JASA ARTICLE

A numerically stable constrained optimal filter design method for multichannel active noise control using dual conic formulation

Yongjie Zhuang and Yangfan Liu[a]

Ray W. Herrick Laboratories, Mechanical Engineering, Purdue University, West Lafayette, Indiana 47907, USA

ABSTRACT:

In practical active noise control (ANC) applications, various constraints are usually required, for example, the disturbance enhancement constraint, the robust stability constraint, and the controller output power constraint. One commonly used approach for designing a constrained ANC filter is to formulate a constrained optimization problem using an H_2/H_∞ framework, which requires significant computational power to solve. Recent work has shown that such an ANC filter design problem can be reformulated into a convex optimization problem and then further reformulated to a cone programming problem to reduce the required computational time by several orders. However, the standard cone programming reformulation procedure leads to a large number of free variables, which, in many applications, can adversely influence the numerical behavior of the optimization algorithm. In the current work, the ANC filter design problem structure is exploited in its dual conic form, which allows an elimination of free variables and can result in a numerically more stable solving process for the filter design problem while keeping the noise control performance unchanged. It is demonstrated that when compared with the reformulation using the standard procedure, the proposed formulation significantly improves its numerical stability and computational efficiency.

https://doi.org/10.1121/10.0014627

(Received 9 June 2022; revised 15 September 2022; accepted 20 September 2022; published online 13 October 2022)

[Editor: Jordan Cheer] Pages: 2169–2182

I. INTRODUCTION

For the past several decades, the computing capability of the digital controller hardware and signal processors (DSP) has improved rapidly, which enables a wider range of possible applications for active noise control (ANC), such as headrests,[1–3] automobiles,[4,5] magnetic resonance imaging (MRI) equipment,[6] ventilation windows,[7,8] etc. One of the challenges of practical ANC products is that multiple types of constraints, such as the robust stability constraint, the controller output power constraint, the disturbance enhancement constraint, etc., need to be considered when designing the ANC filters.[9] For example, the robust stability constraint is needed to ensure stability and performance when manufactural uncertainties or uncertainties in operating environments are considered. And the filter output power constraint is needed to ensure transducers operate in the linear response range or for power consumption considerations.

This article focuses on the constrained optimal filter design where the filter coefficients are not adaptive. For some commercialized products in which the primary path does not change significantly over time, such as ANC headphones, a fixed-coefficient controller is becoming the de facto approach[10] because of its low real-time computational complexity. Fixed-coefficient controllers are especially preferred and when a large number of channels and high filter orders are required, the computational requirement to realize the adaptive control can either exceed the capability of current on-market signal processing hardware or lead to an unrealistic cost for a commercialized product.[11] In some applications, it is also difficult to install error sensors for adaptive algorithms, e.g., in the active mitigation of noises transmitting through open windows.[12–14] The use of adaptive algorithms may be superior compared with nonadaptive algorithms when the environment is time varying. However, the nonadaptive filters can be used for time-varying environments by switching between different pre-trained fixed-coefficient filters such as the selective fixed-filter ANC.[15] Furthermore, when new types of sensors or control sources are being introduced in ANC technologies, optimal nonadaptive filters are usually used to give an evaluation of the best achievable performance (i.e., the nonadaptive optimal control performance), for example, the introduction of virtual sensing technology,[16,17] the use of secondary multipole sources,[18,19] the incorporation into beamforming approach,[20] etc.

Various methods have been developed to design a constrained ANC filter. One of the most commonly used approaches is to introduce a regularization term in the filter coefficients optimization process,[9] which minimizes a weighted sum of the residual error signal power and the power of the ANC filter response. The widely used leaky LMS algorithm relies on this principle but is implemented in an adaptive control process. Besides a trial and error

[a]Electronic mail: yangfan@purdue.edu

0001-4966/2022/152(4)/2169/14/$30.00

tuning, methods to choose an appropriate leak factor have been proposed in previous studies for one particular type of constraint, such as the output power constraint,[21–25] the disturbance enhancement constraint,[26,27] or the robust stability constraint.[28,29] However, the noise control performance may be degraded when multiple types of constraints are required[30,31] because the satisfaction of one constraint usually leads to an over-satisfaction of other constraints. Another commonly used approach that can prevent over-satisfaction of constraints and usually results in better noise control performance[30,31] is to formulate a constrained optimization problem for the ANC filter design using an H_2/H_∞ framework and solve it using the sequential quadratic programming (SQP) or similar algorithms.[32–36] This approach has been further developed and applied to active control of the interior road noises in vehicles.[5] However, the required computational power is significant, especially when the channel count, number of filter coefficients, and number of constraints are large. The long computation time makes this filter design method difficult to implement in practical applications that require filter adaptation. Even when nonadaptive filter structures can be used, such significant computational requirements will likely lead to an unacceptable cost and time in the design phase of a commercialized ANC product as many prototype trials usually need to be performed in the product design phase.

Recently, Zhuang and Liu[30] proposed a convex formulation based on the traditional H_2/H_∞ formulation for the ANC filter design, which can be further reformulated as an equivalent cone programming formulation. Efficient optimization algorithms can then be applied to significantly improve the computational efficiency and reliability. Their results demonstrated that the computational time can be reduced from the order of hours to seconds with various numerical benefits. However, the standard procedure to reformulate the convex formulation into a cone programming formulation[37–39] leads to a large number of free variables,[40,41] which, in many applications, can adversely influence the numerical behavior of the optimization algorithm. The numerical stability of an algorithm, i.e., the ability to converge to an optimal solution when the algorithm is implemented numerically, is an important consideration aside from the computational efficiency. A preliminary investigation by Zhuang and Liu[41] has further demonstrated that different treatments to the free variables in the reformulated cone programming formulation combined with the dual formulation can influence the numerical stability of the optimization algorithm in the solving process. Thus, proper alternative rearrangements and reformulations concerning the dual formulation and free variables can potentially improve the algorithm's numerically stability.

In this article, a more numerically stable and efficient ANC filter design method based on the previous work of Zhuang and Liu[30] was proposed. The convex formulation for the ANC filter design proposed in the previous work[30] is first reformulated into the cone reformulation using the standard procedure. Based on the cone formulation from the standard procedure, the proposed numerically stable cone formulation is derived by exploiting the sparsity pattern and using the dual formulation techniques, which results in a more compact formulation with smaller problem dimensions. The free variables that adversely affect the numerical stability can also be eliminated in the proposed formulation. The results demonstrate that compared with rearranging using standard procedure, the proposed formulation improves the numerical stability and computational efficiency significantly.

II. THEORY

In this section, the general structure of a multichannel ANC system is described along with the convex ANC design problem formulation.[30] The further reformulation into a cone programming form using the standard method is introduced, and the associated numerical instability issues caused by free variables in this formulation are also discussed. Then, the proposed dual form method is described in which the numerical issue can be effectively alleviated by eliminating the free variables and reducing the problem dimension. The formulations in this section are presented in the context of feedforward control, but it is noted that the identical approach can be applied to feedback controller design with the application of an internal model structure.[5,30]

A. Review of the H_2/H_∞ framework and associated convex formulation

The multichannel ANC system structure and associated formulations are adopted from the previous work of Zhuang and Liu[30] and reviewed in this section. The system block diagram is shown in Fig. 1(a), where N_r microphones, N_s loudspeakers, and N_e microphones are used as reference sensors, secondary sources, and error sensors, respectively. In the control diagram, $\vec{\mathbf{x}}$ denotes the noise signals from the primary noise sources at reference sensor locations; $\vec{\mathbf{r}}$ denotes the reference signals measured by the reference sensors, which include sound from the primary and secondary sources; $\vec{\mathbf{y}}$ denotes the output of the controller $\mathbf{H}$; $\vec{\mathbf{d}}_e$ denotes the disturbance signals; $\vec{\mathbf{e}}$ denotes the sum of the primary and secondary sound fields at error sensor locations whose total power is attenuated by the ANC system; and $\mathbf{G}_{s_0}$ denotes the acoustic feedback path matrix (the path where the sound generated by secondary sources propagates to the reference sensor locations). To eliminate the acoustic feedback path effect on the ANC filtering process, an internal model control (IMC) structure is used in the controller $\mathbf{H}$,[5,9,32] where $\hat{\mathbf{G}}_{s_0}$ is a model of the physical feedback path, $\mathbf{G}_{s_0}$. In the current work, $\hat{\mathbf{G}}_{s_0}$ is assumed to be a perfect model when operating in a nominal working condition, and the estimated noise signals, $\hat{\vec{\mathbf{x}}}$, will be the same as the actual primary noise signals, $\vec{\mathbf{x}}$. Thus, this system becomes a standard feedforward system, which is shown in Fig. 1(b). $\mathbf{G}_e$ denotes the acoustical responses matrix of the secondary sources at the error sensor positions, and $\mathbf{W}_x$ denotes the

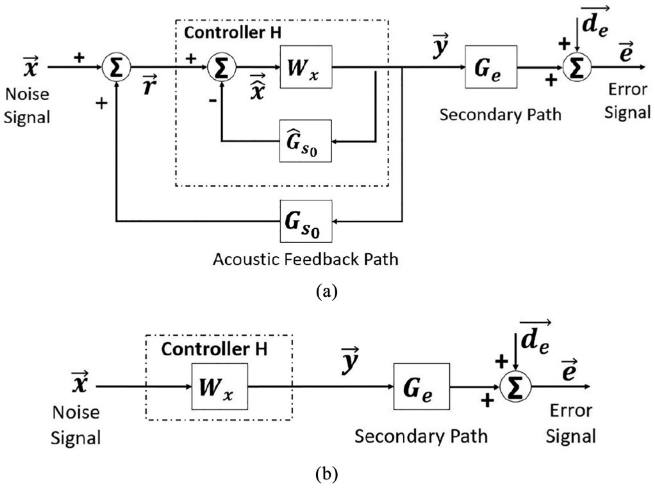


FIG. 1. The block diagram of a MIMO (multiple-input multiple-output) feedforward active noise control system showing (a) with the acoustic feedback path and internal model structure and (b) in a standard feedforward form when assuming internal model structure perfectly cancels the acoustic feedback path effect (Ref. 30).

frequency response matrix of the multichannel ANC filters. In this work, the finite impulse response (FIR) filter structure is used for each input-output channel in $\mathbf{W}_x$; thus, each element of $\mathbf{W}_x$ is a linear function of the FIR filter coefficients, $\mathbf{w}_F$, such that[30]

$$\mathbf{W}_{x_{i,j}}(f) = \vec{\mathbf{F}}_z^T(f)\vec{\mathbf{w}}_{F_{i,j}}, \tag{1}$$

where

$$\vec{\mathbf{F}}_z(f) = \left[1 \quad e^{-j2\pi f(1/f_s)} \quad e^{-j2\pi f(2/f_s)} \quad \cdots \quad e^{-j2\pi f(N_t-1)/f_s}\right]^T,$$

f denotes the frequency, and f_s denotes the sampling frequency. $\mathbf{W}_{x_{i,j}}$ expresses the ith row and jth column of the frequency response matrix, $\mathbf{W}_x$, and T denotes the matrix transpose operation. Each $\vec{\mathbf{w}}_{F_{i,j}}$ is a column vector that includes the N_t ANC FIR filter coefficients associated with the ith output and jth input channels.

The multichannel ANC filter design problem can be formulated as a constrained optimization problem using the convex H_2/H_∞ framework,[30] which is a convexified formulation from the work of Cheer and Elliott,[5]

$$\text{minimize}\ \vec{\mathbf{w}}^T\left(\sum_{k=1}^{N_f}\mathbf{A}_J(f_k)\right)\vec{\mathbf{w}} + \sum_{k=1}^{N_f}\vec{\mathbf{b}}_J^T(f_k)\vec{\mathbf{w}}, \tag{2a}$$

$$\text{subject to}\quad \vec{\mathbf{w}}^T\mathbf{A}_J(f_k)\vec{\mathbf{w}} + \vec{\mathbf{b}}_J^T(f_k)\,\vec{\mathbf{w}} + \tilde{c}_J(f_k) \le 0, \tag{2b}$$

$$\|\vec{\mathbf{F}}_z^T(f_k)\vec{\mathbf{w}}_{F_{i,j}}\|_2 - C_{i,j}(f_k) \le 0, \tag{2c}$$

$$\max\left(\lambda\left(\frac{\mathbf{A}_s + \mathbf{A}_s^H}{2}\right)\right) - (1-\epsilon_s) \le 0, \tag{2d}$$

$$\max(\sigma(\mathbf{A}_s))B(f_k) - 1 \le 0, \tag{2e}$$

for all of the frequency components of interest, f_k, and all of the channels (indexed by i and j). The objective function, Eq. (2a), is the total power of error signals across all of the desired frequencies, where $\vec{\mathbf{w}}$ is a vector constructed by stacking FIR filter coefficients of all of the channel pairs, i.e., stacking $\vec{\mathbf{w}}_{F_{i,j}}$ with all of the i-j combinations into a single vector. Thus, the elements in $\vec{\mathbf{w}}$ are the variables that need to be found by solving the constrained optimization problem. To list more details of the above formulation,

$$\mathbf{A}_J(f_k) = \Re\left(\left(\mathbf{G}_e^H(f_k)\mathbf{G}_e(f_k)\right)\otimes\mathbf{S}_{xx}^T(f_k)\otimes\left(\vec{\mathbf{F}}_z^*(f_k)\vec{\mathbf{F}}_z^T(f_k)\right)\right),$$
$$\vec{\mathbf{b}}_J(f_k) = 2\Re\left(\text{vec}\left(\left(\mathbf{S}_{xd_e}(f_k)\mathbf{G}_e(f_k)\right)\otimes\vec{\mathbf{F}}_z(f_k)\right)\right),$$
$$\vec{\mathbf{w}} = \left[\vec{\mathbf{w}}_{F_{1,1}}^T \ \cdots\ \vec{\mathbf{w}}_{F_{1,N_r}}^T\ \vec{\mathbf{w}}_{F_{2,1}}^T\ \cdots\ \vec{\mathbf{w}}_{F_{N_s,N_r}}^T\right]^T,$$

where H denotes the complex conjugate transpose operation, $\otimes$ denotes the Kronecker product, $*$ denotes the complex conjugate operation, and vec( ) denotes the vectorization operation that converts a matrix to a vector by stacking the columns of this matrix;[42] $\mathbf{S}_{xx}(f_k)$ and $\mathbf{S}_{d_ed_e}(f_k)$ are the cross-spectral density matrices of $\vec{\mathbf{x}}$ and $\vec{\mathbf{d}}_e$, respectively, at frequency f_k; $\mathbf{S}_{xd_e}(f_k)$ is the cross-spectral density matrix between the primary noise signals, $\vec{\mathbf{x}}$, and the disturbance signals, $\vec{\mathbf{d}}_e$, at frequency f_k; and $\Re(\,)$ denotes the real part of a complex number (or matrix).

For constraints, Eq. (2b) is applied to prevent large enhancement at some frequency band (in some frequency bands, the resulting noise may be higher than the original level after active control due to the effort to reduce the noise level in other bands), where

$$\tilde{c}_J(f_k) = \text{tr}\left(\mathbf{S}_{d_ed_e}(f_k)\right)(1 - A_e(f_k)),$$

and $A_e(f_k)$ is specified as the upper bound of the enhancement ratio of error signals at frequency f_k. Equation (2c) is a constraint on the amplitude of the ANC filter response for each channel, $\mathbf{W}_{x_{i,j}}(f_k)$, to ensure that the loudspeakers should operate in its linear response range, where $C_{i,j}(f_k)$ is the required upper bound on the amplitude of filter $\mathbf{w}_{F_{i,j}}$ at frequency f_k. Because of the use of IMC structure, Eqs. (2d) and (2e) are applied to ensure the stability and robustness of controller, $\mathbf{H}$, respectively, where

$$\mathbf{A}_s = -\mathbf{W}_x(f_k)\hat{\mathbf{G}}_{s_0}(f_k),$$

$\lambda(\,)$ and $\sigma(\,)$ denote the eigenvalues and singular values of a matrix, respectively, and $B(f_k)$ is the upper bound on the output multiplicative plant uncertainty at frequency f_k. More details on this convex formulation can be found in the work by Zhuang and Liu.[30]

B. Reformulation into cone programming using standard procedure

In the previous study,[30] it was shown that the computational efficiency in solving the ANC filter design optimization problem can be significantly improved if the above convex formulation, Eq. (2), is reformulated into an equivalent cone programming problem and solved by primal-dual interior-point algorithms that were designed for cone programming. To perform this conic reformulation, some rearrangements of the formulation are needed to transform the problem to a standard semidefinite-quadratic-linear programming (SQLP) problem[43] (sometimes also referred to as a mixed semidefinite and second-order cone optimization problem[44]). This standard SQLP form is especially neccessary if general cone programming solvers, such as SeDuMi,[37,44] SDPT3,[38,43] or MOSEK,[39] are used. In general, the standard SQLP problem is expressed as

$$\begin{aligned} &\text{minimize}(\vec{\mathbf{c}}^l)^T\vec{\mathbf{x}}^l + (\vec{\mathbf{c}}^q)^T\vec{\mathbf{x}}^q + (\vec{\mathbf{c}}^s)^T\vec{\mathbf{x}}^s,\\ &\text{subject to } \mathbf{A}^l\vec{\mathbf{x}}^l + \mathbf{A}^q\vec{\mathbf{x}}^q + \mathbf{A}^s\vec{\mathbf{x}}^s = \vec{\mathbf{b}},\\ &\vec{\mathbf{x}}^l \in \Re_+^{k_l}, \quad \vec{\mathbf{x}}^q \in K^q, \quad \vec{\mathbf{x}}^s \in K^s, \end{aligned} \tag{3}$$

where $\Re_+^{k_l}$ is the nonnegative orthant with dimension k_l, $K^q = K_1^q \times \cdots \times K_{k_q}^q$ is a Cartesian product of k_q second-order cones (sometimes also referred to as Lorentz cones) such that

$$K_i^q = \left\{(y, \vec{\mathbf{x}}) \in \Re \times \Re^{n_i - 1} \,|\, y \geq \|\vec{\mathbf{x}}\|_2\right\}, \tag{4}$$

and $K^s = K_1^s \times \cdots \times K_{k_s}^s$ is a Cartesian product of k_s vectorizations of positive semidefinite (PSD) cones,

$$K_i^s = \left\{\text{vec}(X) \in \Re^{n_i^2} \,|\, X \in \Re^{n_i \times n_i} \text{ is PSD}\right\}. \tag{5}$$

Due to diverse mathematical characteristics of the expressions in Eq. (2), different reformulation treatments need to be followed to accomplish the conic reformulation. The objective function, disturbance enhancement constraint, and filter response magnitude constraint are reformulated into second-order cones because they are quadratic functions in essence. Those quadratic functions are linearized by replacing the second-order terms with introduced additional variables. To ensure that the reformulated objective function and constraints are equivalent to the original convex formulation, the introduced variables are constrained by second-order cones related to the second-order terms in the original quadratic functions. The reformulated functions are further rearranged into the standard SQLP form as in Eqs. (A4), (A7), and (A10) (a detailed description of this reformulation process can be found in Appendix A).

The stability constraint and robustness constraint are reformulated into PSD cones because they involve constraining the maximum eigenvalues or singular values of some matrices. Those reformulations are also equivalent, and the reformulated functions are further rearranged into the standard SQLP form as in Eqs. (B7) and (B8) (a detailed description of this reformulation process can be found in Appendix B).

Finally, the constrained multichannel ANC filter design optimization problem, Eq. (2), can be reformulated in a standard SQLP form by combining Eqs. (A4), (A7), (A10), (B7), and (B8) such that

$$\begin{aligned} &\text{minimize}\begin{bmatrix} (\vec{\mathbf{c}}_w)^T & (\vec{\mathbf{c}}_x)^T \end{bmatrix}\begin{bmatrix} \vec{\mathbf{w}} \\ \vec{\mathbf{x}} \end{bmatrix},\\ &\text{subject to } \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}\begin{bmatrix} \vec{\mathbf{w}} \\ \vec{\mathbf{x}} \end{bmatrix} = \vec{\mathbf{b}},\\ &\vec{\mathbf{w}} \in \Re^{N_r N_s N_t}, \quad \vec{\mathbf{x}} \in K, \end{aligned} \tag{6}$$

where $\vec{\mathbf{x}}$ is a concatenation of $\vec{\mathbf{x}}_0$ in Eq. (A4), $\vec{\mathbf{x}}_{1,k}$ in Eq. (A7), $\vec{\mathbf{x}}_{2,l,k}$ in Eq. (A10), $\vec{\mathbf{x}}_{3,k}$ in Eq. (B7), and $\vec{\mathbf{x}}_{4,k}$ in Eq. (B8) for all k and l, $\vec{\mathbf{c}}_x$ is a sparse vector whose first element is the only nonzero element with a value of one, $\mathbf{A}$ is obtained by vertically concatenating $\mathbf{A}_0$, $\mathbf{A}_{1,k}$, $\mathbf{A}_{2,l,k}$, $\mathbf{A}_{3,k}$, and $\mathbf{A}_{4,k}$ for all k and l, $\mathbf{B}$ is a sparse block diagonal matrix obtained by diagonally concatenating $\mathbf{B}_0$, $\mathbf{B}_{1,k}$, $\mathbf{B}_{2,l,k}$, $\mathbf{B}_{3,k}$, and $\mathbf{B}_{4,k}$ for all k and l, $\vec{\mathbf{b}}$ is obtained by vertically concatenating $\vec{\mathbf{b}}_0$, $\vec{\mathbf{b}}_{1,k}$, $\vec{\mathbf{b}}_{2,l,k}$, $\vec{\mathbf{b}}_{3,k}$, and $\vec{\mathbf{b}}_{4,k}$ for all k and l, and K is a Cartesian product of all associated cones (a detailed expression for these constant matrices and vectors can be found in Appendixes A and B).

The cone programming problem specified in Eq. (6) can be directly solved by solvers with primal-dual interior-point algorithms designed specifically for cone programming, e.g., SeDuMi,[37,44] SDPT3,[38,43] or MOSEK.[39] However, in the practical ANC applications, where the number of frequencies of interest is much larger than the order of filters, the number of additionally introduced variables, $\vec{\mathbf{x}}$, is much larger than the dimension of $\vec{\mathbf{w}}$, which enlarges the problem size significantly. Also, the definition of cone programming, Eq. (3), shows that no free variables (i.e., the variables that are not in any conic constraints) are allowed, whereas $\vec{\mathbf{w}}$ in Eq. (6) is a set of free variables. Although various approaches can be used to convert free variables to variables constrained by conic constraints, numerical behavior may be degraded due to these additional redundant constraints.[37,38] A previous study by Zhuang and Liu also demonstrated the numerical stability issue associated with the free variables in ANC applications.[41] In a 4-input-2-output ANC system with filter order 128, the number of free variables $\vec{\mathbf{w}}$ in Eq. (6) is 1024, and such a large number of free variables can cause severe numerical instability issues. Thus, the direct use of Eq. (6) may not converge to a satisfactory filter coefficients solution or even fail to give a solution,[41] and these numerical issues will be demonstrated in the simulation results section (Sec. III).

As a physical interpretation, the free variables in ANC application come from the fact that most of the constraints on filter design are applied in the frequency domain instead of directly on the filter coefficients. By using the standard procedure, redundant variables and constraints have to be introduced, which, although equivalent in theory, causes the numerical stability problems in practice. In Sec. II C, an improved formulation is proposed to eliminate those redundant variables and constraints to prevent numerical instability.

C. Proposed numerically stable cone formulation

In this section, the problem structure is further exploited and simplified via dual formulation to achieve better numerical stability when common cone programming algorithms are used to solve this problem. The goal of the proposed formulation is to eliminate the free variables and reduce the problem formulation dimension because they both contribute to the abovementioned numerical stability issues.

For a convex conic formulation (usually referred to as the primal problem), there exists a dual conic problem. The variables of the primal problem are associated with the constraints in the dual problem and, similarly, the variables of the dual problem are associated with the constraints in the primal problem. The optimal objective function values for the primal and dual formulations are the same. Although, mathematically, solving a dual problem has the same computational complexity as solving a primal problem. In the following derivation, it is shown that it will be easier to exploit the sparsity patterns and simplify the formulation if a dual formulation technique is applied.

1. Derivation of proposed cone formulation via dual formulation technique

First, in Eq. (6), it is noted that $\vec{\mathbf{c}}_x$ has only one nonzero element because only t_0 appears in the objective function. One introduced new variable that is equal to t_0 can be appended to the end of $\vec{\mathbf{w}}$. After this groups, this vector is referred to as $\vec{\tilde{\mathbf{w}}}$. Equation (6) can then be reformulated equivalently as

$$\begin{aligned}&\text{minimize}(\vec{\tilde{\mathbf{c}}}_w)^T\vec{\tilde{\mathbf{w}}},\\ &\text{subject to}\;\;\begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\end{bmatrix}\begin{bmatrix}\vec{\tilde{\mathbf{w}}}\\ \vec{\mathbf{x}}\end{bmatrix}=\vec{\tilde{\mathbf{b}}},\\ &\vec{\tilde{\mathbf{w}}}\in\Re^{N_rN_sN_t+1},\quad \vec{\mathbf{x}}\in K,\end{aligned}\qquad(7)$$

where

$$\vec{\tilde{\mathbf{c}}}_w=\begin{bmatrix}\vec{\mathbf{c}}_w\\ 1\end{bmatrix},\;\vec{\tilde{\mathbf{b}}}=\begin{bmatrix}0\\ \vec{\mathbf{b}}\end{bmatrix},\;\tilde{\mathbf{A}}=\begin{bmatrix}\mathbf{0} & -1\\ \mathbf{A} & 0\end{bmatrix},\;\tilde{\mathbf{B}}=\begin{bmatrix}1 & \mathbf{0}\\ & \mathbf{B}\end{bmatrix}.$$

Reformulation to Eq. (7) has two benefits compared with Eq. (6): (1) variables that are being constrained by cones no longer appear in objective function directly, which is a key step to allow simplification described later in this section; and (2) the added one row, $[1\quad \mathbf{0}]$, to $\mathbf{B}$ makes $\tilde{\mathbf{B}}$ a square block diagonal matrix with each block being square. It can be further noticed that all blocks in $\tilde{\mathbf{B}}$ are invertible and well conditioned, which suggests that $\tilde{\mathbf{B}}$ is always invertible and well conditioned, whereas calculating $\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}^{-1}\vec{\tilde{\mathbf{b}}}$ does not require extra computational effort (see Appendix C for details). With the inverse of $\tilde{\mathbf{B}}$, Eq. (7) can then be formulated as

$$\begin{aligned}&\text{minimize}(\vec{\tilde{\mathbf{c}}}_w)^T\vec{\tilde{\mathbf{w}}},\\ &\text{subject to}\;\;\begin{bmatrix}\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\vec{\tilde{\mathbf{w}}}\\ \vec{\mathbf{x}}\end{bmatrix}=\tilde{\mathbf{B}}^{-1}\vec{\tilde{\mathbf{b}}},\\ &\vec{\tilde{\mathbf{w}}}\in\Re^{N_rN_sN_t+1},\quad \vec{\mathbf{x}}\in K.\end{aligned}\qquad(8)$$

It is noted that $\vec{\tilde{\mathbf{w}}}$ is the variable vector containing the ANC filter coefficients and one dummy variable. By noting that the cones in Eq. (8) are all self-dual cones, then the dual formulation of Eq. (8) is[37,38,44,45]

$$\begin{aligned}&\text{minimize}-(\tilde{\mathbf{B}}^{-1}\vec{\tilde{\mathbf{b}}})^T\vec{\mathbf{y}},\\ &\text{subject to}\;\;\begin{bmatrix}\left(\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}\right)^{\mathrm{T}}\\ \mathbf{I}\end{bmatrix}\vec{\mathbf{y}}+\begin{bmatrix}\vec{\mathbf{z}}_1\\ \vec{\mathbf{z}}_2\end{bmatrix}=\begin{bmatrix}\vec{\tilde{\mathbf{c}}}_w\\ \mathbf{0}\end{bmatrix},\\ &\vec{\mathbf{y}}\in\Re^m,\;\;\vec{\mathbf{z}}_1=\mathbf{0},\;\;\vec{\mathbf{z}}_2\in K,\end{aligned}\qquad(9)$$

where $\vec{\mathbf{y}}$, $\vec{\mathbf{z}}_1$, and $\vec{\mathbf{z}}_2$ are the dual variables associated with equality constraints, free variables, $\vec{\tilde{\mathbf{w}}}$, and conic constraints, respectively, in Eq. (8), and m is the row number of $\tilde{\mathbf{B}}$. It is noted from the lower portion of the equality constraints in Eq. (9) that $\vec{\mathbf{y}}=-\vec{\mathbf{z}}_2$ (this is the direct result of the first benefit mentioned above), and by further substitution of $\vec{\mathbf{z}}_1=\mathbf{0}$ back into the equality constraints, Eq. (9) can be simplified by eliminating $\vec{\mathbf{z}}_1$ and $\vec{\mathbf{z}}_2$:

$$\begin{aligned}&\text{minimize}(\tilde{\mathbf{B}}^{-1}\vec{\tilde{\mathbf{b}}})^T(-\vec{\mathbf{y}}),\\ &\text{subject to}\;\;\left(\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}\right)^T(-\vec{\mathbf{y}})=-\vec{\tilde{\mathbf{c}}}_w,\quad(-\vec{\mathbf{y}})\in K.\end{aligned}\qquad(10)$$

Equation (10) is the proposed simplified formulation. Because the dimension of the problem is greatly reduced and no free variables are involved in this formulation, numerical stability can be significantly improved when standard cone programming algorithms are used to solve this problem.

2. Obtaining filter coefficients from proposed cone formulation

It is noted that Eq. (10) does not have filter coefficients as its variables explicitly. Instead, the filter coefficients are the dual variables associated with equality constraints of Eq. (10) if Eq. (10) is treated as a standard primal SQLP form with $-\vec{\mathbf{y}}$ as primal variables. When primal-dual interior-point algorithms are used for this simplified problem [Eq. (10)], the dual variables associated with the optimum primal variables can be exported without extra computational effort because the dual solution is calculated along with the primal solution.[37,38,44,45] This means that although the problem formulated in Eq. (10) is not directly based on ANC filter coefficients, the optimal filter coefficients are indeed direct results of common primal-dual cone programming algorithms.

To show the relationship between filter coefficients, $\vec{\tilde{\mathbf{w}}}$, and the proposed formulation, Eq. (10), rewrite Eq. (8) as

$$\begin{aligned}&\text{maximize}(-\vec{\tilde{\mathbf{c}}}_w)^T\vec{\tilde{\mathbf{w}}},\\ &\text{subject to}\;\;\left(\left(\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}\right)^T\right)^T\vec{\tilde{\mathbf{w}}}+\vec{\mathbf{x}}=\tilde{\mathbf{B}}^{-1}\vec{\tilde{\mathbf{b}}},\\ &\vec{\tilde{\mathbf{w}}}\in\Re^{N_rN_sN_t+1},\quad \vec{\mathbf{x}}\in K.\end{aligned}\qquad(11)$$

Equation (11) can be treated directly as a standard SQLP dual form of another primal problem, where $\vec{\tilde{\mathbf{w}}}$ and $\vec{\mathbf{x}}$ are the

TABLE I. A comparison of problem dimensions using different formulation methods.

	Method 1 Lorentz cones	Method 1 Nonnegative variables	Method 2
Primal variables	$m+N_rN_sN_t+1$	$m+2N_rN_sN_t$	m
Equality constraints	m	m	$N_rN_sN_t+1$
Inequality constraints	$m+N_rN_sN_t+1$	$m+2N_rN_sN_t$	m

dual variables associated with equality and conic constraints in that primal problem, respectively. Then it is easy to obtain the associated primal form as[37,38,44,45]

$$\begin{aligned}&\text{minimize}(\tilde{\mathbf{B}}^{-1}\vec{\tilde{\mathbf{b}}})^T\vec{\mathbf{s}},\\&\text{subject to }\left(\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}\right)^T\vec{\mathbf{s}}=-\vec{\tilde{\mathbf{c}}}_w,\\&\vec{\mathbf{s}}\in K,\end{aligned}\tag{12}$$

where $\vec{\mathbf{s}}$ are the primal variables associated with the equality constraints in Eq. (11). By a change of variable, $\vec{\mathbf{s}}=-\vec{\mathbf{y}}$, it can be shown that Eqs. (10) and (12) are exactly the same. This derivation shows the relationship between ANC filter coefficients and the dual variables clearer when described in Eq. (11) compared with the former derivation. This means that when Eq. (12) [or Eq. (10)] is treated as a primal form and solved by common cone programming algorithms, the optimal dual variables associated with equality constraints in Eq. (12) [or Eq. (10)] will be the optimal ANC filter coefficients. Because of the primal-dual properties of common cone programming algorithms, obtaining the optimal dual variables does not cause any extra computations compared with only obtaining the optimal primal variables.[37,38,44,45]

3. Analysis of proposed cone formulation

To explain why Eqs. (10) or (12) (referred to as "method 2") is a more preferable way to calculate the optimal ANC filter coefficients compared with Eq. (6) (referred to as "method 1"), the dimensions of the two problem formulations are compared in Table I. It is noted that the number of variables and constraints are considered because when solving the primal problem using primal-dual interior-point algorithms, the dual problem is solved at the same time along with the primal problem. The dimension of the dual problem variables is determined by the number of constraints for the primal problem.[37,38,44,45] m is the summation of the dimensions of all of the cones, which is usually much larger than $N_rN_sN_t$. In the column of method 1 in Table I, there are two possible dimensions because of different treatments of free variables, $\vec{\mathbf{w}}$ (i.e., the free variables can be either placed inside a Lorentz cone or converted into the difference of two nonnegative variables).[41] By comparing the problem dimension of two different methods, it is obvious that method 2 has a much smaller dimension than method 1 (e.g., for one of the cases in the simulation result Sec. III, m is around 2×10^4 while $N_rN_sN_t$ is around 10^3). Therefore, the optimization problem obtained by method 1 has redundant constraint information and large sparse patterns which may not be well exploited by the standard algorithms in solvers. The existence of free variables, $\vec{\mathbf{w}}$, in method 1 may also compromise the numerical behavior,[37,38,41,44,45] and those free variables are eliminated by using the method 2. In Sec. III, experimentally measured data will be used to confirm that using method 2 has a better numerical behavior and higher computational efficiency compared with using method 1.

The procedure of implementing method 1, i.e., the cone formulation using standard procedure, and method 2, i.e., the proposed formulation, is shown in Fig. 2. It is noted that because of the use of primal-dual interior-point algorithm, the optimal primal and dual solutions are obtained simultaneously. Thus, using the proposed formulation does not complicate the method implementation procedure.

III. SIMULATION RESULTS

Simulations based on experimentally measured data were performed to investigate the numerical behavior of two different methods, i.e., the directly formulated SQLP formulation using standard procedure, Eq. (6) (method 1), and the proposed simplified formulation in dual form, Eq. (10) or (12) (method 2). In the measurement phase, the ANC system

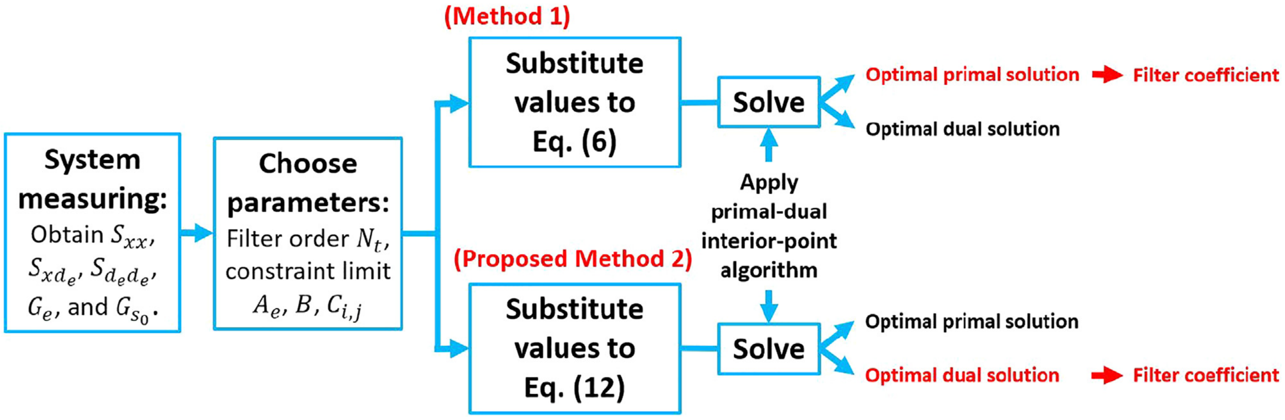

(Method 1)
Substitute values to Eq. (6)
Solve
Optimal primal solution
Filter coefficient
Optimal dual solution
System measuring: Obtain S_{xx}, S_{xd_e}, $S_{d_ed_e}$, G_e, and G_{s_0}.
Choose parameters: Filter order N_t, constraint limit A_e, B, $C_{i,j}$
Apply primal-dual interior-point algorithm
(Proposed Method 2)
Substitute values to Eq. (12)
Solve
Optimal primal solution
Optimal dual solution
Filter coefficient

FIG. 2. (Color online) A flow chart showing the procedures of implementing method 1 (conic formulation using standard procedure) and method 2 (proposed formulation).

was installed on the wind channel of a central air handling system, which is similar to the previous work of Zhuang and Liu[30] and is shown in Fig. 3. When acquiring measurement data, the sampling rate of the data acquisition system is 48 kHz. For calculating the correlation matrices and frequency responses, 2×10^6 sampling points for each channel were acquired, where a Hamming window of 48 000 points is used for averaging with 50% overlapping.

Due to its large duct size and wide targeted control frequency range, a multichannel ANC system is required. The system used in the current work includes four reference microphones ($N_r = 4$), two loudspeakers ($N_s = 2$), and four error microphones ($N_e = 4$). Thus, 2×4 (a total of eight) FIR filters are required to be designed. The sampling frequency f_s is 7000 Hz, and the desired noise attenuation band is from 100 to 3400 Hz. In the desired noise attenuation band, 1101 equally spaced frequencies (i.e., 3 Hz frequency resolution) were used for the objective function, and 413 equally spaced frequencies (i.e., 8 Hz frequency resolution) were used for the disturbance enhancement constraint. The allowed disturbance enhancement is 3 dB, i.e., $A_e = 2$ for $\tilde{c}_J(f_k)$ in Eq. (2b). A different number of frequencies were used in stability and robustness constraints in diverse comparative cases described in Secs. III A and III B. The reason that different numbers of frequencies are used for stability and robustness constraints is that those two types of constraints were demonstrated to be closely related to the numerical stability behaviors.[41] The ϵ_s in Eq. (2d) is set to 0.1 to ensure strict stability. The $B(f_k)$ in Eq. (2e) is chosen to be the upper bound on the output multiplicative plant uncertainty, which is 10% in this experimental setup. Outside of the desired attenuation band, 68 frequencies (i.e., 3 Hz frequency resolution) were used in filter response magnitude constraints. The $C_{i,j}(f_k)$ in Eq. (2c) is set to 0.01 to ensure that the speakers used in the experimental setup operate in their linear response range. Two solvers, SeDuMi[37,44] and SDPT3,[38,43] were used to implement different variants of primal-dual interior-point algorithms to obtain more general conclusions on the improvements brought by the proposed reformulation. Both solvers will stop iterating when either an optimal solution is found or the generated step length becomes sufficiently small (a numerical issue can be considered happening if the solver stops iterating while an optimal solution is not found yet). In this section, the processor used in the current work is Intel(R) Core(TM) i7–7700 CPU at 3.60 GHz (Santa Clara, CA), and the programming platform is MATLAB (Natick, MA) installed on Windows 10 64-bit operating system (Microsoft, Redmond, WA). The double-precision floating-point numbers are used.

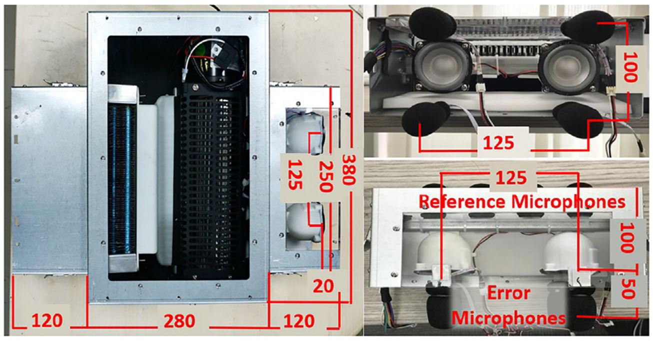


FIG. 3. (Color online) The picture and dimensions of the experimental setup (the units are mm; Ref. 30).

A. Comparison of numerical stability

For typical primal-dual interior-point algorithms designed for cone programming, numerical issues may occur when the solution is close to the optimal solution or boundary of constraints.[38,44,46] In practical algorithms, one of the stopping criteria is the duality gap, which measures the difference between primal and dual objective function values. The duality gap converges to zero when the current solution is approaching the optimal solution.[37,45] To compare the numerical stability of different formulating methods, the duality gap just before a numerical issue occurs is chosen to be the criterion, i.e., the smaller the duality gap is, then the better the numerical stability is.

In ANC filter design, the numerical issues usually occur when PSD cones, i.e., due to stability and robustness constraints, exist.[41] Three different frequency resolutions for stability and robustness constraints were used as different cases: 56, 83, and 111 frequencies over the desire noise control band, i.e., 60, 40, and 30 Hz frequency resolutions, respectively. After trials of different frequency resolutions, the 56 frequencies (60 Hz frequency resolution) are the minimum number of frequencies to ensure robust stability across the whole frequency band. In each case, different FIR filter lengths, N_t, were also used to compare the numerical behavior for different problem dimensions. The duality gaps before numerical issues occur for different cases, filter lengths, methods, and solvers are shown in Fig. 4. These plots show that for both solvers, method 2 always has a smaller final duality gap, i.e., is more numerically stable, than method 1, especially if using solver SeDuMi for high order FIR filters. When method 1 is used for higher order FIR filters, it is possible that no reasonable solution can be obtained by SeDuMi since some duality gap in these plots can only reach around 1×10^{-3}, which is several orders larger than the duality gap achieved by the proposed formulation (method 2), suggesting that it is not converged to the optimum solution when the solver stops iterating because of a sufficiently small step length. The results also demonstrated that the solver SDPT3 is usually more numerically stable compared with SeDuMi when the same formulating method is used.

To confirm that such a small duality gap is indeed needed to achieve satisfactory ANC performance, the ANC performance (in nominal condition) comparison of 96-point and 192-point FIR filters designed by different methods and different solvers is shown in Fig. 5, where 111 frequencies for stability and robustness constraints are used. "ANC OFF" denotes the sound pressure power spectral density

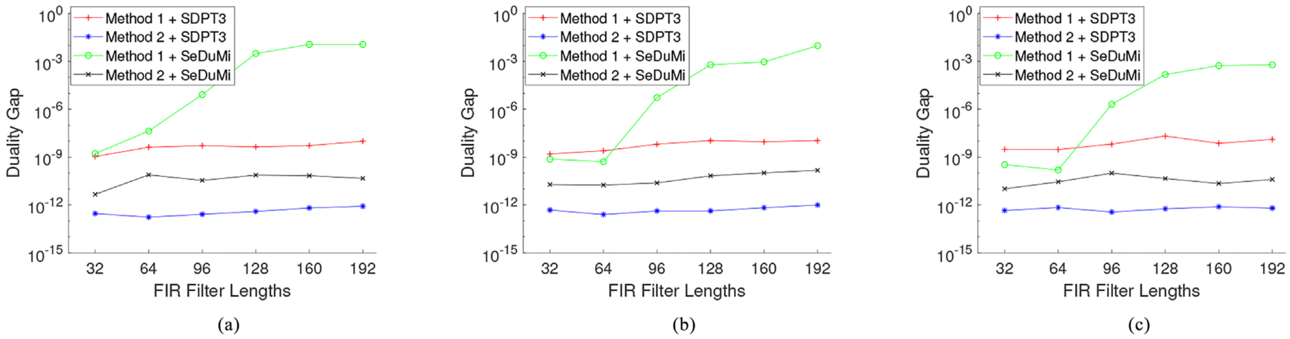


FIG. 4. (Color online) The duality gap before numerical issues occur when using (a) 56 frequencies, (b) 83 frequencies, and (c) 111 frequencies for stability and robustness constraints.

function (PSDF) of original disturbance signals averaged among all of the error microphones. In Fig. 5, "normalized SPL" (normalized sound pressure level) is the ratio of the sensor averaged PSDF when the ANC system is in operation to the PSDF averaged among error sensors and the whole frequency band when ANC OFF. In Fig. 5, the performances of ANC filters solved by using method 1 and SeDuMi, which has duality gaps of 2.04×10^{-6} for Fig. 5(a) and 6.02×10^{-4} for Fig. 5(b), is obviously worse than other combinations which have duality gaps that are all less than 10^{-8}. This confirms that using method 1 may not give a satisfactory result if the solver SeDuMi is used and the problem dimension is large. Although in this specific ANC application, the solver SDPT3 can use method 1 to give an accurate enough solution, it is still less numerically stable (higher duality gap) than the proposed method 2, i.e., when higher precision is required in other ANC applications, the proposed method 2 may be preferable. In Sec. III B, it will also be shown that proposed method 2 will always have a better computational efficiency compared with method 1 in both solvers.

A comparison of the noise control performance with and without using constraints is shown in Fig. 6(a). The "Wiener filter" indicates the optimal filter without considering any constraints. From Fig. 6(a), results without constraints, the disturbance enhancement is high in lower and higher frequency ranges for the Wiener filter case. Furthermore, the stability constraint function values are shown in Fig. 6(b). When the values are higher than the blue dot line, the designed filter is not stable. For the Wiener filter case, it exceeds the bound around 1500 and 2500 Hz, which indicates that it is not stable when implemented in real time. This suggests that the stability constraint is indeed necessary in this particular application and the proposed method can ensure the satisfaction of the required stability constraints.

B. Comparison of computational efficiency

Besides the numerical stability behavior, the computational efficiency was also investigated for the two formulating methods. The computational times for different cases, filter lengths, methods, and solvers are shown in Fig. 7. It is noted that for the combination of method 1 and SeDuMi, the results only include cases with filter lengths of 32 and 64 because the algorithm cannot give an accurate enough solution for a higher order FIR filter as confirmed in Sec. III A. In Fig. 7, results demonstrated that for both solvers, method 2 is much more computationally efficient than method 1, especially when solver SDPT3 is used (computational time

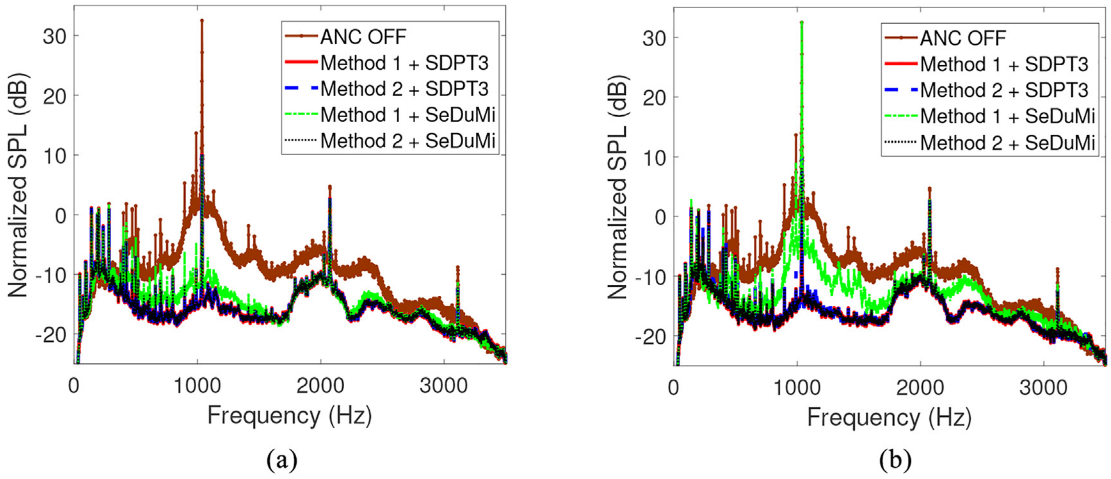
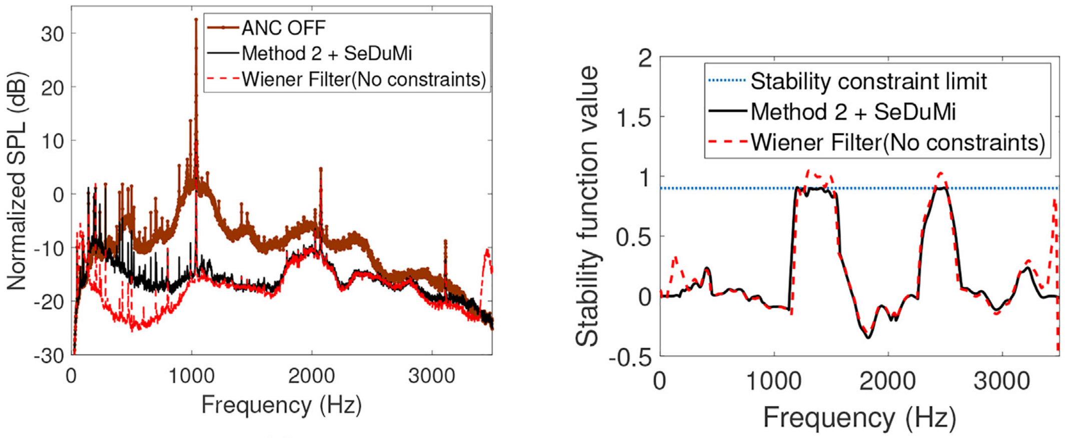


FIG. 5. (Color online) The comparison of noise control performance using (a) 96-point FIR filters, and (b) 192-point FIR filters, when 111 frequencies are used for the stability and robustness constraints.

https://doi.org/10.1121/10.0014627



(a)

(b)

FIG. 6. (Color online) The comparison of (a) the noise control performance, and (b) the stability constraint function values using 96-point FIR filters.

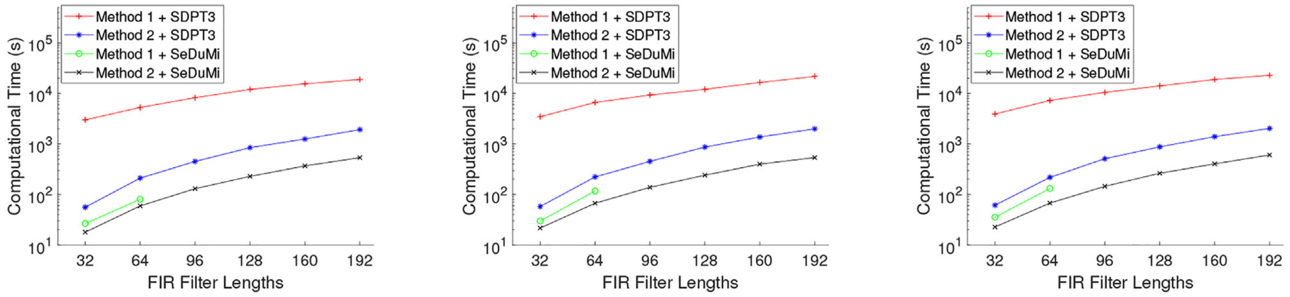
was reduced by two orders). The results also showed that using solver SeDuMi is significantly more efficient than using solver SDPT3, but the use of SeDuMi is only possible when the proposed formulation is used (i.e., method 2).

To summarize the analysis results, the proposed method 2 is more numerically stable and computationally efficient compared with method 1 in both solvers. Also, the combination of method 2 and SeDuMi is more preferable to solve the multichannel ANC filter design problem because it usually has better numerical stability and computational efficiency. If in some cases where the optimal solution needs to be extremely accurate, i.e., a very small duality gap is required, then the combination of method 2 and SDPT3 can be chosen to solve the problem to have a higher accuracy.

IV. SUMMARY AND CONCLUSIONS

Although the computational efficiency in designing optimal filters for a constrained multichannel ANC system can be significantly improved when the traditional H_2/H_∞ formulation is converted into a cone programming formulation, numerical instability issues are likely to occur when using the standard conic reformulation procedure. In the current work, it is found that this numerical issue is caused by the presence of free variables that occurred in the standard cone programming reformulation process. A conic formulation with improved numerical stability is proposed. This alternative reformulation is achieved by converting the formulation that resulted from the standard conic reformulation to its dual formulation, where all of the free variables can be eliminated through proper matrix rearrangements. The removal of free variables not only effectively alleviates the numerical stability issue but also further improves the computational efficiency in solving the optimization problem.

The experiment results confirmed that compared with the conic formulation using standard procedure, the proposed simplified conic formulation is more numerically stable and computationally efficient. It shows that the SeDuMi solver is more computationally efficient in most cases but less numerically stable than the SDPT3 solver. The achievable numerical stability is usually satisfactory when using the SeDuMi solver if the proposed formulation is used. The improvement of numerical stability and computational efficiency brought by the proposed formulation is an important step to potentially realize constrained adaptive control for ANC systems with multiple types of constraints by a continuous redesign procedure.



(a)

(b)

(c)

FIG. 7. (Color online) The computational time when using (a) 56 frequencies, (b) 83 frequencies, and (c) 111 frequencies for the stability and robustness constraints.

ACKNOWLEDGMENTS

The authors thank Beijing Ancsonic Technology Co. Ltd for the help on the experimental system setup and data collection and providing financial support related to the presented work.

APPENDIX A: SECOND-ORDER CONE REFORMULATION

In this appendix, the expressions related to the second-order cone reformulation, i.e., $\mathbf{A}_0$, $\mathbf{A}_{1,k}$, $\mathbf{A}_{2,l,k}$, $\mathbf{B}_0$, $\mathbf{B}_{1,k}$, $\mathbf{B}_{2,l,k}$, $\vec{\mathbf{b}}_0$, $\vec{\mathbf{b}}_{1,k}$, and $\vec{\mathbf{b}}_{2,l,k}$ in Eq. (6), are derived.

For convenience, $\vec{v} \in K_n^q$ and $\vec{v} \in K_n^s$ are used to represent all of the conic relations without assigning a different notation for each conic constraint.

For the objective function, a new variable t_0 and new constraint can be introduced. Minimizing Eq. (2a) is equivalent to

$$\text{minimize}_{\mathbf{w},t_0}\; t_0 + \sum_{k=1}^{N_f} \vec{\mathbf{b}}_J^T(f_k)\vec{\mathbf{w}}, \text{ subject to } \|\mathbf{M}_0\vec{\mathbf{w}}\|_2 \le \sqrt{t_0}, \tag{A1}$$

where the matrix $\mathbf{M}_0$ is a matrix such that $\mathbf{M}_0^T\mathbf{M}_0 = \sum_{k=1}^{N_f} \mathbf{A}_J(f_k)$, which, depending on the rank of matrix, can be obtained by either a square root of matrix,[47] Cholesky factorization, or full rank factorization using singular value decomposition. Note that the two sets,

$$K = \left\{(\vec{\mathbf{x}},y_1,y_2) \in \Re^n \times \Re \times \Re \,|\, y_1,y_2 \ge 0, \sqrt{y_1y_2} \ge \|\vec{\mathbf{x}}\|_2\right\},$$
$$K = \left\{(\vec{\mathbf{x}},y_1,y_2) \in \Re^n \times \Re \times \Re \,\middle|\, \left\|\begin{bmatrix} 2\vec{\mathbf{x}} \\ y_1 - y_2 \end{bmatrix}\right\|_2 \le y_1 + y_2\right\}, \tag{A2}$$

are equivalent.[48] Thus, Eq. (A1) can be reformulated equivalently as

$$\text{minimize}\; t_0 + \sum_{k=1}^{N_f} \vec{\mathbf{b}}_J^T(f_k)\vec{\mathbf{w}}, \text{ subject to } \left\|\begin{bmatrix} 2\mathbf{M}_0\vec{\mathbf{w}} \\ t_0 - 2 \end{bmatrix}\right\|_2 \le t_0. \tag{A3}$$

By introducing a new set of variables, $\vec{\mathbf{x}}_0 \in \Re^{N_rN_sN_t+2}$, and eliminating t_0, Eq. (A3) can be rearranged as

$$\begin{aligned} &\text{minimize} (\vec{\mathbf{c}}_w)^T\vec{\mathbf{w}} + (\vec{\mathbf{c}}_0)^T\vec{\mathbf{x}}_0, \\ &\text{subject to } \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 \end{bmatrix} \begin{bmatrix} \vec{\mathbf{w}} \\ \vec{\mathbf{x}}_0 \end{bmatrix} = \vec{\mathbf{b}}_0, \\ &\vec{\mathbf{x}}_0 \in K_n^q, \end{aligned} \tag{A4}$$

where $\vec{\mathbf{w}}$ and $\vec{\mathbf{x}}_0$ are variables and the constants, $\vec{\mathbf{c}}_w \in \Re^{N_rN_sN_t}$, $\vec{\mathbf{c}}_0 \in \Re^{N_rN_sN_t+2}$, $\vec{\mathbf{b}}_0 \in \Re^{N_rN_sN_t+1}$, $\mathbf{A}_0 \in \Re^{(N_rN_sN_t+1)\times(N_rN_sN_t)}$, and $\mathbf{B}_0 \in \Re^{(N_rN_sN_t+1)\times(N_rN_sN_t+2)}$ are

$$\vec{\mathbf{c}}_w = \sum_{k=1}^{N_f} \vec{\mathbf{b}}_J(f_k), \quad \vec{\mathbf{c}}_0 = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix},$$
$$\vec{\mathbf{b}}_0 = \begin{bmatrix} \mathbf{0} \\ 2 \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} 2\mathbf{M}_0 \\ \mathbf{0} \end{bmatrix},$$
$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{I}_{N_sN_rN_t} \\ 1 & -1 & \mathbf{0} \end{bmatrix},$$

and $\mathbf{0}$ represents the zero matrix with appropriate dimension.

For the disturbance enhancement constraint, Eq. (2b), it is also in a quadratic form, which can be reformulated equivalently in a similar way as

$$t_{1,k} + \vec{\mathbf{b}}_J^T(f_k)\,\vec{\mathbf{w}} + \tilde{c}_J(f_k) = 0, \quad \|\mathbf{M}_{1,k}\,\vec{\mathbf{w}}\|_2 \le \sqrt{t_{1,k}}, \tag{A5}$$

where

$$\mathbf{M}_{1,k} = \left(\mathbf{G}_e^H(f_k)\mathbf{G}_e(f_k)\right)^{1/2} \otimes \left(\mathbf{S}_{xx}^T(f_k)\right)^{1/2} \otimes \mathbf{F}_z^T(f_k),$$

$\left(\mathbf{G}_e^H(f_k)\mathbf{G}_e(f_k)\right)^{1/2}$ and $\left(\mathbf{S}_{xx}^T(f_k)\right)^{1/2}$ are the complex matrix square roots of $\mathbf{G}_e^H(f_k)\mathbf{G}_e(f_k)$ and $\mathbf{S}_{xx}^T(f_k)$, respectively, which can be obtained by methods discussed in the objective function reformulation above. Equation (A5) can be rearranged from complex form to an equivalent real form, and then reformulated in a similar way as in the objective function reformulation into

$$t_{1,k} - 1 + \vec{\mathbf{b}}_J^T(f_k)\,\vec{\mathbf{w}} + \tilde{c}_J(f_k) = 0,$$
$$\left\|\begin{bmatrix} 2\Re(\mathbf{M}_{1,k})\vec{\mathbf{w}} \\ 2\Im(\mathbf{M}_{1,k})\vec{\mathbf{w}} \\ t_{1,k} - 2 \end{bmatrix}\right\|_2 \le t_{1,k}, \tag{A6}$$

where $\Im$ denotes the imaginary part of a complex value. By introducing a new set of variables, $\vec{\mathbf{x}}_{1,k} \in \Re^{2N_rN_s+2}$, and eliminating $t_{1,k}$, Eq. (A6) can be rearranged as

$$\begin{bmatrix} \mathbf{A}_{1,k} & \mathbf{B}_{1,k} \end{bmatrix} \begin{bmatrix} \vec{\mathbf{w}} \\ \vec{\mathbf{x}}_{1,k} \end{bmatrix} = \vec{\mathbf{b}}_{1,k}, \quad \vec{\mathbf{x}}_{1,k} \in K_n^q, \tag{A7}$$

for all frequencies f_k, where $\vec{\mathbf{w}}$ and $\vec{\mathbf{x}}_{1,k}$ are variables, and the constants, $\mathbf{A}_{1,k} \in \Re^{(2N_rN_s+2)\times(N_rN_sN_t)}$, $\mathbf{B}_{1,k} \in \Re^{(2N_rN_s+2)\times(2N_rN_s+2)}$, and $\vec{\mathbf{b}}_{1,k} \in \Re^{2N_rN_s+2}$, are

$$\mathbf{A}_{1,k} = \begin{bmatrix} 2\Re(\mathbf{M}_{1,k}) \\ 2\Im(\mathbf{M}_{1,k}) \\ \vec{\mathbf{b}}_J^T(f_k) \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_{1,k} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{I}_{2N_sN_r} \\ 1 & 0 & \mathbf{0} \\ 1 & -1 & \mathbf{0} \end{bmatrix},$$
$$\vec{\mathbf{b}}_{1,k} = \begin{bmatrix} \mathbf{0} \\ 1 - \tilde{c}_J(f_k) \\ 2 \end{bmatrix}.$$

For the filter response magnitude constraint, Eq. (2c), it is already in a similar form as a second-order cone. Note that

$$\begin{aligned}\|\mathbf{F}_z^T(f_k)\vec{\mathbf{w}}_{F_{i,j}}\|_2 &= \|\vec{\mathbf{e}}_l^T\left(\mathbf{I}_{N_r}\otimes\mathbf{I}_{N_s}\otimes\mathbf{F}_z^T(f_k)\right)(\mathbf{P}_r\otimes\mathbf{I}_{N_t})\vec{\mathbf{w}}\|_2\\ &= \|\vec{\mathbf{e}}_l^T\left(\mathbf{P}_r\otimes\mathbf{F}_z^T(f_k)\right)\vec{\mathbf{w}}\|_2,\end{aligned}\qquad \text{(A8)}$$

where $\vec{\mathbf{e}}_l\in\Re^{N_rN_s}$ is a unit vector for selecting information from an individual channel, it has values one in the lth element and zero in other elements; and sparse matrix $\mathbf{P}_r$ is used to rearrange the sequence of elements in vector. So the ith row and jth column of $\mathbf{P}_r\in\Re^{(N_rN_s)\times(N_rN_s)}$ is one if $(i,j)\in\mathbb{S}_{P_r}$, where $\mathbb{S}_{P_r}=\{((r-1)N_s+s,(s-1)N_r+r)\,|\,1\le r\le N_r, 1\le s\le N_s, r,s\in\mathbb{Z}\}$. Thus, Eq. (2c) can be converted to an equivalent real second-order cone formulation,

$$\left\|\begin{bmatrix}\vec{\mathbf{e}}_l^T\Re(\mathbf{P}_r\otimes\mathbf{F}_z(f_k))\\ \vec{\mathbf{e}}_l^T\Im(\mathbf{P}_r\otimes\mathbf{F}_z(f_k))\end{bmatrix}\vec{\mathbf{w}}\right\|_2\le C_{i,j}(f_k),\qquad \text{(A9)}$$

for all l and f_k, where $l=(j-1)N_s+i$. By introducing a new set of variables, $\vec{\mathbf{x}}_{2,l,k}\in\Re^3$, Eq. (A9) can be rearranged as

$$\begin{bmatrix}\mathbf{A}_{2,l,k} & \mathbf{B}_{2,l,k}\end{bmatrix}\begin{bmatrix}\vec{\mathbf{w}}\\ \vec{\mathbf{x}}_{2,l,k}\end{bmatrix}=\vec{\mathbf{b}}_{2,l,k},\quad \vec{\mathbf{x}}_{2,l,k}\in K_n^q.\qquad \text{(A10)}$$

for all frequencies f_k and all l, where $\vec{\mathbf{w}}$ and $\vec{\mathbf{x}}_{2,l,k}$ are variables, and the constants, $\mathbf{A}_{2,l,k}\in\Re^{3\times(N_rN_sN_t)}$, $\mathbf{B}_{2,l,k}\in\Re^{3\times3}$, and $\vec{\mathbf{b}}_{2,l,k}\in\Re^3$ are

$$\mathbf{A}_{2,l,k}=\begin{bmatrix}\vec{\mathbf{e}}_l^T\Re(\mathbf{P}_r\otimes\mathbf{F}_z(f_k))\\ \vec{\mathbf{e}}_l^T\Im(\mathbf{P}_r\otimes\mathbf{F}_z(f_k))\\ \mathbf{0}\end{bmatrix},$$

$$\mathbf{B}_{2,l,k}=\begin{bmatrix}0&-1&0\\0&0&-1\\1&0&0\end{bmatrix},\quad \vec{\mathbf{b}}_{2,l,k}=\begin{bmatrix}0\\0\\C_{i,j}(f_k)\end{bmatrix}.$$

APPENDIX B: PSD CONE REFORMULATION

In this appendix, the expressions related to the PSD cone reformulation, i.e., the $\mathbf{A}_{3,k}$, $\mathbf{A}_{4,k}$, $\mathbf{B}_{3,k}$, $\mathbf{B}_{4,k}$, $\vec{\mathbf{b}}_{3,k}$, and $\vec{\mathbf{b}}_{4,k}$ in Eq. (6), are derived.

Since each element of $\mathbf{A}_s$ and $\mathbf{A}_s^H$ is a linear function of variables $\vec{\mathbf{w}}_F$, the stability constraint, Eq. (2d), can be equivalently reformulated as the PSD cone with respect to the introduced variables as

$$-\mathbf{A}_s-\mathbf{A}_s^H+2(1-\epsilon_s)\mathbf{I}_{N_s}\succcurlyeq 0,\qquad \text{(B1)}$$

where $\succcurlyeq 0$ means that the matrix is PSD, and $\mathbf{I}_{N_s}$ is an $N_s\times N_s$ identity matrix.

For the robustness constraint, Eq. (2e), the Schur complement lemma is needed (because the elements of $\mathbf{A}_s^H\mathbf{A}_s$ are not linear transforms of $\vec{\mathbf{w}}_F$) to equivalently convert it to PSD cones,[49]

$$\begin{bmatrix}\dfrac{1}{B(f_k)}\mathbf{I}_{N_s} & -\mathbf{A}_s\\ -\mathbf{A}_s^H & \dfrac{1}{B(f_k)}\mathbf{I}_{N_s}\end{bmatrix}\succcurlyeq 0.\qquad \text{(B2)}$$

Note that Eqs. (B1) and (B2) can be converted into equivalent real forms by using the property that a Hermitian matrix $A\succcurlyeq 0$ is equivalently to

$$\begin{bmatrix}\Re(A) & -\Im(A)\\ \Im(A) & \Re(A)\end{bmatrix}\succcurlyeq 0.\qquad \text{(B3)}$$

To express matrices in vectors forms, first,

$$\begin{aligned}\mathrm{vec}(-\mathbf{A}_s)&=\mathrm{vec}\left(\mathbf{W}_x(f_k)\hat{\mathbf{G}}_{s_0}(f_k)\right)\\ &=\left(\left(\hat{\mathbf{G}}_{s_0}^T(f_k)\otimes\mathbf{I}_{N_s}\right)\mathbf{P}_r\right)\otimes\mathbf{F}_z(f_k)\vec{\mathbf{w}}\\ &\triangleq\left(\mathbf{Q}_k^r+i\mathbf{Q}_k^i\right)\vec{\mathbf{w}},\end{aligned}\qquad \text{(B4)}$$

where the last equality is a notation of real and imaginary parts of the constant matrices for the use in latter formulations, and $\mathbf{P}_r$ is defined in Eq. (A8). Similarly,

$$\begin{aligned}\mathrm{vec}(-\mathbf{A}_s^H)&=\left(\mathbf{I}_{N_s}\otimes\hat{\mathbf{G}}_{s_0}^H(f_k)\right)\otimes\mathbf{F}_z^*(f_k)\,\vec{\mathbf{w}}\\ &\triangleq\left(\mathbf{R}_k^r+i\mathbf{R}_k^i\right)\vec{\mathbf{w}}.\end{aligned}\qquad \text{(B5)}$$

Thus, let $\mathbf{A}_{3,k}\in\Re^{(4N_s^2)\times(N_rN_sN_t)}$ be

$$\mathbf{A}_{3,k}=\begin{bmatrix}\bar{\mathbf{A}}_{3,k}\\ \mathbf{0}_{2N_s^2}\end{bmatrix},\quad \bar{\mathbf{A}}_{3,k}=\begin{bmatrix}\mathbf{A}_{3,k,1}^r\\ \mathbf{A}_{3,k,1}^i\\ \vdots\\ \mathbf{A}_{3,k,N_s}^r\\ \mathbf{A}_{3,k,N_s}^i\end{bmatrix},$$

where

$$\begin{aligned}\mathbf{A}_{3,k,s}^r&=[\mathbf{Q}_k^r((s-1)N_s+1:sN_s,:)\\ &\quad+\mathbf{R}_k^r((s-1)N_s+1:sN_s,:)],\\ \mathbf{A}_{3,k,s}^i&=[\mathbf{Q}_k^i((s-1)N_s+1:sN_s,:)\\ &\quad+\mathbf{R}_k^i((s-1)N_s+1:sN_s,:)],\end{aligned}$$

and $\mathbf{Q}_k^r(1:N_s,:)$ means the first to N_sth rows of matrix $\mathbf{Q}_k^r$ with all columns (similar for other notations, and the dimension of zero matrices is labeled to prevent confusion); constant matrix $\mathbf{B}_{3,k}\in\Re^{(4N_s^2)\times(4N_s^2)}$ is

$$\mathbf{B}_{3,k}=\begin{bmatrix}-\mathbf{I}_{2N_s^2} & \mathbf{0}\\ \mathbf{I}_{N_s}\otimes\mathbf{P}_s\otimes\mathbf{I}_{N_s} & -\mathbf{I}_{2N_s^2}\end{bmatrix},\quad \mathbf{P}_s=\begin{bmatrix}0&-1\\1&0\end{bmatrix},\qquad \text{(B6)}$$

where $\mathbf{P}_s$ comes from symmetric matrix structure. The ith element of the sparse vector $\vec{\mathbf{b}}_{3,k}\in\Re^{4N_s^2}$ is $-2(1-\epsilon_s)$ if

$i \in \mathbb{S}_{b_3}$, where $\mathbb{S}_{b_3} = \left\{2N_s(s-1)+s \,|\, 1 \le s \le N_s, s \in \mathbb{Z}\right\}$. The constant matrix, $\mathbf{A}_{4,k} \in \Re^{(16N_s^2)\times(N_rN_sN_t)}$, is

$$\mathbf{A}_{4,k} = \begin{bmatrix} \bar{\mathbf{A}}_{4,k} \\ \mathbf{0}_{8N_s^2,N_sN_rN_t} \end{bmatrix}, \quad \bar{\mathbf{A}}_{4,k} = \begin{bmatrix} \mathbf{A}^r_{4,k,\mathbf{R},1} \\ \mathbf{A}^i_{4,k,\mathbf{R},1} \\ \vdots \\ \mathbf{A}^r_{4,k,\mathbf{R},N_s} \\ \mathbf{A}^i_{4,k,\mathbf{R},N_s} \\ \mathbf{A}^r_{4,k,\mathbf{Q},1} \\ \mathbf{A}^i_{4,k,\mathbf{Q},1} \\ \vdots \\ \mathbf{A}^r_{4,k,\mathbf{Q},N_s} \\ \mathbf{A}^i_{4,k,\mathbf{Q},N_s} \end{bmatrix},$$

where

$$\mathbf{A}^r_{4,k,\mathbf{R},s} = \begin{bmatrix} \mathbf{0}_{N_s,N_sN_rN_t} \\ \mathbf{R}^r_k((s-1)N_s+1 : sN_s, :) \end{bmatrix},$$

$$\mathbf{A}^r_{4,k,\mathbf{Q},s} = \begin{bmatrix} \mathbf{Q}^r_k((s-1)N_s+1 : sN_s, :) \\ \mathbf{0}_{N_s,N_sN_rN_t} \end{bmatrix},$$

$$\mathbf{A}^i_{4,k,\mathbf{R},s} = \begin{bmatrix} \mathbf{0}_{N_s,N_sN_rN_t} \\ \mathbf{R}^i_k((s-1)N_s+1 : sN_s, :) \end{bmatrix},$$

$$\mathbf{A}^i_{4,k,\mathbf{Q},s} = \begin{bmatrix} \mathbf{Q}^i_k((s-1)N_s+1 : sN_s, :) \\ \mathbf{0}_{N_s,N_sN_rN_t} \end{bmatrix},$$

constant matrix, $\mathbf{B}_{4,k} \in \Re^{(16N_s^2)\times(16N_s^2)}$, is

$$\mathbf{B}_{4,k} = \begin{bmatrix} -\mathbf{I}_{8N_s^2} & \mathbf{0} \\ \mathbf{I}_{2N_s} \otimes \mathbf{P}_s \otimes \mathbf{I}_{2N_s} & -\mathbf{I}_{8N_s^2} \end{bmatrix},$$

and the ith element of the sparse vector, $\vec{\mathbf{b}}_{4,k} \in \Re^{16N_s^2}$, is $-1/B(f_k)$ if i or $(i-4N_s^2-N_s) \in \mathbb{S}_{b_4}$, where $\mathbb{S}_{b_4} = \left\{4N_s(s-1)+s \,|\, 1 \le s \le N_s, s \in \mathbb{Z}\right\}$.

Finally, with the new set of variables, $\vec{\mathbf{x}}_{3,k} \in \Re^{4N_s^2}$, Eq. (B1) can be equivalently reformulated as

$$\begin{bmatrix} \mathbf{A}_{3,k} & \mathbf{B}_{3,k} \end{bmatrix} \begin{bmatrix} \vec{\mathbf{w}} \\ \vec{\mathbf{x}}_{3,k} \end{bmatrix} = \vec{\mathbf{b}}_{3,k}, \quad \vec{\mathbf{x}}_{3,k} \in K^s_n, \tag{B7}$$

for all frequencies f_k. Similar relations can be obtained for Eq. (B2) as

$$\begin{bmatrix} \mathbf{A}_{4,k} & \mathbf{B}_{4,k} \end{bmatrix} \begin{bmatrix} \vec{\mathbf{w}} \\ \vec{\mathbf{x}}_{4,k} \end{bmatrix} = \vec{\mathbf{b}}_{4,k}, \quad \vec{\mathbf{x}}_{4,k} \in K^s_n. \tag{B8}$$

APPENDIX C: COMPUTATION OF $\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}$ AND $\tilde{\mathbf{B}}^{-1}\tilde{\vec{\mathbf{b}}}$

In this appendix, the analytical expressions for $\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}^{-1}\tilde{\vec{\mathbf{b}}}$ in Eq. (8) are given. Thus, no numerical inversion is required.

To inverse block diagonal matrix $\tilde{\mathbf{B}}$, only the inverse of each block of $\tilde{\mathbf{B}}$ is needed. The difference between $\tilde{\mathbf{B}}$ and $\mathbf{B}$ is the added row in Eq. (7). This added row should be considered in the first block of $\tilde{\mathbf{B}}$, i.e., above the $\mathbf{B}_0$ in Eq. (A4),

$$\tilde{\mathbf{B}}_0 = \begin{bmatrix} 1 & \mathbf{0} \\ & \mathbf{B}_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_{N_sN_rN_t} \\ 1 & -1 & \mathbf{0} \end{bmatrix}. \tag{C1}$$

It can be easily seen that the inverse of $\tilde{\mathbf{B}}_0$ is

$$\tilde{\mathbf{B}}_0^{-1} = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{0} & -1 \\ \mathbf{0} & -\mathbf{I}_{N_sN_rN_t} & \mathbf{0} \end{bmatrix}. \tag{C2}$$

Similarly, one can easily check that the inverse of each type of block in $\tilde{\mathbf{B}}$ is

$$\mathbf{B}^{-1}_{1,k} = \begin{bmatrix} \mathbf{0} & 1 & 0 \\ \mathbf{0} & 1 & -1 \\ -\mathbf{I}_{2N_sN_r} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{B}^{-1}_{2,l,k} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix},$$

$$\mathbf{B}^{-1}_{3,k} = \begin{bmatrix} -\mathbf{I}_{2N_s^2} & \mathbf{0} \\ -\mathbf{I}_{N_s} \otimes \mathbf{P}_s \otimes \mathbf{I}_{N_s} & -\mathbf{I}_{2N_s^2} \end{bmatrix},$$

$$\mathbf{B}^{-1}_{4,k} = \begin{bmatrix} -\mathbf{I}_{8N_s^2} & \mathbf{0} \\ -\mathbf{I}_{2N_s} \otimes \mathbf{P}_s \otimes \mathbf{I}_{2N_s} & -\mathbf{I}_{8N_s^2} \end{bmatrix}. \tag{C3}$$

Let $\tilde{\mathbf{A}}_0$ and $\tilde{\vec{\mathbf{b}}}_0$ be

$$\tilde{\mathbf{A}}_0 = \begin{bmatrix} \mathbf{0} & -1 \\ 2\mathbf{M}_0 & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \quad \tilde{\vec{\mathbf{b}}}_0 = \begin{bmatrix} \mathbf{0} \\ 2 \end{bmatrix}. \tag{C4}$$

Then, $\tilde{\mathbf{A}}$ can be seen as vertically concatenating $\tilde{\mathbf{A}}_0$, $\tilde{\mathbf{A}}_{1,k}$, $\tilde{\mathbf{A}}_{2,l,k}$, $\tilde{\mathbf{A}}_{3,k}$, and $\tilde{\mathbf{A}}_{4,k}$ for all k and l. It is noted that $\tilde{\mathbf{A}}_{1,k}$, $\tilde{\mathbf{A}}_{2,l,k}$, $\tilde{\mathbf{A}}_{3,k}$, and $\tilde{\mathbf{A}}_{4,k}$ have one more zero column to the right compared with $\mathbf{A}_{1,k}$, $\mathbf{A}_{2,l,k}$, $\mathbf{A}_{3,k}$, and $\mathbf{A}_{4,k}$. $\tilde{\mathbf{B}}^{-1}$ can be seen as a sparse block diagonal matrix obtained by diagonally concatenating $\tilde{\mathbf{B}}_0^{-1}$, $\mathbf{B}^{-1}_{1,k}$, $\mathbf{B}^{-1}_{2,l,k}$, $\mathbf{B}^{-1}_{3,k}$, and $\mathbf{B}^{-1}_{4,k}$ for all k and l. Thus, $\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}$ can be seen as vertically concatenating $\tilde{\mathbf{B}}_0^{-1}\tilde{\mathbf{A}}_0$, $\mathbf{B}^{-1}_{1,k}\tilde{\mathbf{A}}_{1,k}$, $\mathbf{B}^{-1}_{2,l,k}\tilde{\mathbf{A}}_{2,l,k}$, $\mathbf{B}^{-1}_{3,k}\tilde{\mathbf{A}}_{3,k}$, and $\mathbf{B}^{-1}_{4,k}\tilde{\mathbf{A}}_{4,k}$ for all k and l. By using the expression of each block of $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}^{-1}$, they can be easily computed as

$$\tilde{\mathbf{B}}_0^{-1}\tilde{\mathbf{A}}_0 = \begin{bmatrix} \mathbf{0} & -1 \\ \mathbf{0} & -1 \\ -2M_0 & \mathbf{0} \end{bmatrix},$$

$$\mathbf{B}_{1,k}^{-1}\tilde{\mathbf{A}}_{1,k} = \begin{bmatrix} \vec{b}_J^T(f_k) & 0 \\ \vec{b}_J^T(f_k) & 0 \\ -2\Re(M_{1,k}) & \mathbf{0} \\ -2\Im(M_{1,k}) & \mathbf{0} \end{bmatrix},$$

$$\mathbf{B}_{2,l,k}^{-1}\mathbf{A}_{2,l,k} = \begin{bmatrix} \mathbf{0} & 0 \\ -\vec{e}_l^T\Re(\mathbf{P}_r \otimes \mathbf{F}_z(f_k)) & 0 \\ -\vec{e}_l^T\Im(\mathbf{P}_r \otimes \mathbf{F}_z(f_k)) & 0 \end{bmatrix},$$

$$\mathbf{B}_{3,k}^{-1}\tilde{\mathbf{A}}_{3,k} = \begin{bmatrix} -\bar{\mathbf{A}}_{3,k} & \mathbf{0} \\ \mathbf{A}^i_{3,k,1} & \mathbf{0} \\ -\mathbf{A}^r_{3,k,1} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{A}^i_{3,k,N_s} & \mathbf{0} \\ -\mathbf{A}^r_{3,k,N_s} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_{4,k}^{-1}\tilde{\mathbf{A}}_{4,k} = \begin{bmatrix} -\bar{\mathbf{A}}_{4,k} & \mathbf{0} \\ \mathbf{A}^i_{4,k,R,1} & \mathbf{0} \\ -\mathbf{A}^r_{4,k,R,1} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{A}^i_{4,k,R,N_s} & \mathbf{0} \\ -\mathbf{A}^r_{4,k,R,N_s} & \mathbf{0} \\ \mathbf{A}^i_{4,k,Q,1} & \mathbf{0} \\ -\mathbf{A}^r_{4,k,Q,1} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{A}^i_{4,k,Q,N_s} & \mathbf{0} \\ -\mathbf{A}^r_{4,k,Q,N_s} & \mathbf{0} \end{bmatrix},$$

Similarly, $\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{b}}$ can be seen as vertically concatenating $\tilde{\mathbf{B}}_0^{-1}\vec{\tilde{\mathbf{b}}}_0$, $\mathbf{B}_{1,k}^{-1}\vec{\mathbf{b}}_{1,k}$, $\mathbf{B}_{2,l,k}^{-1}\vec{\mathbf{b}}_{2,l,k}$, $\mathbf{B}_{3,k}^{-1}\vec{\mathbf{b}}_{3,k}$, $\mathbf{B}_{4,k}^{-1}\vec{\mathbf{b}}_{4,k}$, and for all k and l. By using the expression of each block of $\vec{\tilde{\mathbf{b}}}$ and $\tilde{\mathbf{B}}^{-1}$, they can be easily computed as

$$\tilde{\mathbf{B}}_0^{-1}\vec{\tilde{\mathbf{b}}}_0 = \begin{bmatrix} 0 \\ -2 \\ \mathbf{0} \end{bmatrix}; \quad \mathbf{B}_{1,k}^{-1}\vec{\mathbf{b}}_{1,k} = \begin{bmatrix} 1-\tilde{c}_J(f_k) \\ -1-\tilde{c}_J(f_k) \\ \mathbf{0} \end{bmatrix};$$

$$\mathbf{B}_{2,l,k}^{-1}\vec{\mathbf{b}}_{2,l,k} = \begin{bmatrix} C_{l,J}(f_k) \\ 0 \\ 0 \end{bmatrix};$$

and the ith element of the sparse vector, $\mathbf{B}_{3,k}^{-1}\vec{\mathbf{b}}_{3,k}$, is $2(1-\epsilon_s)$ if i or $(i-2N_s^2-N_s)\in\mathbb{S}_{b_3}$; the ith element of the sparse vector, $\mathbf{B}_{4,k}^{-1}\vec{\mathbf{b}}_{4,k}$, is $1/\mathbf{B}(f_k)$ if i, $(i-4N_s^2-N_s)$, $(i-8N_s^2-2N_s)$, or $(i-12N_s^2-3N_s)\in\mathbb{S}_{b_4}$.

Thus, calculating $\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}^{-1}\vec{\tilde{\mathbf{b}}}$ is merely rearranging of the elements sequence in the matrix and does not require extra computational effort in practice.

[1] C.-Y. Chang, A. Siswanto, C.-Y. Ho, T.-K. Yeh, Y.-R. Chen, and 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).

[2] A. Siswanto, C.-Y. Chang, and S. M. Kuo, "Active noise control for headrests," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (IEEE, New York, 2015), pp. 688–692.

[3] J. Bean, N. Schiller, and C. Fuller, "Numerical modeling of an active headrest," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings* (Institute of Noise Control Engineering, Reston, VA, 2017), Vol. 255, pp. 4065–4075.

[4] P. N. Samarasinghe, W. Zhang, and T. D. Abhayapala, "Recent advances in active noise control inside automobile cabins: Toward quieter cars," IEEE Signal Process. Mag. **33**(6), 61–73 (2016).

[5] J. Cheer and S. J. Elliott, "Multichannel control systems for the attenuation of interior road noise in vehicles," Mech. Syst. Signal Process. **60-61**, 753–769 (2015).

[6] N. Miyazaki and Y. Kajikawa, "Head-mounted active noise control system with virtual sensing technique," J. Sound Vib. **339**, 65–83 (2015).

[7] B. Lam, D. Shi, W.-S. Gan, S. J. Elliott, and M. Nishimura, "Active control of broadband sound through the open aperture of a full-sized domestic window," Sci. Rep. **10**(1), 10021 (2020).

[8] B. Lam and W.-S. Gan, "Active acoustic windows: Towards a quieter home," IEEE Potentials **35**(1), 11–18 (2016).

[9] S. Elliott, "Signal processing for active control," in *Signal Processing and its Applications* (Academic, London, 2001), Chap. 6, pp. 271–327.

[10] B. Lam, W.-S. Gan, D. Shi, M. Nishimura, and S. Elliott, "Ten questions concerning active noise control in the built environment," Build. Environ. **200**, 107928 (2021).

[11] D. Y. Shi, B. Lam, and W.-S. Gan, "A novel selective active noise control algorithm to overcome practical implementation issue," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York, 2018), pp. 1130–1134.

[12] C. Shi, T. Murao, D. Shi, B. Lam, and W.-S. Gan, "Open loop active control of noise through open windows," in *Proceedings of Meetings on Acoustics 172ASA* (Acoustical Society of America, Melville, NY, 2016), Vol. 29, p. 030007.

[13] L. Bhan, T. Murao, C. Shi, W.-S. Gan, and S. Elliott, "Feasibility of the full-rank fixed-filter approach in the active control of noise through open windows," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings* (Institute of Noise Control Engineering, Reston, VA, 2016), Vol. 253, pp. 3548–3555.

[14] R. Ranjan, T. Murao, B. Lam, and W.-S. Gan, "Selective active noise control system for open windows using sound classification," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings* (Institute of Noise Control Engineering, Reston, VA, 2016), Vol. 253, pp. 1921–1931.

[15] D. Shi, W.-S. Gan, B. Lam, and S. Wen, "Feedforward selective fixed-filter active noise control: Algorithm and implementation," IEEE/ACM Trans. Audio, Speech, Lang. Process. **28**, 1479–1492 (2020).

[16] W. Jung, S. J. Elliott, and J. Cheer, "Local active control of road noise inside a vehicle," Mech. Syst. Signal Process. **121**, 144–157 (2019).

[17] W. Jung, S. J. Elliott, and J. Cheer, "Combining the remote microphone technique with head-tracking for local active sound control," J. Acoust. Soc. Am. **142**(1), 298–307 (2017).

[18] J. S. Bolton, B. K. Gardner, and T. A. Beauvilain, "Sound cancellation by the use of secondary multipoles," J. Acoust. Soc. Am. **98**(4), 2343–2362 (1995).

[19] T. A. Beauvilain, J. S. Bolton, and B. K. Gardner, "Sound cancellation by the use of secondary multipoles: Experiments," J. Acoust. Soc. Am. **107**(3), 1189–1202 (2000).

[20] S.-C. Huang, C.-H. Ma, Y.-C. Hsu, and M. R. Bai, "Feedforward active noise global control using a linearly constrained beamforming approach," J. Sound Vib. **537**, 117190 (2022).

[21] D. Shi, W.-S. Gan, B. Lam, and X. Shen, "Comb-partitioned frequency-domain constraint adaptive algorithm for active noise control," Signal Process. **188**, 108222 (2021).

[22] D. Shi, W.-S. Gan, B. Lam, S. Wen, and X. Shen, "Optimal output-constrained active noise control based on inverse adaptive modeling leak factor estimate," IEEE/ACM Trans. Audio, Speech, Lang. Process. **29**, 1256–1269 (2021).

[23] J. C. Bermudez and M. H. Costa, "Optimum leakage factor for the MOV-LMS algorithm in nonlinear modeling and control systems," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing* (IEEE, New York, 2002), Vol. 2, pp. II–1393.

[24] W. J. Kozacky and T. Ogunfunmi, "An active noise control algorithm with gain and power constraints on the adaptive filter," EURASIP J. Adv. Signal Process. **2013**(1), 17 (2013).

[25] W. J. Kozacky and T. Ogunfunmi, "A cascaded IIR-FIR adaptive ANC system with output power constraints," Signal Process. **94**, 456–464 (2014).

[26] C. Zhou, H. Zou, and X. Qiu, "A frequency band constrained filtered-x least mean square algorithm for feedback active control systems," J. Acoust. Soc. Am. **148**(4), 1947–1951 (2020).

[27] L. Wu, X. Qiu, and 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).

[28] F. An, Y. Cao, M. Wu, H. Sun, B. Liu, and J. Yang, "Robust Wiener controller design with acoustic feedback for active noise control systems," J. Acoust. Soc. Am. **145**(4), EL291–EL296 (2019).

[29] L. Wu, X. Qiu, and Y. Guo, "A simplified adaptive feedback active noise control system," Appl. Acoust. **81**, 40–46 (2014).

[30] Y. Zhuang and Y. Liu, "Constrained optimal filter design for multi-channel active noise control via convex optimization," J. Acoust. Soc. Am. **150**(4), 2888–2899 (2021).

[31] C. Boultifat, P. Loiseau, P. Chevrel, J. Lohéac, and M. Yagoubi, "FxLMS versus H_∞ control for broadband acoustic noise attenuation in a cavity," IFAC-PapersOnLine **50**(1), 9204–9210 (2017).

[32] B. Rafaely and S. J. Elliott, "H_2/H_∞ active control of sound in a headrest: Design and implementation," IEEE Trans. Control Syst. Technol. **7**(1), 79–84 (1999).

[33] S. Elliott, "Signal processing for active control," in *Signal Processing and its Applications* (Academic, London, 2001), Chap. 5, pp. 33–270.

[34] P. Loiseau, P. Chevrel, M. Yagoubi, and J.-M. Duffal, "Investigating achievable performances for robust broadband active noise control in an enclosure," IEEE Trans. Control Syst. Technol. **27**(1), 426–433 (2019).

[35] P. Loiseau, P. Chevrel, M. Yagoubi, and J.-M. Duffal, "Robust active noise control in a car cabin: Evaluation of achievable performances with a feedback control scheme," Control Eng. Practice **81**, 172–182 (2018).

[36] J.-h. Seo, Y.-c. Park, and D. H. Youn, "Design of feedback active noise control system based on a constrained optimization for headphone/earphone applications," in *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)* (IEEE, New York, 2016), pp. 1–4.

[37] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," Optim. Methods Software **1**1(1-4), 625–653 (1999).

[38] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "On the implementation and usage of SDPT3—A MATLAB software package for semidefinite-quadratic-linear programming, version 4.0," in *Handbook on Semidefinite, Conic and Polynomial Optimization* (Springer, New York, 2012), pp. 715–754.

[39] M. ApS, "The MOSEK optimization toolbox for MATLAB manual. Version 9.0" (2019), available at docs.mosek.com/9.0/toolbox/index.html (Last viewed 10/7/2022).

[40] Y. Zhuang and Y. Liu, "Study on the cone programming reformulation of active noise control filter design in the frequency domain," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings* (Institute of Noise Control Engineering, Reston, VA, 2019), Vol. 260, pp. 126–136.

[41] Y. Zhuang and Y. Liu, "Development and application of dual form conic formulation of multichannel active noise control filter design problem in frequency domain," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings* (Institute of Noise Control Engineering, Reston, VA, 2020), Vol. 261, pp. 676–687.

[42] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Chap. 10.2.2, p. 60, available at http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html (Last viewed 8/3/2022), version 20121115.

[43] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "On the implementation and usage of SDPT3ôA Matlab software package for semidefinite-quadratic-linear programming, version 4.0," in *Handbook on Semidefinite, Conic and Polynomial Optimization* (Springer, Boston, MA, 2012), pp. 715–754.

[44] J. F. Sturm, "Implementation of interior point methods for mixed semidefinite and second order cone optimization problems," Optim. Methods Software **17**(6), 1105–1154 (2002).

[45] R. H. Tütüncü, K.-C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," Math. Program. **95**(2), 189–217 (2003).

[46] A. Ben-Tal and A. Nemirovski, "Lectures on modern convex optimization," in *MOS-SIAM Series on Optimization*, Chap. 6, pp. 377–442, available at epubs.siam.org/doi/abs/10.1137/1.9780898718829.ch6 (Last viewed 10/7/2022).

[47] E. Deadman, N. J. Higham, and R. Ralha, "Blocked Schur algorithms for computing the matrix square root," in *International Workshop on Applied Parallel Computing* (Springer, New York, 2012), pp. 171–182.

[48] A. Ben-Tal and A. Nemirovski, "Lectures on modern convex optimization," in *MOS-SIAM Series on Optimization*, Chap. 3, 79–138, available at epubs.siam.org/doi/abs/10.1137/1.9780898718829.ch3 (Last viewed 10/7/2022).

[49] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004).