Writing – review & editing. **Alberto Gonzalez:** Conceptualization, Methodology, Formal analysis, Funding acquisition, Writing – review & editing.

Data availability

Data will be made available on request.

Appendix A. Computation of the matrix R

Matrix \mathbf{R} is calculated from $(L \times L)$ submatrices $\mathbf{R}_{x_{f_{rm},c_m}}$. Each element (f, c) of the $\mathbf{R}_{x_{f_{rm},c_m}}$ submatrix is computed as $E\{x_{f_{rm}}(n + f - 1)x_{c_m}(n + c - 1)\}$ which becomes $E\{x_{f_{rm}}(n + |f - c|)x_{c_m}(n)\} = R_{x_{f_{rm},c_m}}(|f - c|)$, for stationary signals. Therefore, the elements of this matrix are the values of the cross-correlation of the signals $x_{f_{rm}}(n)$ and $x_{c_m}(n)$, which are the results of filtering the reference signal $x(n)$ by the acoustic paths $s_{rm}(n)$ and $s_{cm}(n)$ respectively. Consequently, this cross-correlation can be written as a function of the reference signal self-correlation as:

$$\mathbf{R}_{x_{f_{rm},c_m}}(l) = \mathbf{R}_{xx}(l) * s_{rm}(l) * s_{cm}(-l), \quad (\text{A.1})$$

where the symbol $*$ represents the linear convolution.

Appendix B. Inertia algorithm for the stability assessment of ANC networks

Algorithm 1 Algorithm 2.3 from [33] adapted for stability assessment

```

1: Input:  $\mathbf{A}, \mathbf{G}, \mathbf{H}$ 
2: Output: out
3: Given a Hermitian Toeplitz Matrix  $\mathbf{A} \in \mathbf{C}^{M \times M}$  and its generators  $\mathbf{G}$  and  $\mathbf{H} \in \mathbf{C}^{M \times \alpha}$ , this function computes a real variable out such that out = 1 if all the eigenvalues of  $\mathbf{A}$  are positive and out = 0 if there is at least one negative eigenvalue of  $\mathbf{A}$ .  $\alpha$  is the number of nodes and  $M = L\alpha$ , where  $L$  is the length of the adaptive filters.
4:  $q = \text{zeros}(M, 1)$ ;
5:  $q(1) = A(1, 1)$ ;
6:  $w(1, 1) = A(1, 2) / q(1)$ ;
7:  $f(1, 1 : \alpha) = G(1, 1 : \alpha)' ./ q(1)$ ;
8:  $k = 1$ ;
9: while  $q(k) \geq 0$  and  $k < M$  do
10:    $k = k + 1$ ;
11:    $fn = \text{zeros}(k, \alpha)$ ;
12:    $vmm1 = A(1:k-1, k-1)$ ;
13:    $q(k) = A(k, k) - vmm1' * w(1:k-1, k-1)$ ;
14:    $y(1:k, k) = [w(1:k-1, k-1); -1]$ ;
15:   for  $i = 1$  to  $\alpha$  do
16:      $fn(:, i) = [f(:, i); 0] - (G(k, i) - vmm1' * f(1:k-1, i)) * y(1:M, k) / q(k)$ ;
17:   end for
18:    $w(1:k, k) = [0; w(:, k-1)] - fn(:, 1:\alpha)' * (H(:, 1:k))' * y(1:k, k)$ ;
19:    $f = fn$ ;
20: end while
21: if  $q(k) < 0$  then
22:    $out = 0$ ;
23: else
24:    $out = 1$ ;
25: end if

```

References

- [1] Y. Chu, M. Wu, H. Sun, J. Yang, M. Chen, Some practical acoustic design and typical control strategies for multichannel active noise control, *Applied Sciences* 12 (4) (2022), doi:10.3390/app12042244.
- [2] M. Bouchard, S. Quednau, Multichannel recursive-least-square algorithms and fast-transversal-filter algorithms for active noise control and sound reproduction systems, *IEEE Transactions on Speech and Audio Processing* 8 (5) (2000) 606–618, doi:10.1109/89.861382.
- [3] M. de Diego, A. Gonzalez, Performance evaluation of multichannel adaptive algorithms for local active noise control, *Journal of Sound and Vibration* 244 (4) (2001) 615–634.
- [4] S. Spors, H. Buchner, Efficient massive multichannel active noise control using wave-domain adaptive filtering, in: 3rd International Symposium on Communications, Control and Signal Processing, 2008, pp. 1480–1485, doi:10.1109/ISCCSP.2008.4537461.
- [5] S.M. Kuo, D. Morgan, *Active Noise Control Systems*, John Wiley-Sons, 1996.
- [6] D. Shi, W.-S. Gan, J. He, B. Lam, Practical implementation of multichannel filtered-x least mean square algorithm based on the multiple-parallel-branch with folding architecture for large-scale active noise control, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28 (4) (2020) 940–953, doi:10.1109/TVLSI.2019.2956524.
- [7] K. Jyothi, A. Sahoo, Efficient adaptive strategies over distributed networks, in: *Computing, Electronics and Electrical Technologies (ICCEET)*, 2012 International Conference on, 2012, pp. 798–801, doi:10.1109/ICCEET.2012.6203839.
- [8] C. Lopes, A. Sayed, Incremental adaptive strategies over distributed networks, *IEEE Transactions on Signal Processing* 55 (8) (2007) 4064–4077, doi:10.1109/TSP.2007.896034.
- [9] C. Lopes, A. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis, *IEEE Transactions on Signal Processing* 56 (7) (2008) 3122–3136, doi:10.1109/TSP.2008.917383.
- [10] M. Ferrer, M. de Diego, G. Piero, A. Gonzalez, Active noise control over adaptive distributed networks, *Signal Processing* 107 (2015) 82–95, doi:10.1016/j.sigpro.2014.07.026.
- [11] S. Elliott, I. Stothers, P. Nelson, A multiple error LMS algorithm and its application to the active control of sound and vibration, *IEEE Transactions on*

- Acoustics, Speech and Signal Processing 35 (10) (1987) 1423–1434, doi:10.1109/TASSP.1987.1165044.
- [12] E. Bjarnason, Analysis of the filtered-x lms algorithm, *IEEE Transactions on Speech and Audio Processing* 3 (6) (1995) 504–514, doi:10.1109/89.482218.
- [13] C. Antonanzas-Manuel, M. Ferrer, M. de Diego, A. Gonzalez, Collaborative method based on the acoustical interaction effects on active noise control systems over distributed networks, *Proc. of 42nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2017)*, New Orleans, USA: (2017) 606–610.
- [14] Y. Dong, J. Chen, W. Zhang, Wave-domain active noise control over distributed networks of multi-channel nodes, *Signal Processing* 184 (2021) 108050, doi:10.1016/j.sigpro.2021.108050.
- [15] S. Elliott, C. Boucher, Interaction between multiple feedforward active control systems, *IEEE Transactions on Speech and Audio Processing* 2 (4) (1994) 521–530, doi:10.1109/89.326611.
- [16] G. Zhang, J. Tao, X. Qiu, I.S. Burnett, Decentralized two-channel active noise control for single frequency by shaping matrix eigenvalues, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27 (1) (2019) 44–52, doi:10.1109/TASLP.2018.2869686.
- [17] S. Pradhan, G. Zhang, S. Zhao, K. Niwa, W. Bastiaan Kleijn, On eigenvalue shaping for two-channel decentralized active noise control systems, *Applied Acoustics* 205 (2023) 109260, doi:10.1016/j.apacoust.2023.109260. <https://www.sciencedirect.com/science/article/pii/S0003682X23000580>.
- [18] S. Elliott, *Signal Processing for Active Control*, Academic Press, 2001.
- [19] S. Haykin, *Adaptive Filter Theory*, fourth, Prentice-Hall, Ed., Upper Saddle River, NJ, 2002.
- [20] C. Gong, M. Wu, J. Guo, J. Chen, Z. Zhang, Y. Cao, J. Yang, Statistical analysis of multichannel FxLMS algorithm for narrowband active noise control, *Signal Processing* 200 (2022) 108646, doi:10.1016/j.sigpro.2022.108646. <https://www.sciencedirect.com/science/article/pii/S0165168422001852>.
- [21] J. Chen, J. Yang, A distributed FxLMS algorithm for narrowband active noise control and its convergence analysis, *Journal of Sound and Vibration* 532 (2022) 116986, doi:10.1016/j.jsv.2022.116986. <https://www.sciencedirect.com/science/article/pii/S0022460X22002097>.
- [22] T. Li, S. Lian, S. Zhao, J. Lu, I.S. Burnett, Distributed active noise control based on an augmented diffusion fxlms algorithm, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023) 1–15, doi:10.1109/TASLP.2023.3261742.
- [23] P.A. Nelson, S.J. Elliott, *Active Control of Sound*, Academic Press, 1992.
- [24] S.J. Elliott, K.H. Back, Effort constraints in adaptive feedforward control, *IEEE Signal Processing Letters* 3 (1) (1996) 7–9, doi:10.1109/97.475821.
- [25] D. Bismor, M. Pawelczyk, Stability conditions for the leaky lms algorithm based on control theory analysis, *Archives of Acoustics* 41 (4) (2016) 731–739. <https://acoustics.ippt.pan.pl/index.php/aa/article/view/1753>.
- [26] C. Antonanzas, M. Ferrer, M. de Diego, A. Gonzalez, Control effort strategies for acoustically coupled distributed acoustic nodes, *Wireless Communications and Mobile Computing* 2017 (2017) 1–15.
- [27] G.H. Golub, C.F. Van Loan, *Matrix Computations*, third, The Johns Hopkins University Press, 1996.
- [28] P. Grosdidier, M. Morari, Interaction measures for systems under decentralized control, *Automatica* 22 (3) (1986) 309–319, doi:10.1016/0005-1098(86)90029-4.
- [29] A. Berkhoff, Optimum sensor-actuator distance for decentralized acoustic control, *Journal of the Acoustical Society of America* 110 (2001) 260–266.
- [30] MATLAB, (R2018b), The MathWorks Inc., Natick, Massachusetts, 2018.
- [31] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *LAPACK Users' Guide*, third, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
- [32] B. Datta, *Numerical Methods for Linear Control Systems*, EBSCO ebook academic collection, Elsevier Science, 2004.
- [33] M. Ng, W. Trench, Numerical solution of the eigenvalue problem for hermitian toeplitz matrices, *SIAM Journal on Matrix Analysis and Applications* 10 (1997), doi:10.1137/0610010.
- [34] T.J.I. Bromwich, *Quadratic Forms and Their Classification by Means of Invariant-Factors*, 1906.
- [35] F.M. Gomes, D.C. Sorensen, ARPACK++ - An object-oriented version of ARPACK eigenvalue package., 2000.
- [36] J. Allen, B. D.A, Image method for efficiently simulating small-room acoustics, *Journal Acoustic Society of America* 65 (4) (1979) 943.
- [37] E.A.P. Habets, Room impulse response generator, Technical Report, Technische Universiteit Eindhoven, 2006.
- [38] E.A. Habets, Room impulse response generator, 2010, <https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator>.