Numerical Linear Algebra with Applications

WILEY

RESEARCH ARTICLE OPEN ACCESS

Efficient Solution of Sequences of Parametrized Lyapunov Equations With Applications

Davide Palitta[1] | Zoran Tomljanović[2] | Ivica Nakić[3] | Jens Saak[4]

[1]Università di Bologna, Centro AM2, Dipartimento di Matematica, Bologna, Italy | [2]School of Applied Mathematics and Informatics, J. J. Strossmayer University of Osijek, Osijek, Croatia | [3]Department of Mathematics, Faculty of Science, University of Zagreb, Zagreb, Croatia | [4]Computational Methods in Systems and Control Theory (CSC), Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

Correspondence: Davide Palitta (davide.palitta@unibo.it)

Received: 13 December 2023 | **Revised:** 18 October 2024 | **Accepted:** 23 October 2024

Funding: This work was supported by Hrvatska Zaklada za Znanost (IP-2019-04-6774) and the Gruppo Nazionale per il Calcolo Scientifico (CUP_E53C22001930001).

Keywords: extended Krylov methods | multi-agent systems | parametrized Lyapunov equations | recycling Krylov methods | vibrational systems

ABSTRACT

Sequences of parametrized Lyapunov equations can be encountered in many application settings. Moreover, solutions of such equations are often intermediate steps of an overall procedure whose main goal is the computation of trace(EX), where X denotes the solution of a Lyapunov equation and E is a given matrix. We are interested in addressing problems where the parameter dependency of the coefficient matrix is encoded as a low-rank modification to a *seed*, fixed matrix. We propose two novel numerical procedures that fully exploit such a common structure. The first one builds upon the Sherman-Morrison-Woodbury (SMW) formula and recycling Krylov techniques, and it is well-suited for small dimensional problems as it makes use of dense numerical linear algebra tools. The second algorithm can instead address large-scale problems by relying on state-of-the-art projection techniques based on the extended Krylov subspace. We test the new algorithms on several problems arising in the study of damped vibrational systems and the analyses of output synchronization problems for multi-agent systems. Our results show that the algorithms we propose are superior to state-of-the-art techniques as they are able to remarkably speed up the computation of accurate solutions.

MSC2020 Classification: MSC1, MSC2, MSC3

1 | Introduction and Setting

The main goal of this study is the design of efficient numerical procedures for solving sequences of parameter-dependent Lyapunov equations of the form

$$A(\mathrm{v})X + XA(\mathrm{v})^{\mathrm{T}} = -Q \tag{1}$$

where $A(\mathrm{v})$ has the form

$$A(\mathrm{v}) = A_0 - B_l D(\mathrm{v}) B_r^{\mathrm{T}} \tag{2}$$

Here $A_0, Q \in \mathbb{R}^{n\times n}$, and $B_l, B_r \in \mathbb{R}^{n\times k}$ are fixed matrices, and the matrix $D(\mathrm{v}) = \mathrm{diag}(v_1, v_2, \ldots, v_k)$ contains parameters $v_i \in \mathbb{R}\setminus\{0\}$, for $i = 1\ldots, k$, encoded in the parameter vector $\mathrm{v} = \left[v_1, v_2, \ldots, v_k\right]^{\mathrm{T}} \in (\mathbb{R}\setminus\{0\})^k$ [1]. Note that, without loss of generality, we assume that the parameters in v are nonzero. The number of parameters k is supposed to be very small, $k = \mathcal{O}(1)$, as this is the case in the application settings we are interested in; see Section 3.

Structured equations of the form (1) arise in many problem settings. Indeed, let b_l^i, b_r^i, $i = 1, \ldots, k$, be the columns of the

matrices B_l and B_r, respectively. Then $A(\mathrm{v}) = A_0 - \sum_{i=1}^{k} v_i b_l^i (b_r^i)^{\mathrm{T}} =: A_0 + \sum_{i=1}^{k} v_i A_i$, and hence matrices of the form (2) can describe a class of parameter-dependent matrices with an affine dependency on the parameters. Lyapunov equations with such coefficient matrices naturally arise in the study of vibrational systems [1], parametric model reduction [2], in the design of low gain feedback [3], and the analysis of multi-agent systems [4, 5]. The applications we are going to study more closely are optimal damping of vibrational systems and output-synchronization problems for multi-agent systems.

We are especially interested in devising efficient procedures for calculating $\mathrm{trace}(EX(\mathrm{v}))$, for some matrix E, for many different parameter vectors v, with calculating $X(\mathrm{v})$ only being an intermediate step. For instance, this is the case when computing the H_2-norm of a linear system. An efficient computation of $\mathrm{trace}(EX(\mathrm{v}))$ is crucial for obtaining competitive numerical schemes in terms of both computational performance and memory requirements for the applications mentioned above. Our method also covers efficient calculation of quantities of the form $f(X(\mathrm{v}))$, for a given function f that is assumed to be *additive*, that is, $f(X+Y) = f(X) + f(Y)$. To simplify the notation, in the sequel we will denote $\mathrm{trace}(EX)$ by $f(X)$.

Naturally, one can solve (1) from scratch for any v. For instance, if the problem dimension allows, one can apply the Bartels-Stewart algorithm [6] to each instance of (1). However, this naive procedure does not exploit the attractive structure of (1) and requires the computation of the Schur decomposition of $A(\mathrm{v})$ any time v changes. This approach is not affordable in terms of computational cost if v varies a lot and $f(X(\mathrm{v}))$ needs to be evaluated many times. More sophisticated schemes for (1) can be found in the literature. For instance, the algorithms presented in [1, 7] do exploit the structure of $A(\mathrm{v})$ in (2). Even though these schemes are more performing than a naive application of the Bartels-Stewart algorithm, they are not designed to capitalize on a possible slow variation in v as it often happens within, for example, an optimization routine. Moreover, they can be applied to small dimensional problems only. Various approaches for solving the parameter-dependent Lyapunov equations can be found in the literature, including those based on the conjugate gradient method [8], low-rank updates [9], interpolation on manifolds [10], and low-rank reduced basis method [11, 12]. On the other hand, they are not well-suited for solving sequences of numerous parameter-dependent equations.

Leveraging the structure of the coefficient matrix $A(\mathrm{v})$—$A(\mathrm{v})$ is an affine function in v—we propose two different schemes which are able to fully separate the parameter-independent calculations from the v-dependent computations. The former will be performed once and for all in an *offline* step, whereas the latter operations take place *online*, namely every time v changes.

The first procedure we suggest in this paper builds upon the work in [1, 7], which we enhance with a recycling Krylov technique. This routine makes use of dense linear algebra tools so that it is well suited for small dimensional problems, that is, for moderate values of n. The second routine addresses the large-scale case employing projection methods for linear matrix equations.

Both our algorithms take inspiration from the low-rank update scheme presented in [9], for which the following assumption is needed.

Assumption 1. The solution X_0 to the Lyapunov equation $A_0 X + X A_0^{\mathrm{T}} = -Q$ can be computed efficiently.

1.1 | Synopsis of the Paper

In Section 2 we present the main contribution of this work. In particular, a novel scheme combining the SMW formula with recycling Krylov-like methods is illustrated in Section 2.1. As already mentioned, this scheme is able to efficiently deal with small dimensional problems only as it employs dense numerical linear algebra tools like, for example, the full eigenvalue decomposition of A_0, so that it cannot be applied in the large-scale setting. We address the latter scenario in Section 2.2, where a sophisticated projection technique, based on the extended Krylov subspace (19), is illustrated. In this framework, we will assume to be able to efficiently solve linear systems with A_0. Some details about two application settings where equations of the form (1) are encountered are reported in Section 3. In particular, damped vibrational systems are treated in Section 3.1, whereas in Section 3.2 multi-agent systems are considered. A panel of diverse numerical experiments displaying the effectiveness of our methodology is presented in Section 4, while Section 5 collects our conclusions.

Throughout the paper, we adopt the following notation: The symbol $\otimes$ denotes the Kronecker product, and $\circ$ denotes the Hadamard (component-wise) product. Capital letters, both in roman (A) and italic ($\mathcal{A}$), denote matrices with no particular structure, whereas capital bold letters ($\mathbf{A}$) denote matrices having a Kronecker structure. Given a matrix $X \in \mathbb{R}^{m\times n}$, $\mathrm{vec}(X) \in \mathbb{R}^{mn}$ denotes the vector obtained by stacking the columns of X on top of each other. The matrix I_n is the $n \times n$ identity matrix; the subscript is omitted whenever the dimension of I is clear from the context. The symbol e_i denotes the i-th canonical basis vector of $\mathbb{R}^s$, where s is either specified in the text or clear from the context. The symbol $\delta_{i,j}$ denotes the Kronecker delta, that is, $\delta_{i,j} = 1$ if $i = j$ and zero otherwise.

2 | The Novel Solution Methods

Equation (1) can be written as

$$(A_0 - B_l D(\mathrm{v}) B_r^{\mathrm{T}}) X(\mathrm{v}) + X(\mathrm{v})(A_0 - B_l D(\mathrm{v}) B_r^{\mathrm{T}})^{\mathrm{T}} = -Q$$

By exploiting the low rank of $B_l D(\mathrm{v}) B_r^{\mathrm{T}}$ and using the low-rank update approach presented in [9], we can also rewrite X as the sum of a fixed matrix and a low-rank update term, that is,

$$X(\mathrm{v}) = X_0 + X_\delta(\mathrm{v}) \tag{3}$$

where X_0 and $X_\delta(\mathrm{v})$ are as follows. The matrix X_0 is independent of the parameter vector v, and it solves the Lyapunov equation

$$A_0 X_0 + X_0 A_0^{\mathrm{T}} = -Q \tag{4}$$

whereas $X_\delta(\mathrm{v})$ is such that

$$A(\mathrm{v})X_\delta(\mathrm{v}) + X_\delta(\mathrm{v})A(\mathrm{v})^{\mathrm{T}} = B_l D(\mathrm{v})B_r^{\mathrm{T}} X_0 + X_0 B_r D(\mathrm{v})B_l^{\mathrm{T}} \quad (5)$$

Assumption 1 implies that X_0 can be efficiently calculated.

We notice that the right-hand side in (5) can be written as

$$B_l D(\mathrm{v})B_r^{\mathrm{T}} X_0 + X_0 B_r D(\mathrm{v})B_l^{\mathrm{T}} = [X_0 B_r, B_l] \begin{bmatrix} 0 & D(\mathrm{v}) \\ D(\mathrm{v}) & 0 \end{bmatrix} [X_0 B_r, B_l]^{\mathrm{T}} = P\big(\mathcal{I}_2 \otimes D(\mathrm{v})\big)P^{\mathrm{T}} \quad (6)$$

where $P = [X_0 B_r, B_l]$ has rank $2k$, and $\mathcal{I}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. This formulation of the right-hand side will be crucial for the projection-based method we design for large-scale problems in Section 2.2.

In the next sections, we show how to efficiently compute $X_\delta(\mathrm{v})$. In particular, it turns out that many of the required operations do not depend on v so that they can be carried out in the offline stage. Therefore, we split the solution process into offline and online phases, where only the operations that actually depend on v are performed online.

2.1 | Small-Scale Setting: SMW and Recycling Krylov Methods

In this section we address the numerical solution of the Lyapunov equation (5) assuming the problem dimension n to be small, namely it is such that dense linear algebra operations costing up to $\mathcal{O}(n^3)$ flops are feasible.

Even though there are many efficient methods for solving Lyapunov equations of moderate dimensions, see, for example [13] and references therein, we want to efficiently solve (5) for a sequence of parameter vectors v.

The strategy proposed in this section builds upon the work in [1, 7]. We extend their approach to the more general class of problems of the form (1), and we allow for more than one parameter, that is, k can be larger than one. The scheme we are going to present relies on the computation of the full eigenvalue decomposition of A_0, namely $A_0 = Q_0\Lambda_0 Q_0^{-1}$ with $Q_0 \in \mathbb{C}^{n\times n}$ and $\Lambda_0 = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_i \in \mathbb{C}$, $i = 1, \ldots, n$.

Considering the notation introduced in (6), we can write.

$$\begin{aligned} &A_0 X_\delta(\mathrm{v}) + X_\delta(\mathrm{v})A_0^{\mathrm{T}} - B_l D(\mathrm{v})B_r^{\mathrm{T}} X_\delta(\mathrm{v}) \\ &\quad - X_\delta(\mathrm{v})B_r^{\mathrm{T}} D(\mathrm{v})B_l = P\big(\mathcal{I}_2 \otimes D(\mathrm{v})\big)P^{\mathrm{T}} \end{aligned}$$

If $A_0 = Q_0\Lambda_0 Q_0^{-1}$, $\Lambda_0 = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, we have

$$\begin{aligned} &\Lambda_0 \widetilde{X}_\delta(\mathrm{v}) + \widetilde{X}_\delta(\mathrm{v})\Lambda_0 - Q_0^{-1} B_l D(\mathrm{v})B_r^{\mathrm{T}} Q_0 \widetilde{X}_\delta(\mathrm{v}) \\ &\quad - \widetilde{X}_\delta(\mathrm{v})Q_0^{\mathrm{T}} B_r^{\mathrm{T}} D(\mathrm{v})B_l Q_0^{-\mathrm{T}} = Q_0^{-1} P\big(\mathcal{I}_2 \otimes D(\mathrm{v})\big)P^{\mathrm{T}} Q_0^{-\mathrm{T}} \end{aligned}$$

where $X_\delta(\mathrm{v}) = Q_0 \widetilde{X}_\delta(\mathrm{v}) Q_0^{\mathrm{T}}$. Solving the equation above is equivalent to inverting a linear operator composed of two parts. The first one, namely the Lyapunov operator defined by Λ_0, is easy to invert thanks to the diagonal pattern of Λ_0, whereas the second one has a low-rank structure. This is the perfect scenario for applying the Sherman-Morrison-Woodbury (SMW) formula. We remind the reader that for a standard linear system of the form

$$(\mathbf{L} - \mathbf{M}\mathbf{D}\mathbf{N}^{\mathrm{T}})x = b$$

where $\mathbf{M}$ and $\mathbf{N}$ have low rank, the SMW formula says that

$$x = \mathbf{L}^{-1}b + \mathbf{L}^{-1}\mathbf{M}(\mathbf{D}^{-1} - \mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M})^{-1}\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}b \quad (7)$$

see, for example [14]. By applying the matrix-oriented SMW formula (see, e.g., [15]) in our setting, the matrix $\widetilde{X}_\delta(\mathrm{v})$ can be written as the summation of two components $\widetilde{X}_\delta(\mathrm{v}) = \widetilde{Z}(\mathrm{v}) + \widetilde{W}(\mathrm{v})$ as well, where $\widetilde{Z}(\mathrm{v})$ is such that

$$\Lambda_0 \widetilde{Z}(\mathrm{v}) + \widetilde{Z}(\mathrm{v})\Lambda_0 = Q_0^{-1} P\big(\mathcal{I}_2 \otimes D(\mathrm{v})\big)P^{\mathrm{T}} Q_0^{-\mathrm{T}} \quad (8)$$

whereas the expression of $\widetilde{W}(\mathrm{v})$ is more intricate. Up to a similarity transformation, the matrices $\widetilde{Z}(\mathrm{v})$ and $\widetilde{W}(\mathrm{v})$ are the matrix counterparts of the first and second term in the right-hand side of (7), respectively.

Our goal is thus to compute

$$X_\delta(\mathrm{v}) = Z(\mathrm{v}) + W(\mathrm{v}) = Q_0 \widetilde{Z}(\mathrm{v}) Q_0^{\mathrm{T}} + Q_0 \widetilde{W}(\mathrm{v}) Q_0^{\mathrm{T}} \quad (9)$$

efficiently by taking into account the structure of the problem and the dependency on the parameters in v.

The use of the Sherman-Morrison-Woodbury formula in the matrix setting is not new; see, for example [15, 16], [17, Section 3], [18, Section 5], [19, 20]. However, applying the general-purpose schemes presented in these papers to our parameter-dependent framework would not lead to an efficient solution process. Indeed, this would be equivalent to explicitly computing $X_\delta(\mathrm{v})$ every time v changes. Our main goal here is to divide the operations that depend on v from those that do not, as much as possible, such that the v-independent calculations can be performed offline, only once. In the interest of the reader, we thus spell out the details of our procedure, step by step, in spite of some similarities with prior work.

Step 1: Compute $Z(\mathrm{v})$. The matrix $Z(\mathrm{v})$ is such that $Z(\mathrm{v}) = Q_0 \widetilde{Z}(\mathrm{v}) Q_0^{\mathrm{T}}$ where $\widetilde{Z}(\mathrm{v})$ solves the Lyapunov equation (8). Since $\Lambda_0 = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, by defining the Cauchy matrix $\mathcal{L} \in \mathbb{C}^{n\times n}$ whose (i, j)-th entry is given by $\mathcal{L}_{i,j} = 1/(\lambda_i + \lambda_j)$, we can write

$$Z(\mathrm{v}) = Q_0\big(\mathcal{L} \circ \big(Q_0^{-1} P\big(\mathcal{I}_2 \otimes D(\mathrm{v})\big)P^{\mathrm{T}} Q_0^{-\mathrm{T}}\big)\big)Q_0^{\mathrm{T}} \quad (10)$$

The matrix $\mathcal{I}_2 \otimes D(\mathrm{v})$ can be written as

$$\mathcal{I}_2 \otimes D(\mathrm{v}) = \sum_{i=1}^{k} v_i (e_{k+i} e_i^{\mathrm{T}} + e_i e_{k+i}^{\mathrm{T}})$$

and by plugging this expression into (10), we get

$$Z(\mathrm{v}) = \sum_{i=1}^{k} v_i Z_i = \sum_{i=1}^{k} v_i Q_0 \widetilde{Z}_i Q_0^{\mathrm{T}},$$
$$\widetilde{Z}_i := \mathcal{L}\circ\left(Q_0^{-1} P\left(e_{k+i}e_i^{\mathrm{T}} + e_i e_{k+i}^{\mathrm{T}}\right) P^{\mathrm{T}} Q_0^{-\mathrm{T}}\right) \tag{11}$$

We can thus compute the matrices Z_i before any change in the parameters v takes place. Whenever we need $Z(\mathrm{v})$, we just compute the linear combination in (11).

Step 2: Compute $W(\mathrm{v})$. As we already mentioned, the expression for $W(\mathrm{v})$ is more involved than the one for $Z(\mathrm{v})$. In particular, the computation of the former goes through the solution of a $2nk \times 2nk$ linear system along with some further minor steps; see, for example [15, Algorithm 0].

In the following, we show how to efficiently construct the right-hand side (**Step 2.1**) and the coefficient matrix (**Step 2.2**) of this linear system. Moreover, its solution will be carried out by an iterative method for which we design a structure-aware preconditioner (**Step 2.3**). We then illustrate the final computations for obtaining $W(\mathrm{v})$ (**Step 2.4**).

Step 2.1: The right-hand side. In the linear system setting, this step corresponds to computing the vector $\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}b$ in (7). In principle, one could look at the Kronecker form of (5) and realize that we could set $\mathbf{N} = \left[Q_0^{\mathrm{T}}B_r \otimes I, I \otimes Q_0^{\mathrm{T}}B_r\right]$ in our setting, whereas $\mathbf{L}^{-1}b$ corresponds to the vectorized form of $\widetilde{Z}(\mathrm{v})$. Therefore, thanks to the properties of the Kronecker product, we can write

$$\mathbf{N}^{\mathrm{T}}\mathrm{vec}(\widetilde{Z}(\mathrm{v})) = \mathbf{N}^{\mathrm{T}}\left(\sum_{i=1}^{k} v_i \mathrm{vec}\left(\widetilde{Z}_i\right)\right) = \sum_{i=1}^{k} v_i \begin{bmatrix} \mathrm{vec}\left(\widetilde{Z}_i Q_0^{\mathrm{T}} B_r\right) \\ \mathrm{vec}\left(B_r^{\mathrm{T}} Q_0 \widetilde{Z}_i\right) \end{bmatrix}$$

We can thus construct the quantities $\mathrm{vec}\left(\widetilde{Z}_i Q_0^{\mathrm{T}} B_r\right)$, $\mathrm{vec}\left(B_r^{\mathrm{T}} Q_0 \widetilde{Z}_i\right)$ once, and compute the linear combination above whenever a change in the parameters v_i's takes place.

Step 2.2: The coefficient matrix. We now focus on the solution of the inner linear system involved in the SMW formula. In the notation of (7), this corresponds to solving the linear system

$$\left(\mathbf{D} - \mathbf{N}^{\mathrm{T}}\mathbf{L}\mathbf{M}\right) w = \mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}b \tag{12}$$

In our context, this amounts to a $2nk \times 2nk$ linear system where $\mathbf{D} = \mathbf{D}(\mathrm{v})$ is a parameter-dependent diagonal matrix such that $\mathbf{D}(\mathrm{v})^{-1} = \begin{bmatrix} D(\mathrm{v})^{-1} \otimes I & \\ & I \otimes D(\mathrm{v})^{-1} \end{bmatrix}$. The structure of $\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}$ is more involved. A first study of its structure can be found in [1]. Unlike what is done in [1], here we exploit the diagonal pattern of $\mathbf{L}^{-1} = (\Lambda_0 \otimes I + I \otimes \Lambda_0)^{-1}$, which leads to a useful representation of the blocks of $\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}$ when the latter is viewed as a 2×2 block matrix. Such a block structure will help us to design efficient preconditioners for the solution of the linear system in (12) as well (see **Step 2.3**).

Let $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2]$ and $\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2]$ where $\mathbf{M}_1 = Q_0^{-1}B_l \otimes I$, $\mathbf{M}_2 = I \otimes Q_0^{-1}B_l$, $\mathbf{N}_1 = Q_0^{\mathrm{T}}B_r \otimes I$, and $\mathbf{N}_2 = I \otimes Q_0^{\mathrm{T}}B_r$. Then we can write

$$\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M} = \begin{bmatrix} \mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 & \mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2 \\ \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 & \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2 \end{bmatrix}$$

We first focus on the $(1,1)$-block, namely $\mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 \in \mathbb{C}^{nk\times nk}$. By recalling that the h-th basis vector of $\mathbb{C}^{nk}$ can be written as $e_h = \mathrm{vec}(e_s e_t^{\mathrm{T}})$, with e_s and e_t being the canonical vectors in $\mathbb{C}^n$ and $\mathbb{C}^k$, respectively, and $h = (t-1)n + s$, we can write the (i,j)-th entry of $\mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 \in \mathbb{C}^{nk\times nk}$ as

$$e_i^{\mathrm{T}}\mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 e_j = \mathrm{vec}(e_s e_t^{\mathrm{T}})^{\mathrm{T}}\mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 \mathrm{vec}(e_q e_p^{\mathrm{T}})$$

where $i = (t-1)n + s$ and $j = (p-1)n + q$. Since $\mathbf{N}_1$ and $\mathbf{M}_1$ have a Kronecker structure, we have

$$\begin{aligned} e_i^{\mathrm{T}}\mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 e_j &= \mathrm{vec}(e_s e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0)^{\mathrm{T}}\mathbf{L}^{-1}\mathrm{vec}(e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}}) \\ &= \mathrm{vec}(e_s e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0)^{\mathrm{T}}\mathrm{vec}(\mathcal{L}\circ(e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}})) \\ &= \mathrm{trace}\left(Q_0^{\mathrm{T}} B_r e_t e_s^{\mathrm{T}} (\mathcal{L}\circ(e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}}))\right) \\ &= e_s^{\mathrm{T}}(\mathcal{L}\circ(e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}})) Q_0^{\mathrm{T}} B_r e_t \end{aligned}$$

where the last equality holds thanks to the cyclic property of the trace operator. Now, by applying the following property of the Hadamard product

$$x^{\mathrm{T}}(A\circ B)y = \mathrm{trace}(\mathrm{diag}(x) A \mathrm{diag}(y) B^{\mathrm{T}})$$

we get

$$\begin{aligned} e_i^{\mathrm{T}}\mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 e_j &= \mathrm{trace}\left(\mathrm{diag}(e_s)\mathcal{L}\mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) Q_0^{-1} B_l e_p e_q^{\mathrm{T}}\right) \\ &= e_q^{\mathrm{T}}\mathrm{diag}(e_s)\mathcal{L}\mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) Q_0^{-1} B_l e_p \\ &= e_q^{\mathrm{T}} e_s e_s^{\mathrm{T}}\mathcal{L}\mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) Q_0^{-1} B_l e_p \\ &= \delta_{q,s}\cdot e_s^{\mathrm{T}}\mathcal{L}\mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) Q_0^{-1} B_l e_p \\ &= \delta_{q,s}\cdot e_s^{\mathrm{T}}\left(\mathcal{L}\left((Q_0^{\mathrm{T}} B_r e_t)\circ(Q_0^{-1} B_l e_p)\right)\right) \end{aligned} \tag{13}$$

where $\delta_{q,s}$ denotes the Kronecker delta. The relation in (13) shows that $\mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1$ is a $k \times k$ block matrix whose blocks are all diagonal. In particular, the block in the (t,p) position is given by the matrix $\mathrm{diag}\left(\mathcal{L}\left((Q_0^{\mathrm{T}} B_r e_t)\circ(Q_0^{-1} B_l e_p)\right)\right)$.

We now derive the structure of the $(2,2)$-block $\mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2$. Following the same reasoning as before, we have

$$\begin{aligned} e_i^{\mathrm{T}}\mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2 e_j &= \mathrm{vec}(Q_0^{\mathrm{T}} B_r e_t e_s^{\mathrm{T}})^{\mathrm{T}}\mathbf{L}^{-1}\mathrm{vec}(Q_0^{-1} B_l e_p e_q^{\mathrm{T}}) \\ &= \mathrm{vec}(Q_0^{\mathrm{T}} B_r e_t e_s^{\mathrm{T}})^{\mathrm{T}}\mathrm{vec}(\mathcal{L}\circ(Q_0^{-1} B_l e_p e_q^{\mathrm{T}})) \\ &= \mathrm{trace}\left(e_s e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0 (\mathcal{L}\circ(Q_0^{-1} B_l e_p e_q^{\mathrm{T}}))\right) \\ &= e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0 (\mathcal{L}\circ(Q_0^{-1} B_l e_p e_q^{\mathrm{T}})) e_s \end{aligned}$$

Notice that in the expression above we wrote $e_i = \mathrm{vec}(e_t e_s^{\mathrm{T}})$, with e_t, and e_s canonical vectors in $\mathbb{C}^k$ and $\mathbb{C}^n$, respectively, so that $i = (s-1)k + t$. Similarly, $j = (q-1)k + p$. As before, we can write

$$\begin{aligned} e_i^{\mathrm{T}}\mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2 e_j &= \mathrm{trace}\left(\mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t)\mathcal{L}\mathrm{diag}(e_s) e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}}\right) \\ &= e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}}\mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t)\mathcal{L} e_s e_s^{\mathrm{T}} e_q \end{aligned}$$

$$= \delta_{s,q} \cdot e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}} \mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) \mathcal{L} e_s$$
$$= \delta_{s,q} \cdot \left((Q_0^{-1} B_l e_p) \circ (Q_0^{\mathrm{T}} B_r e_t)\right)^{\mathrm{T}} \mathcal{L} e_s$$

Therefore, $\mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ is a block diagonal matrix with n diagonal blocks of order k. In particular, the (t, p)-th entry of the s-th diagonal block is given by $\left((Q_0^{-1} B_l e_p) \circ (Q_0^{\mathrm{T}} B_r e_t)\right)^{\mathrm{T}} \mathcal{L} e_s$.

Unlike its diagonal blocks, the off-diagonal blocks of $\mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$, namely $\mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$, and $\mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$, do not possess a structured sparsity pattern. Nevertheless, we can still apply the same strategy we employed above to construct $\mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$ and $\mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ while avoiding the explicit computation of $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{N}_1$, and $\mathbf{N}_2$. If $i = (s-1)k + t$ and $j = (p-1)n + q$, we have

$$\begin{aligned}
e_i^{\mathrm{T}} \mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1 e_j &= \mathrm{vec}(Q_0^{\mathrm{T}} B_r e_t e_s^{\mathrm{T}})^{\mathrm{T}} \mathbf{L}^{-1} \mathrm{vec}(e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}}) \\
&= \mathrm{vec}(Q_0^{\mathrm{T}} B_r e_t e_s^{\mathrm{T}})^{\mathrm{T}} \mathrm{vec}(\mathcal{L} \circ (e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}})) \\
&= \mathrm{trace}\Big(e_s e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0 (\mathcal{L} \circ (e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}})) \Big) \\
&= e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0 (\mathcal{L} \circ (e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}})) e_s \\
&= \mathrm{trace}\Big(\mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) \mathcal{L} \mathrm{diag}(e_s) Q_0^{-1} B_l e_p e_q^{\mathrm{T}} \Big) \\
&= e_q^{\mathrm{T}} \mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) \mathcal{L} \mathrm{diag}(e_s) Q_0^{-1} B_l e_p \\
&= (e_q^{\mathrm{T}} Q_0^{\mathrm{T}} B_r e_t)(e_q^{\mathrm{T}} \mathcal{L} e_s)(e_s^{\mathrm{T}} Q_0^{-1} B_l e_p)
\end{aligned} \tag{14}$$

Similarly, if $i = (t-1)n + s$ and $j = (q-1)k + p$, it holds

$$\begin{aligned}
e_i^{\mathrm{T}} \mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2 e_j &= \mathrm{vec}(e_s e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0)^{\mathrm{T}} \mathbf{L}^{-1} \mathrm{vec}(Q_0^{-1} B_l e_p e_q^{\mathrm{T}}) \\
&= \mathrm{vec}(e_s e_t^{\mathrm{T}} B_r^{\mathrm{T}} Q_0)^{\mathrm{T}} \mathrm{vec}(\mathcal{L} \circ (Q_0^{-1} B_l e_p e_q^{\mathrm{T}})) \\
&= \mathrm{trace}\Big(Q_0^{\mathrm{T}} B_r e_t e_s^{\mathrm{T}} (\mathcal{L} \circ (Q_0^{-1} B_l e_p e_q^{\mathrm{T}})) \Big) \\
&= e_s^{\mathrm{T}} (\mathcal{L} \circ (Q_0^{-1} B_l e_p e_q^{\mathrm{T}})) Q_0^{\mathrm{T}} B_r e_t \\
&= \mathrm{trace}\Big(\mathrm{diag}(e_s) \mathcal{L} \mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) e_q e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}} \Big) \\
&= e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}} \mathrm{diag}(e_s) \mathcal{L} \mathrm{diag}(Q_0^{\mathrm{T}} B_r e_t) e_q \\
&= (e_p^{\mathrm{T}} B_l^{\mathrm{T}} Q_0^{-\mathrm{T}} e_s)(e_s^{\mathrm{T}} \mathcal{L} e_q)(e_q^{\mathrm{T}} Q_0^{\mathrm{T}} B_r e_t)
\end{aligned} \tag{15}$$

The construction of $\mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$ can be carried out before starting changing v. Once this task is accomplished, we have to solve a $2nk \times 2nk$ linear system with $\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$ every time we have to solve (1) for a different v.

Step 2.3: Solving the SMW inner linear system. We want to efficiently compute the vector $w(\mathrm{v}) \in \mathbb{C}^{2nk}$ such that

$$\left(\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}\right) w(\mathrm{v}) = \sum_{i=1}^{k} v_i \begin{bmatrix} \mathrm{vec}\left(\widetilde{Z}_i Q_0^{\mathrm{T}} B_r\right) \\ \mathrm{vec}\left(B_r^{\mathrm{T}} Q_0 \widetilde{Z}_i\right) \end{bmatrix} \tag{16}$$

We assume the coefficient matrix and the right-hand side in (16) to slowly change with v. For instance, if (1) is embedded in a parameter optimization procedure, the parameters $\mathrm{v}_\ell = [v_1^{(\ell)}, \ldots, v_k^{(\ell)}]^{\mathrm{T}}$ in one step do not dramatically differ from the ones used in the previous step, namely $\mathrm{v}_{\ell-1} = [v_1^{(\ell-1)}, \ldots, v_k^{(\ell-1)}]^{\mathrm{T}}$, especially in the later steps of the adopted optimization procedure. For a sequence of linear systems of this nature, recycling Krylov techniques represent one of the most valid families of solvers available in the literature. See, for example [21, 22] and the recent survey paper [23].

We employ the GCRO-DR method proposed in [22] to solve the sequence of linear systems in (16). See [24, 25] for a Matlab implementation and a thorough discussion on certain computational aspects of this algorithm.

Once the $(\ell - 1)$-th linear system is solved, GCRO-DR uses the space spanned by s_ℓ approximate eigenvectors of $\mathbf{D}(\mathrm{v}_{\ell-1})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$ to enhance the solution of the ℓ-th linear system

$$\left(\mathbf{D}(\mathrm{v}_\ell)^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}\right) w(\mathrm{v}_\ell) = \sum_{i=1}^{k} v_i^{(\ell)} \begin{bmatrix} \mathrm{vec}\left(\widetilde{Z}_i Q_0^{\mathrm{T}} B_r\right) \\ \mathrm{vec}\left(B_r^{\mathrm{T}} Q_0 \widetilde{Z}_i\right) \end{bmatrix}$$

We employ the s_ℓ eigenvectors corresponding to the s_ℓ eigenvalues of smallest magnitude[2], but different approximate eigenvectors can be used as well; see [22, Section 2.4].

The convergence of GCRO-DR is driven by the spectral properties of $\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$. However, the latter ones are extremely tricky to identify in general. Only by allowing further assumptions on B_l, B_r, and A_0, we may come up with sensible statements. For instance, if $B_l = B_r$ and A_0 is Hermitian, $\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$ is Hermitian too, and GCRO-DR can take advantage of that; see [22, Theorem 3.1]. Similarly, if, in addition, all the parameters v_i's are, for example, strictly positive and A_0 is stable, then $\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$ is also positive definite. GCRO-DR can take advantage of this property as well.

GCRO-DR allows for preconditioning. In particular, thanks to the pattern of $\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$ we are able to design efficient preconditioning operators, which can be cheaply tuned to address the variation in the parameters.

We have

$$\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M} = \begin{bmatrix} D(\mathrm{v})^{-1} \otimes I - \mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1 & \mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2 \\ \mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1 & I \otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2 \end{bmatrix}$$

and we want to maintain the (2×2)-block structure in the preconditioner as well.

In $\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$, only the diagonal blocks $D(\mathrm{v})^{-1} \otimes I - \mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$ and $I \otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ depend on the current parameters. Moreover, both $D(\mathrm{v})^{-1} \otimes I$ and $I \otimes D(\mathrm{v})^{-1}$ are diagonal matrices so that the sparsity pattern of $\mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$ and $\mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ is preserved. In particular, $D(\mathrm{v})^{-1} \otimes I - \mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$ is still a $k \times k$ block matrix with diagonal blocks whereas $I \otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ is block-diagonal with $k \times k$ blocks on the diagonal. By exploiting such a significant sparsity pattern, we are thus able to efficiently solve linear systems with $D(\mathrm{v})^{-1} \otimes I - \mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$ and $I \otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ by a sparse direct method, regardless of the change in v.

The off-diagonal blocks of $\mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}$, namely $\mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ and $\mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$, are dense in general. We approximate these blocks by means of their truncated SVD (TSVD) of order $\overline{p}$, for a user-defined $\overline{p} > 0$. Notice that these TSVDs can be computed once and for all, since neither $\mathbf{N}_1^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_2$ nor $\mathbf{N}_2^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_1$ depend on v. We denote the results of the TSVD by $\mathcal{N}_1 \mathcal{M}_2^{\mathrm{T}}$ and $\mathcal{N}_2 \mathcal{M}_1^{\mathrm{T}}$

where $\mathcal{N}_i, \mathcal{M}_i \in \mathbb{C}^{nk\times\bar{p}}$, $i = 1, 2$, and $\mathcal{N}_1\mathcal{M}_2^{\mathrm{T}} \approx \mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2$ whereas $\mathcal{N}_2\mathcal{M}_1^{\mathrm{T}} \approx \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1$.

Consequently, the preconditioning operator we employ is

$$\mathcal{P}(\mathrm{v}) = \begin{bmatrix} D(\mathrm{v})^{-1}\otimes I - \mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 & \mathcal{N}_1\mathcal{M}_2^{\mathrm{T}} \\ \mathcal{N}_2\mathcal{M}_1^{\mathrm{T}} & I\otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2 \end{bmatrix} \approx \mathbf{D}(\mathrm{v})^{-1} - \mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M} \tag{17}$$

We use right preconditioning within GCRO-DR so that at each iteration a linear system with $\mathcal{P}(\mathrm{v})$ needs to be solved. However, such a task is particularly cheap. Indeed, thanks to the 2×2 block structure of $\mathcal{P}(\mathrm{v})$, we define its Schur complement

$$\mathcal{S}(\mathrm{v}) = D(\mathrm{v})^{-1}\otimes I - \mathbf{N}_1^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_1 - \mathcal{N}_1\mathcal{M}_2^{\mathrm{T}}\left(I\otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2\right)^{-1}\mathcal{N}_2\mathcal{M}_1^{\mathrm{T}}$$

and, since $\mathcal{N}_1\mathcal{M}_2^{\mathrm{T}}\left(I\otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2\right)^{-1}\mathcal{N}_2\mathcal{M}_1^{\mathrm{T}}$ has rank $\bar{p}$, we can employ the SMW formula to efficiently solve linear systems with $\mathcal{S}(\mathrm{v})$. We thus have

$$\mathcal{P}(\mathrm{v})^{-1}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \mathcal{S}(\mathrm{v})^{-1}\left(v_1 - \mathcal{N}_1\mathcal{M}_2^{\mathrm{T}}\left(I\otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2\right)^{-1}v_2\right) \\ \left(I\otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2\right)^{-1}\left(\mathcal{N}_2\mathcal{M}_1^{\mathrm{T}}\mathcal{S}(\mathrm{v})^{-1}\left(\mathcal{N}_1\mathcal{M}_2^{\mathrm{T}}\left(I\otimes D(\mathrm{v})^{-1} - \mathbf{N}_2^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}_2\right)^{-1}v_2 - v_1\right) + v_2\right) \end{bmatrix}$$

Step 2.4: Final computations. Once the linear system in (16) is solved and the vector $w(\mathrm{v}) \in \mathbb{C}^{2nk}$ is computed, we proceed with the remaining operations to compute $X_\delta(\mathrm{v})$. We first write

$$w(\mathrm{v}) = \begin{bmatrix} w_1(\mathrm{v}) \\ w_2(\mathrm{v}) \end{bmatrix} = \begin{bmatrix} \mathrm{vec}(W_1(\mathrm{v})) \\ \mathrm{vec}(W_2(\mathrm{v})) \end{bmatrix}, \quad W_1(\mathrm{v}) \in \mathbb{C}^{n\times k}, W_2(\mathrm{v}) \in \mathbb{C}^{k\times n}$$

.

Then the matrix $\widetilde{W}(\mathrm{v})$ in (9) is such that

$$\widetilde{W}(\mathrm{v}) = \mathcal{L}\circ\left(W_1(\mathrm{v})B_l^{\mathrm{T}}Q_0^{-\mathrm{T}} + Q_0^{-1}B_lW_2(\mathrm{v})\right)$$

This operation corresponds to performing $\mathbf{L}^{-1}\mathbf{M}w$ in the linear system setting (7). By the change of coordinates with Q_0 we retrieve the second component of the solution $X_\delta(\mathrm{v}) = Z_\delta(\mathrm{v}) + W_\delta(\mathrm{v})$, namely $W(\mathrm{v}) = Q_0\widetilde{W}(\mathrm{v})Q_0^{\mathrm{T}}$.

The solution $X(\mathrm{v})$ to (1), in case of moderate n, can thus be computed as follows:

$$\begin{aligned} X(\mathrm{v}) &= X_0 + X_\delta(\mathrm{v}) = X_0 + Z(\mathrm{v}) + W(\mathrm{v}) \\ &= X_0 + \sum_{i=1}^{k} v_i^{(\ell)}Z_i + Q_0\left(\mathcal{L}\circ\left(W_1(\mathrm{v})B_l^{\mathrm{T}}Q_0^{-\mathrm{T}} + Q_0^{-1}B_lW_2(\mathrm{v})\right)\right)Q_0^{\mathrm{T}} \end{aligned} \tag{18}$$

so that

$$f(X(\mathrm{v})) = f(X_0) + \sum_{i=1}^{k} v_i^{(\ell)}f(Z_i) + f\left(Q_0\left(\mathcal{L}\circ\left(W_1(\mathrm{v})B_l^{\mathrm{T}}Q_0^{-\mathrm{T}} + Q_0^{-1}B_lW_2(\mathrm{v})\right)\right)Q_0^{\mathrm{T}}\right)$$

In Table 1, we summarize the operations that can be performed offline and online. Notice that the $\mathcal{O}(n^3)$ flops needed to compute $W(\mathrm{v})$ come from the final matrix-matrix multiplications by Q_0 and Q_0^{T} in (18). These operations can be avoided for $f = \mathrm{trace}$ and Q_0 orthogonal, thus reducing the cost of computing $W(\mathrm{v})$ to $\mathcal{O}(n^2)$ flops.

We conclude this section by recalling the reader that $w(\mathrm{v})$ has been computed by GCRO-DR, an iterative method that has been run up to a certain tolerance on the relative residual norm associated to (16). Therefore, we can think of $w(\mathrm{v})$ as being of the form $w(\mathrm{v}) = w_*(\mathrm{v}) + \ell(\mathrm{v})$, where $w_*(\mathrm{v})$ is the exact solution to (16) whereas $\ell(\mathrm{v})$ is the error vector coming from the computation of $w(\mathrm{v})$. The magnitude of $\ell(\mathrm{v})$ plays an important role in the overall accuracy that can be attained by $X(\mathrm{v})$ in (18). In particular, by writing $\ell(\mathrm{v}) = [\ell_1(\mathrm{v}); \ell_2(\mathrm{v})]$, $\ell_1(\mathrm{v}), \ell_2(\mathrm{v}) \in \mathbb{C}^{nk}$, and $E_1(\mathrm{v}) \in \mathbb{C}^{n\times k}$, $\ell_1(\mathrm{v}) = \mathrm{vec}(E_1(\mathrm{v}))$, $E_2(\mathrm{v}) \in \mathbb{C}^{k\times n}$, $\ell_2(\mathrm{v}) = \mathrm{vec}(E_2(\mathrm{v}))$, a direct computation shows that we can write

$$X(\mathrm{v}) = X_0 + Z(\mathrm{v}) + W_*(\mathrm{v}) + E(\mathrm{v})$$

where $E(\mathrm{v}) = Q_0\left(\mathcal{L}\circ\left(E_1(\mathrm{v})B_l^{\mathrm{T}}Q_0^{-\mathrm{T}} + Q_0^{-1}B_lE_2(\mathrm{v})^{\mathrm{T}}\right)\right)Q_0^{\mathrm{T}}$. To bound $\|E(\mathrm{v})\|_F$, it may be more convenient to look at $E(\mathrm{v})$ as the solution to the following Lyappunov equation

$$A_0E(\mathrm{v}) + E(\mathrm{v})A_0^{\mathrm{T}} = Q_0E_1(\mathrm{v})B_l^{\mathrm{T}} + B_lE_2(\mathrm{v})^{\mathrm{T}}Q_0^{\mathrm{T}}$$

TABLE 1 | Offline and online operations involved in the computation of the solution $X(\mathrm{v})$ to (1) for a moderate n. m denotes the number of iterations needed by GCRO-DR to converge.

Offline		Online	
Operation	**Flops**	**Operation**	**Flops**
Compute X_0 and P	$\mathcal{O}(n^3)$	Compute $Z(\mathrm{v})$	$\mathcal{O}(kn^2)$
Eig $A_0 = Q_0\Lambda_0Q_0^{-1}$	$\mathcal{O}(n^3)$	Solve (16)	$\mathcal{O}(mk^2n^2)$
Assemble $\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}$	$\mathcal{O}(k^2n^2)$	Compute $W(\mathrm{v})$	$\mathcal{O}(n^3)$
TSVD $\mathcal{N}_1^{\mathrm{T}}\mathcal{M}_2$, $\mathcal{N}_2^{\mathrm{T}}\mathcal{M}_1$	$\mathcal{O}(n^3)$		
Compute $\widetilde{Z}_i$, Z_i, $i = 1, \ldots, k$	$\mathcal{O}(n^3)$		

Note: Depending on the properties of f, the cost of computing $W(\mathrm{v})$ reduces to $\mathcal{O}(n^2)$ flops.

Indeed, for stable A_0, by defining α_0 as the largest eigenvalue of the symmetric part of A_0, namely $\alpha_0 = \lambda_{\max}((A_0 + A_0^{\mathrm{T}})/2)$, if $\alpha_0 < 0$, standard results on the norm of solutions to Lyapunov equations say that

$$\|E(\mathrm{v})\|_F \leqslant \frac{1}{2|\alpha_0|}\|Q_0 E_1(\mathrm{v})B_l^{\mathrm{T}} + B_l E_2(\mathrm{v})^{\mathrm{T}} Q_0^{\mathrm{T}}\|_F$$

see, for example [26].

2.2 | Large-Scale Setting: A Projection Framework

In this section, we address the numerical solution of problems of large dimensions. In particular, due to a too large value of n, we assume that many of the operations described in the previous section, for example, the computation of the eigenvalue decomposition of A_0, are not affordable. On the other hand, we assume that it is possible, and efficient, to solve linear systems with A_0.

Assumption 2. Solving linear systems with A_0 and possibly multiple right-hand sides is possible and efficient.

In all our numerical results, we computed (and stored) the LU factors of A_0.

Taking inspiration from well-established projection methods for large-scale Lyapunov equations, we design a novel scheme for (5). As before, the main goal is to reuse the information we already have at hand, as much as possible, every time (5) has to be solved for a new v.

We employ the extended Krylov subspace method presented in [27] and further studied in [28]. In this section, we illustrate how to fully exploit its properties for our purposes.

In the case of a parameter independent Lyapunov equation of the form (4) with $Q = PP^{\mathrm{T}}$, the extended Krylov subspace

$$\begin{aligned}\mathbf{EK}_m^{\square}(A_0, P) &= \mathrm{Range}([P, A_0^{-1}P, A_0P, A_0^{-2}P, \ldots, A_0^{m-1}P, A_0^{-m}P]) \\ &= \mathbf{K}_m^{\square}(A_0, P) + \mathbf{K}_m^{\square}(A_0^{-1}, A_0^{-1}P) \end{aligned} \tag{19}$$

can be employed as approximation space. In (19), $\mathbf{K}_m^{\square}(A_0, P)$ denotes the standard, polynomial (block) Krylov subspace generated by A_0 and P, namely $\mathbf{K}_m^{\square}(A_0, P) = \mathrm{Range}([P, A_0P, \ldots, A_0^{m-1}P])$.

If $V_m \in \mathbb{R}^{n\times 2mk}$, $k = \mathrm{rank}(P)$, has orthonormal columns[3] and it is such that $\mathrm{Range}(V_m) = \mathbf{EK}_m^{\square}(A_0, P)$, the extended Krylov subspace method computes an approximate solution of the form $X_m = V_m Y_m V_m^{\mathrm{T}}$. The square matrix $Y_m \in \mathbb{R}^{2mk\times 2mk}$ is usually computed by imposing a Galerkin condition on the residual matrix $R_m = A_0X_m + X_mA_0^{\mathrm{T}} + PP^{\mathrm{T}}$, namely $V_m^{\mathrm{T}}R_mV_m = 0$. It is easy to show that such a condition is equivalent to computing Y_m by solving a *reduced* Lyapunov equation. With Y_m at hand, an accuracy measure, either the residual norm or the backward error, is computed to assess the quality of the current solution. Whenever this value is sufficiently small, we stop the process; otherwise the space is expanded.

A very similar scheme could also be employed for (5). However, both the right-hand side and the coefficient matrix in (5) depend on v so that, at a first glance, a new extended Krylov subspace has to be computed for each v. This would be too expensive from a computational point of view.

We show that only one extended Krylov subspace can be constructed and employed for solving all the necessary Lyapunov equations (5) regardless of the change in v. In particular, we suggest the use of $\mathbf{EK}_m^{\square}(A_0, P)$ as an approximation space, where P is defined as in (6). This selection is motivated by the following reasoning.

As reported in (6), the right-hand side of (5) can be written in factorized form as

$$P\big(\mathcal{I}_2 \otimes D(\mathrm{v})\big)P^{\mathrm{T}}, \qquad P = [X_0B_r,\ B_l] \in \mathbb{R}^{n\times 2k}, \qquad \mathcal{I}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Therefore, we can use only P as the starting block for the construction of the approximation space for every v. See, for example [31, Section 3.2].

The argument to justify the use of A_0 in place of $A(\mathrm{v})$ is a little more involved. We first observe that we can write

$$A(\mathrm{v})P = (A_0 - B_lD(\mathrm{v})B_r^{\mathrm{T}})P = A_0P - B_l(D(\mathrm{v})B_r^{\mathrm{T}}P)$$

and, since $\mathrm{Range}(B_l) \subseteq \mathrm{Range}(P)$ by definition, we have $\mathrm{Range}(A(\mathrm{v})P) \subseteq \mathrm{Range}(A_0P) + \mathrm{Range}(P)$. Similarly, we can show that

$$\mathrm{Range}(A(\mathrm{v})^jP) \subseteq \mathrm{Range}(A_0^jP) + \mathrm{Range}(P)$$

for all $j > 0$. This implies that $\mathbf{K}_m^{\square}(A(\mathrm{v}), P) \subseteq \mathbf{K}_m^{\square}(A_0, P)$. We now show that also the Krylov subspaces for the inverses are nested, that is, $\mathbf{K}_m^{\square}(A(\mathrm{v})^{-1}, A(\mathrm{v})^{-1}P) \subseteq \mathbf{K}_m^{\square}(A_0^{-1}, A_0^{-1}P)$, so that $\mathbf{EK}_m^{\square}(A(\mathrm{v}), P) \subseteq \mathbf{EK}_m^{\square}(A_0, P)$ and the use of the non-parametric space $\mathbf{EK}_m^{\square}(A_0, P)$ is thus justified.

Thanks to the SMW formula, we have

$$\begin{aligned}A(\mathrm{v})^{-1}P = (A_0 - B_lD(\mathrm{v})B_r^{\mathrm{T}})^{-1}P &= A_0^{-1}P \\ &+ A_0^{-1}B_l(D(\mathrm{v})^{-1} - B_r^{\mathrm{T}}A_0^{-1}B_l)^{-1}B_r^{\mathrm{T}}A_0^{-1}P\end{aligned}$$

and once again, since $\mathrm{Range}(A_0^{-1}B_l) \subseteq \mathrm{Range}(A_0^{-1}P)$, it holds $\mathrm{Range}(A(\mathrm{v})^{-1}P) \subseteq \mathrm{Range}(A_0^{-1}P)$. Similarly, $\mathrm{Range}(A(\mathrm{v})^{-j}P) \subseteq \mathrm{Range}(A_0^{-j}P)$ for all $j > 0$. Therefore, we can use $\mathbf{EK}_m^{\square}(A_0, P)$ for the computation of $X(\mathrm{v})$.

If $V_m \in \mathbb{R}^{n\times 4mk}$ has orthonormal columns, and it is such that $\mathrm{Range}(V_m) = \mathbf{EK}_m^{\square}(A_0, P)$, we seek an approximate solution $X_{\delta,m}(\mathrm{v}) = V_mY_m(\mathrm{v})V_m^{\mathrm{T}}$ to (5). As in the nonparametric case, the matrix $Y_m(\mathrm{v}) \in \mathbb{R}^{4mk\times 4mk}$ is computed by imposing a Galerkin condition on the residual, namely it is the solution of the projected equation

$$\begin{aligned}&(T_m - B_{l,m}D(\mathrm{v})B_{r,m}^{\mathrm{T}})Y(\mathrm{v}) + Y(\mathrm{v})(T_m - B_{l,m}D(\mathrm{v})B_{r,m}^{\mathrm{T}})^{\mathrm{T}} \\ &\quad = E_1G(\mathcal{I}_2 \otimes D(\mathrm{v}))G^{\mathrm{T}}E_1^{\mathrm{T}}\end{aligned} \tag{20}$$

where $T_m = V_m^{\mathrm{T}} A_0 V_m$ can be cheaply computed as suggested in [27], $B_{l,m} = V_m^{\mathrm{T}} B_l$ and $B_{r,m} = V_m^{\mathrm{T}} B_r$ can be updated on the fly by performing $4k$ inner products, whereas $G \in \mathbb{R}^{4k\times 2k}$ is such that $P = V_1 G$. Notice that B_l can be exactly represented in $\mathbf{EK}_m^{\square}(A_0, P)$, namely $B_l = V_m B_{l,m}$, as it is part of the initial block $P = [X_0 B_r, B_l]$. This does not hold for B_r.

Since the Galerkin method is structure preserving, the small-dimensional equation (20) can be solved by the procedure presented in Section 2.1. Notice that the well-posedness of Equation (20) is difficult to guarantee a priori without any further assumptions on B_l, B_r, and $D(\mathrm{v})$. Indeed, even in case of an A_0, and thus T_m, definite, the signature of $T_m - B_{l,m} D(\mathrm{v}) B_{r,m}^{\mathrm{T}}$ cannot be easily predicted. However, in Section 3.1, we will show that the structural properties, inherent to some of the application settings we are interested in, ensure to work with a matrix $A(\mathrm{v})$ that is stable for every v. Even though this does not guarantee Equation (20) to be well-defined for any m, it has been noticed in [32, Section 5.2.1] that it is very unlikely to get singular projected equations in this scenario.

Once $Y_m(\mathrm{v})$ has been computed, we need to decide whether the corresponding numerical solution $X_{\delta,m}(\mathrm{v}) = V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}}$ is sufficiently accurate. To this end, we employ the backward error, which can be computed as shown in the following proposition for the Lyapunov equation (5).

Proposition 1. *The backward error* $\Delta_{m,\mathrm{v}} = \frac{\|R_m(\mathrm{v})\|_F}{2\|A(\mathrm{v})\|_F \|X_{\delta,m}(\mathrm{v})\|_F + \|P(\mathcal{I}_2 \otimes D(\mathrm{v}))P^{\mathrm{T}}\|_F}$ *provided by the approximate solution* $X_{\delta,m}(\mathrm{v}) = V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}}$ *to (5) is such that*

$$\Delta_{m,\mathrm{v}} = \frac{\sqrt{2}\|E_m^{\mathrm{T}} \underline{T}_m Y_m(\mathrm{v})\|_F}{2\sqrt{\|A_0\|_F^2 + \mathrm{v}^{\mathrm{T}}((B_r^{\mathrm{T}} B_l)\circ(B_r^{\mathrm{T}} B_l))\mathrm{v} - 2a^{\mathrm{T}}\mathrm{v}} \cdot \|Y_m(\mathrm{v})\|_F + \|G(\mathcal{I}_2 \otimes D(\mathrm{v}))G^{\mathrm{T}}\|_F} \tag{21}$$

where $a = \mathrm{diag}(B_r^{\mathrm{T}} A_0 B_l)$.

Proof. The expression for the residual matrix

$$R_m = A(\mathrm{v}) X_m(\mathrm{v}) + X_m(\mathrm{v}) A(\mathrm{v})^{\mathrm{T}} - P(\mathcal{I}_2 \otimes D(\mathrm{v}))P^{\mathrm{T}}$$

directly comes from the Arnoldi relation

$$A_0 V_m = V_m T_m + \mathcal{V}_{m+1} E_m^{\mathrm{T}} \underline{T}_m$$

where $\mathcal{V}_{m+1} \in \mathbb{R}^{n\times 4k}$ is the $(m+1)$-th basis block, namely $V_{m+1} = [V_m, \mathcal{V}_{m+1}]$. We can write

$$\begin{aligned} R_m(\mathrm{v}) &= A(\mathrm{v}) X_{\delta,m}(\mathrm{v}) + X_{\delta,m}(\mathrm{v}) A(\mathrm{v})^{\mathrm{T}} - P(\mathcal{I}_2 \otimes D(\mathrm{v}))P^{\mathrm{T}} \\ &= A(\mathrm{v}) V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}} + V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}} A(\mathrm{v})^{\mathrm{T}} - P(\mathcal{I}_2 \otimes D(\mathrm{v}))P^{\mathrm{T}} \\ &= (A_0 - B_l D(\mathrm{v}) B_r^{\mathrm{T}}) V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}} \\ &\quad + V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}} (A_0 - B_l D(\mathrm{v}) B_r^{\mathrm{T}}) - P(\mathcal{I}_2 \otimes D(\mathrm{v}))P^{\mathrm{T}} \end{aligned}$$

Recalling that $B_l = V_m B_{l,m}$, $P = V_m E_1 G$, and plugging the Arnoldi relation above, we get

$$\begin{aligned} R_m(\mathrm{v}) &= V_m((T_m - B_{l,m} D(\mathrm{v}) B_{r,m}^{\mathrm{T}}) Y(\mathrm{v}) \\ &\quad + Y(\mathrm{v})(T_m - B_{l,m} D(\mathrm{v}) B_{r,m}^{\mathrm{T}})^{\mathrm{T}} \\ &\quad - E_1 G(\mathcal{I}_2 \otimes D(\mathrm{v}))G^{\mathrm{T}} E_1^{\mathrm{T}}) V_m^{\mathrm{T}} + \mathcal{V}_{m+1} E_m^{\mathrm{T}} \underline{T}_m Y_m(\mathrm{v}) V_m^{\mathrm{T}} \\ &\quad + V_m Y_m(\mathrm{v}) \underline{T}_m^{\mathrm{T}} E_m \mathcal{V}_{m+1}^{\mathrm{T}} \\ &= \mathcal{V}_{m+1} E_m^{\mathrm{T}} \underline{T}_m Y_m(\mathrm{v}) V_m^{\mathrm{T}} + V_m Y_m(\mathrm{v}) \underline{T}_m^{\mathrm{T}} E_m \mathcal{V}_{m+1}^{\mathrm{T}} \end{aligned}$$

Therefore, thanks to the orthogonality of V_m and $\mathcal{V}_{m+1}$, we have

$$\|R_m(\mathrm{v})\|_F^2 = 2\|E_m^{\mathrm{T}} \underline{T}_m Y_m(\mathrm{v})\|_F^2$$

Similarly, since $P = V_1 G$,

$$\|P(\mathcal{I}_2 \otimes D(\mathrm{v}))P^{\mathrm{T}}\|_F = \|G(\mathcal{I}_2 \otimes D(\mathrm{v}))G^{\mathrm{T}}\|_F$$

We now focus on $\|A(\mathrm{v})\|_F$. By exploiting the cyclic property of the trace operator, and recalling that $D(\mathrm{v})$ is diagonal, we have

$$\begin{aligned} \|A(\mathrm{v})\|_F^2 &= \|A_0 - B_l D(\mathrm{v}) B_r^{\mathrm{T}}\|_F^2 = \|A_0\|_F^2 + \|B_l D(\mathrm{v}) B_r^{\mathrm{T}}\|_F^2 \\ &\quad - 2\mathrm{trace}(B_l D(\mathrm{v}) B_r^{\mathrm{T}} A_0) \\ &= \|A_0\|_F^2 + \mathrm{trace}(B_l D(\mathrm{v}) B_r^{\mathrm{T}} B_l D(\mathrm{v}) B_r^{\mathrm{T}}) \\ &\quad - 2\mathrm{trace}(D(\mathrm{v}) B_r^{\mathrm{T}} A_0 B_l) \\ &= \|A_0\|_F^2 + \mathrm{trace}(D(\mathrm{v})(B_r^{\mathrm{T}} B_l) D(\mathrm{v})(B_r^{\mathrm{T}} B_l)) - 2a^{\mathrm{T}}\mathrm{v} \\ &= \|A_0\|_F^2 + \mathrm{v}^{\mathrm{T}}((B_r^{\mathrm{T}} B_l)\circ(B_r^{\mathrm{T}} B_l))\mathrm{v} - 2a^{\mathrm{T}}\mathrm{v} \end{aligned}$$

where $a = \mathrm{diag}(B_r^{\mathrm{T}} A_0 B_l)$. □

Proposition 1 shows that even though the computation of $\Delta_{m,\mathrm{v}}$ depends on the current parameter v, its evaluation can be carried out at low cost. In fact, quantities involving the full problem dimension n, like $\|A_0\|_F$, $(B_r^{\mathrm{T}} B_l)\circ(B_r^{\mathrm{T}} B_l)$, and a, can be computed in the offline stage. On the other hand, the computational effort for the remaining parameter-dependent terms is only quadratic in the dimension k of the parameter space. Thus, as long as k^2 stays well below n, this procedure is cheap. If the accuracy provided by $Y_m(\mathrm{v})$ is not adequate, we expand the space.

With the corresponding $Y_m(\mathrm{v})$ at hand, we can compute $f(X_{\delta,m}(\mathrm{v})) = f(V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}})$. Notice that, depending on the nature of f, the structure of $X_{\delta,m}(\mathrm{v})$ can be further exploited. For instance, we can cheaply evaluate $\mathrm{trace}(X_{\delta,m}(\mathrm{v}))$, as

$$\mathrm{trace}(X_{\delta,m}(\mathrm{v})) = \mathrm{trace}(V_m Y_m(\mathrm{v}) V_m^{\mathrm{T}}) = \mathrm{trace}(Y_m(\mathrm{v}))$$

thanks again to the cyclic property of the trace and the orthogonality of V_m. Thus, the basis V_m is not necessary to compute $\mathrm{trace}(X_{\delta,m}(\mathrm{v}))$.

The overall procedure is summarized in Algorithm 1. Note that Algorithm 1 can be easily modified to be used in optimization procedures, having instead of the set of parameter vectors V a starting vector v_0 and using Algorithm 1 to calculate $f(V_m Y_m(\mathrm{v}_\ell) V_m^{\mathrm{T}})$ for each new v_ℓ in the iterative optimization scheme while keeping, and possibly expanding, the computed subspace.

ALGORITHM 1 | Extended Krylov subspace method for calculating trace of a structured, parameter dependent Lyapunov equation.

input : $A_0 \in \mathbb{R}^{n\times n}$, V: the set of parameter vectors, $(B_r^\mathsf{T} B_l)\circledast(B_r^\mathsf{T} B_l) \in \mathbb{R}^{k\times k}$, $a = \mathrm{diag}(B_r^\mathsf{T} A_0 B_l) \in \mathbb{R}^k$, $P \in \mathbb{R}^{n\times 2k}$, $\|A_0\|_F$ max. iteration count $m_{\max}$, backward error bound $\epsilon > 0$.

output: $f(X_{\delta,m}(\mathrm{v}))$ for all $\mathrm{v} \in \mathsf{V}$.

1 Compute a skinny QR factorization of $[P, A_0^{-1}P] = V_1[\gamma, \theta]$
2 Set $m = 1$, $\mathsf{V}_0 = \mathsf{V}$
3 **while** $m<m_{\max}$ **do**
4 Compute next basis block $\mathcal{V}_{m+1}$, set $V_{m+1} = [V_m, \mathcal{V}_{m+1}]$, and update $T_m = V_m^\mathsf{T} A_0 V_m$ as in 48
5 Update $B_{l,m} = V_m^\mathsf{T} B_l$, $B_{r,m} = V_m^\mathsf{T} B_r$
6 Compute the offline quantities in Table 1 for solving Equation (20)
7 **for** *all* $v \in \mathsf{V}_0$ **do**
8 Perform the online steps in Table 1 and compute $Y_m(\mathrm{v})$
9 Compute $\Delta_{m,\mathrm{v}}$ as in (21)
10 **if** $\Delta_{m,v} \leqslant \epsilon$ **then**
11 Compute $f(V_m Y_m(\mathrm{v}) V_m^\mathsf{T})$
12 $\mathsf{V}_0 = \mathsf{V} \setminus \{\mathrm{v}\}$
13 **else**
14 **break** and go to line 15
15 Set $m = m + 1$

We recall that the solution of the linear systems with A_0 needed in the construction of the basis of $\mathbf{EK}_m^{\square}(A_0, P)$ can be efficiently carried out thanks to Assumption 2.

As outlined above, the construction of the extended Krylov method requires adding $4k$ vectors per iteration to the current basis even though the initial block P has rank $2k$. This could lead to the allocation of a large basis if many iterations m need to be performed. To avoid the use of an excessive amount of memory, one could combine our methodology with the compress-and-restart paradigm presented in [33] for the polynomial Krylov subspace method.

3 | Applications

In this section we illustrate two important problem settings where (1) needs to be solved several times, for many parameters v. The large number of equations in these scenarios makes our novel solvers very appealing, especially if compared to state-of-the-art procedures, which are not able to fully capitalize on the structure of the problem; see also Section 4.

3.1 | Damped Vibrational Systems

We consider linear vibrational systems described by

$$M\ddot{x} + C(\mathrm{v})\dot{x} + Kx = 0$$
$$x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0 \tag{22}$$

where M denotes the mass matrix, $C(\mathrm{v})$ denotes the parameter dependent damping matrix, K denotes the stiffness matrix, and x_0 and $\dot{x}_0$ are initial data. We assume that M and K are real, symmetric positive definite matrices of order m. The damping matrix is defined as $C(\mathrm{v}) = C_{\mathrm{int}} + C_{\mathrm{ext}}(\mathrm{v})$.

We assume that

$$C_{\mathrm{ext}}(\mathrm{v}) = BD(\mathrm{v})B^\mathsf{T} \tag{23}$$

where the matrix $B \in \mathbb{R}^{m\times k}$ describes the dampers' geometry and the matrix $D(\mathrm{v}) = \mathrm{diag}(v_1, v_2, \ldots, v_k)$ contains the damping viscosities $v_i > 0$, for $i = 1\ldots, k$. The viscosities will be encoded in the parameter vector $\mathrm{v} = \left[v_1, v_2, \ldots, v_k\right]^\mathsf{T} \in \mathbb{R}^k_{>0}$.

The internal damping C_{int} can be modeled in different ways. It is usually modeled as a Rayleigh damping matrix, which means that

$$C_{\mathrm{int}} = \alpha M + \beta K, \quad \text{with} \quad \alpha, \beta \geqslant 0,\ \alpha^2 + \beta^2 > 0 \tag{24}$$

or as a small multiple of the critical damping, that is,

$$C_{\mathrm{int}} = \alpha M^{1/2}\sqrt{M^{-1/2}KM^{-1/2}}M^{1/2}, \quad \text{where} \quad \alpha > 0 \tag{25}$$

See, for example [26, 34, 35] for further details.

Vibrations arise in a wide range of systems, such as mechanical, electrical, or civil engineering structures. Vibrations are a mostly unwanted phenomenon in vibrational structures, since they can lead to many undesired the effects such as the creation of noise, oscillatory loads, or waste of energy that may produce damaging effects on the considered structures. Therefore, in order to attenuate or minimize unwanted vibrations, an important problem is to determine the damping matrix in such a way that the vibrations of the system are as small as possible. This is usually achieved through optimization of the external damping matrix $C_{\mathrm{ext}}(\mathrm{v})$. Within this framework, we will focus on the optimization of the damping parameter vector v defining $C_{\mathrm{ext}}(\mathrm{v})$ as in (23). While damping optimization is a widely studied topic, there are still many challenging tasks that require efficient approaches. There is a vast literature in this field of research. For further details, we list only a few references that address the minimization of dangerous vibrations with different applications [26, 36–41].

The problem of vibration minimization requires a proper optimization criterion. A whole class of criteria are based on eigenvalues; see, for example [42–47]. Another important criterion is based on the total average energy of the system. This has been intensely considered in the last two decades; see, for example [26, 48–52]. Since our approach is also based on the total average energy, in the following we provide more details and set the stage for the application of our framework.

Several approaches trying to fully exploit the structure and accelerate the optimization process have been proposed in the literature about damping systems. In more details, in [53–55] the authors considered approaches that allowed derivation of explicit formulas for the total average energy, but they are adequate only for certain case studies. Furthermore, in [56, 57] the authors employed dimension reduction techniques in order to obtain efficient approaches for the calculation of the total average energy. However, these approaches require specific system

configurations, and they cannot be applied efficiently for general system matrices.

If we write (22) as a first-order ODE in phase space $\dot{y} = A(\mathrm{v})y$, the solution is given by $y(t) = \mathrm{e}^{A(\mathrm{v})t}y_0$, where y_0 denotes the vector of initial data. The average total energy of the system is given by

$$\int_{\|y_0\|=1}\int_0^\infty y(t)^{\mathrm{T}}y(t)\,\mathrm{d}t\,\mathrm{d}\sigma = \int_{\|y_0\|=1}\int_0^\infty y_0^{\mathrm{T}}\mathrm{e}^{A(\mathrm{v})^{\mathrm{T}}t}\mathrm{e}^{A(\mathrm{v})t}y_0\,\mathrm{d}t\,\mathrm{d}\sigma$$

where σ is a given non-negative measure on the unit sphere. From [26, Proposition 21.1] it follows that one can calculate the average total energy as trace($X(\mathrm{v})$), where $X(\mathrm{v})$ solves the Lyapunov equation

$$A(\mathrm{v})X(\mathrm{v}) + X(\mathrm{v})A(\mathrm{v})^{\mathrm{T}} = -Q \tag{26}$$

Here Q is the unique positive semidefinite matrix determined by σ. We will show that $A(\mathrm{v})$ has the form (2) and that Assumptions 1 is satisfied for both types of internal damping mentioned above and for typical choices of σ.

Using the linearization $y_1 = x$, $y_2 = \dot{x}$, the differential equation (22) in phase space can be written as

$$\dot{y} = A(\mathrm{v})y, \qquad \text{where} \qquad A(\mathrm{v}) = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C(\mathrm{v}) \end{bmatrix} \tag{27}$$

In the case of internal Rayleigh damping (24), it is easy to see that $A(\mathrm{v})$ has the form (2) with

$$A_0 = \begin{bmatrix} 0 & I \\ -M^{-1}K & -\alpha I - \beta M^{-1}K \end{bmatrix}, \quad B_l = \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix}, \quad B_r = \begin{bmatrix} 0 \\ B \end{bmatrix}$$

It can be shown that $A(\mathrm{v})$ is a Hurwitz matrix (the eigenvalues of A are in the open left half of the complex plane) if $\alpha^2 + \beta^2 > 0$. Moreover, by using the appropriate permutation matrix, A_0 can be written as a block diagonal matrix with 2×2 blocks on its diagonal so that its eigenvalues can be directly calculated. Assume $Z = \mathrm{diag}(K^{-1}, M^{-1})$, which corresponds to the choice of the surface measure σ generated by the energy norm $\|y\| = \frac{1}{\sqrt{2}}\sqrt{y_1^{\mathrm{T}}Ky_1 + y_2^{\mathrm{T}}My_2}$ on $\mathbb{R}^{2m}$. Then, one can calculate the corresponding X_0 as

$$X_0 = \begin{bmatrix} (\alpha K + \beta K M^{-1}K)^{-1} & -\frac{1}{2}K^{-1} \\ +\alpha K^{-1}M^{-1}K^{-1} + \beta K^{-1} & \\ -\frac{1}{2}K^{-1} & (\alpha K + \beta K M^{-1}K)^{-1} \end{bmatrix} \tag{28}$$

hence Assumption 1 is satisfied. It can be shown that the eigenvalues of A_0 are given by

$$\frac{1}{2}\left(-\alpha - \beta\lambda_i \pm \sqrt{(\alpha + \beta\lambda_i)^2 - 4\lambda_i}\right)$$

where $\lambda_i > 0$, $i = 1, \ldots, m$, are the eigenvalues of the matrix pair (K, M) and that the corresponding eigenvectors can be constructed from the eigenvectors of the pair (K, M). Hence, Assumption 2 is satisfied.

In the case of internal damping of the form (25), a different kind of linearization is more convenient. From the assumptions on M and K, it follows that there exists a matrix Φ that simultaneously diagonalizes M and K, that is,

$$\Phi^{\mathrm{T}}K\Phi = \Omega^2 \quad \text{and} \quad \Phi^{\mathrm{T}}M\Phi = I \tag{29}$$

where $\Omega = \mathrm{diag}(\omega_1, \ldots, \omega_m)$ contains the square roots of the eigenvalues of (K, M), which are eigenfrequencies of the corresponding undamped ($C(\mathrm{v}) = 0$) vibrational system. We assume that they are ordered in ascending order, $0 < \omega_1 \leqslant \omega_2 \leqslant \cdots \leqslant \omega_m$. It holds that the matrix Φ diagonalizes C_{int} as well, that is,

$$\Phi^{\mathrm{T}}C_{\mathrm{int}}\Phi = \alpha\Omega$$

for more details see, for example [26].

Using the linearization $y_1 = L_K^{\mathrm{T}}x$, $y_2 = L_M^{\mathrm{T}}\dot{x}$, where L_K, and L_M are the Cholesky factors of K and M, respectively, the matrix $A(\mathrm{v})$ can be written as

$$A(\mathrm{v}) = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\Phi^{\mathrm{T}}C(\mathrm{v})\Phi \end{bmatrix} \tag{30}$$

where Ω and Φ are given by (29). It is important to notice that the matrix $A(\mathrm{v})$ from (30) is Hurwitz if $\alpha > 0$; see, for example [26].

Since in this case the matrix A_0 is given by

$$A_0 = \begin{bmatrix} 0 & \Omega \\ -\Omega & -\alpha\Omega \end{bmatrix}$$

all its eigenvalues of A_0 are non-real for $\alpha < 2$.

Let $n = 2m$. Typical choices for the matrix Q in this case are $Q = \frac{1}{n}I$, which corresponds to the case when σ is generated by the Lebesgue measure on $\mathbb{R}^{n\times n}$, and, for $s < m$,

$$Q = \frac{1}{2s}\begin{bmatrix} I_s & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I_s & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{31}$$

which corresponds to the case when σ is generated by the Lebesgue measure on the subspace spanned by the vectors $[x_i, 0]^{\mathrm{T}}$ and $[0, x_i]^{\mathrm{T}}$, $i = 1, \ldots, s$, where x_i are the eigenvectors of the first s ω_I's, and on the rest of $\mathbb{R}^n$ it corresponds to the Dirac measure concentrated at zero.

For both choices of the matrix Q mentioned above, Assumption 1 is satisfied. Indeed, if $Q = \frac{1}{n}I$, we have

$$X_0 = \frac{1}{n}\begin{bmatrix} \frac{3\alpha}{2}\Omega^{-1} & -\frac{1}{2}\Omega^{-1} \\ -\frac{1}{2}\Omega^{-1} & \frac{1}{\alpha}\Omega^{-1} \end{bmatrix}$$

and if Q is given by (31), then a direct calculation shows that

$$X_0 = \frac{1}{2s}\begin{bmatrix} \frac{3\alpha}{2}\Omega_s^{-1} & 0 & -\frac{1}{2}\Omega_s^{-1} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}\Omega_s^{-1} & 0 & \frac{1}{\alpha}\Omega_s^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where $\Omega_s = \mathrm{diag}(\omega_1, \ldots, \omega_s)$.

3.2 | Multi-Agent Systems

The second application concerns the analysis of parameter variation in output synchronization for heterogeneous multi-agent systems. In a nutshell, multi-agent systems are a class of dynamical systems on networks, consisting of a number of dynamical systems (*agents*) connected in a network, where the agents can exchange information with the neighboring agents through a given interaction *protocol*, with the communication topology being specified through a (combinatorial) graph. Agent dynamics can be rather complex; see, for example [58]. Examples of systems that can be modeled as multi-agent systems are, for example, wireless sensor networks, power grids, and social networks. Common protocols are those for which the systems achieve consensus or output synchronization, and those for which the systems achieve desired formation or flocking state. The design and analyses of multi-agent systems have been a widely investigated field in recent years, and for more details, see, for example [59–61] and references therein.

The problem of output synchronization is how to design/control a system in such a way that the outputs of the agents converge to the same state. The general setting of output synchronization problems for heterogeneous multi-agent systems was considered, for example, in [4, 58, 62, 63].

An important aspect in heterogeneous multi-agent systems is studying the impact that one agent has, or a subset of agents has, on the whole system. This will, of course, depend on the dynamics of these agents and their location in the communication graph. The goal of this subsection is to analyze this impact with high computational efficiency in the case of the output synchronization protocol. We will analyze how one or more agents influence the entire system by using the H_2-norm (see, e.g., [64]) of the corresponding multi-agent system as a performance measure. By using this information, the system designer can modify or control the heterogeneous multi-agent system in order to achieve a target behavior.

To understand why equations of the form (1) arise in this setting, we first need to briefly introduce the bigger picture. Motivated by work from [65–67], we consider m agents with their dynamics described by

$$\begin{aligned}\dot{\xi}_i &= A^{(i)}\xi_i + B^{(i)}u_i + \omega_i\\ \zeta_i &= C^{(i)}\xi_i\end{aligned} \tag{32}$$

Here the function ξ_i represents the state of the agent i, the function u_i represents the input of the agent i, the function ζ_i represents the output of the agent i, and the function ω_i represents the exogenous disturbance of the agent i, $i \in \{1, 2, \ldots, m\}$. The matrices $A^{(i)}$, $B^{(i)}$, and $C^{(i)}$ are called the state, input, and output matrices, of the agent i, respectively. We assume that all matrices $A^{(i)}$ have the same size, and that the same holds for the matrices $B^{(i)}$ and $C^{(i)}$. Let G be the corresponding communication graph, which models the connections of the agents from (32), with nodes $\{1, \ldots, m\}$. This means that (i, j) is an edge in the graph G if the agents i and j exchange information through the common protocol. A protocol we are going to study is the following:

$$u_i(t) = -K^{(i)} \sum_{j \in \mathfrak{N}_i} (\zeta_i(t) - \zeta_j(t)), \quad i = 1, \ldots, m \tag{33}$$

Here $\mathfrak{N}_i$ denotes the set of neighboring agents of the agent i (described by G), and $K^{(i)}$ is a so-called gain matrix of appropriate dimension [67].

We define the stack vector $\xi = (\xi_1, \ldots, \xi_m)$. Let n be the size of the matrices $A^{(i)}$, hence, nm is the size of the vector ξ. As we want to treat scenarios in which the disturbances ω_i need not all be independent, let us suppose that among the exogenous disturbances $\{\omega_1, \ldots, \omega_m\}$ there are $k \in \{1, \ldots, m\}$ different ones $\{\omega_{i_1}, \ldots, \omega_{i_k}\}$ (e.g., the same ocean waves or wind gusts concurrently disturb several agents). Then $\begin{bmatrix}\omega_1 & \cdots & \omega_m\end{bmatrix}^\top = H\begin{bmatrix}\omega_{i_1} & \cdots & \omega_{i_k}\end{bmatrix}^\top$, where the matrix $H \in \mathbb{R}^{m\times k}$ is given by

$$H_{jl} = \begin{cases} 1, & \omega_j = \omega_{i_l} \\ 0, & \text{otherwise} \end{cases}, \quad j = 1, \ldots, m, \quad l = 1, \ldots, k$$

With $\mathbf{E} = H \otimes I_n$, we denote the corresponding disturbance matrix. Let $\omega = (\omega_{i_1}, \ldots, \omega_{i_k})$. With $\mathbf{C}$ we denote the overall output matrix $\mathbf{C} = \operatorname{diag}(C^{(1)}, \ldots, C^{(m)})$.

The closed-loop dynamic equation of the system described by (32) and (33) can be represented as

$$\begin{aligned}\dot{\xi}(t) &= \mathbf{A}\xi(t) + \mathbf{E}\omega(t)\\ \zeta(t) &= \mathbf{C}\xi(t)\end{aligned} \tag{34}$$

where the system matrix $\mathbf{A} \in \mathbb{R}^{nm\times nm}$ in (34) can be seen as a block matrix whose (i, j)-th block $\mathbf{A}_{ij}$ is given by

$$\mathbf{A}_{ij} = \delta_{i,j}A^{(i)} - \ell_{ij}B^{(i)}K^{(i)}C^{(j)}, \quad i, j \in \{1, \ldots, m\}$$

where ℓ_{ij}, $i, j = 1, 2, \ldots, m$, are the entries of the Laplacian matrix L of the graph G, see, for example [66]. Note, however, that the model in [66] has delays, while our model has no delays.

By calculating the H_2-norm of the system (34), we are calculating the norm of the mapping $\omega \mapsto \zeta$, thus measuring how the disturbance ω is influencing the multi-agent system. This boils down to the calculation of trace($\mathbf{E}\mathbf{E}^\mathrm{T}X$), where X solves the Lyapunov equation $\mathbf{A}^\mathrm{T}X + X\mathbf{A} = -\mathbf{C}^\mathrm{T}\mathbf{C}$. If we were interested in measuring the influence of the disturbance to a part of the system, that is, to a subset of agents, instead of the matrix $\mathbf{C}$ in (34), we would use the matrix $\mathbf{PC}$, where $\mathbf{P}$ has the form $\mathbf{P} = P \otimes I$, with P being an orthogonal projector to the subspace spanned by the canonical vectors corresponding to the subset of agents we are interested in.

We want to study the impact that a variation of dynamics in one or a subset of agents has on the H_2-norm of the resulting system. To this end, we need to compute trace($\mathbf{E}\mathbf{E}^\mathrm{T}X(\mathrm{v})$), where now $X(\mathrm{v})$ denotes the solution to a Lyapunov equation of the form (1) with $Q = \mathbf{C}^\mathrm{T}\mathbf{C}$. In this framework, the low-rank modification $B_l D(\mathrm{v})B_r^\mathrm{T}$ in the coefficient matrix (2) is meant to take into account the variation in the dynamics of the agents of interest. In particular, B_l and B_r will encode the location of the agents

to be modified, whereas $D(\mathrm{v})$ amounts to the parameter variation to be applied to those agents.

In the numerical examples in Section 4 we choose symmetric matrices $A^{(i)}$ and $K^{(i)}$, and we will also take $B^{(i)} = (C^{(i)})^{\mathrm{T}}$, $i = 1, \ldots, m$. Hence, the matrix $\mathbf{A}$ will be symmetric, and therefore Assumptions 1 and 2 will be satisfied.

4 | Numerical Examples

We now illustrate the advantages of the proposed methods through different numerical experiments. First, we construct an artificial example aimed at illustrating some of the features of our routines. We then examine problem settings coming from the two applications described in the previous section. Firstly, we consider an example, including analyzing the influence of damping parameters in a mechanical system. Secondly, we study the impact that variation of agents' dynamics has on a couple of different multi-agent systems. To this end, we employ our novel schemes to efficiently compute the H_2-norm of the underlying linear time-invariant system (34).

Example 1. In the first example, we consider $A_0 \in \mathbb{R}^{n\times n}$, $n = 2\ 500$, coming from the centered finite difference discretization of the 2D Laplacian operator on the unit square with zero Dirichlet boundary conditions. The matrices $B_l,\ B_r \in \mathbb{R}^{n\times k}$, $k = 3$, and $Q = cc^{\mathrm{T}}$, $c \in \mathbb{R}^n$, have normally distributed random entries. We construct the diagonal matrix

$$D(\mathrm{v}) = \begin{bmatrix} v_1 & & \\ & v_2 & \\ & & v_3 \end{bmatrix}$$

as follows. We consider p different values for v_1, v_2, and v_3 so that we solve equation (1) for all the possible p^3 different instances of $\mathrm{v} = [v_1, v_2, v_3]$. In particular, v_1 takes p logarithmically equally spaced values in $[1, 10]$. Similarly, v_2 and v_3 take p logarithmically equally spaced values in $[10^2, 10^3]$ and $[10^4, 10^5]$, respectively. Notice that this selection of the parameters implies that $D(\mathrm{v})$ is never a scalar multiple of the identity.

We start by solving the sequence of Lyapunov equations described above by the SMW+recycling Krylov technique presented in Section 2.1. We select $p = 2$ so that we solve 8 Lyapunov equations. The parameters for the GCRO-DR method are as follows: The threshold on the relative residual norm is 10^{-10}, the maximum number of iterations allowed is 300, and the number of eigenvectors s_ℓ used in the recycling technique is $s_\ell = 10$ for all ℓ.

We first report on Figure 1 (left) the peculiar sparsity pattern of the matrix $\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}$ that we analyzed in detail in Section 2.1. We can easily appreciate the 2×2 block structure of this matrix and the remarkable structured sparsity pattern of its diagonal blocks. The analysis of this sparsity pattern is crucial in different ways. First, it allows us to assemble $\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}$ at low cost. Second, it suggests the design of a natural preconditioner for GCRO-DR, namely the operator $\mathcal{P}$ in (17). We now study whether this preconditioner is indeed effective in reducing the GCRO-DR iteration count. To this end, in Figure 1 (center), we report the number of iterations achieved by GCRO-DR with no preconditioning (red circles) and GCRO-DR preconditioned with (17) (blue stars). In the preconditioner (17), the rank $\bar{p}$ of $\mathcal{N}_1\mathcal{M}_1^{\mathrm{T}}$ and $\mathcal{N}_2\mathcal{M}_2^{\mathrm{T}}$ coming from the TSVD of $\mathbf{N}_1^{\mathrm{T}}\mathbf{L}_0^{-1}\mathbf{M}_1$ and $\mathbf{N}_2^{\mathrm{T}}\mathbf{L}_0^{-1}\mathbf{M}_2$, respectively, is 5.

We can observe how our preconditioner effectively cuts down the number of iterations for every parameter configuration we tested. In particular, for some selection of v, we achieve up to a 50% reduction in the iteration count.

We now consider the novel projection scheme we presented in Section 2.2. We want to compare it with applying the extended Krylov subspace method [27] to every single instance of equation (1). Since, for this problem, we consider a low-rank right-hand side $Q = cc^T$, the application of EKSM to (1) is possible and requires the construction of the subspaces $\mathbf{EK}_m^{\square}(A(\mathrm{v}), c)$. For the linear system solves with $A(\mathrm{v}) = A_0 - B_l D(\mathrm{v}) B_r^T$ needed in the extended Krylov basis construction, we apply the SMW formula, where we precompute the LU factorization of A_0 only once. We set $p = 10$ so that we need to solve a sequence of $1,000$ equations.

The threshold we adopt for the backward error is 10^{-8} in both approaches. The GCRO-DR setting needed in the solution of the projected problems in our scheme is as before.

In Figure 1 (right), we report the dimension of the constructed $\mathbf{EK}_m^{\square}(A(\mathrm{v}), c)$ for any v. These values need to be compared to the dimension of $\mathbf{EK}_m^{\square}(A_0, P)$ our method constructs. For

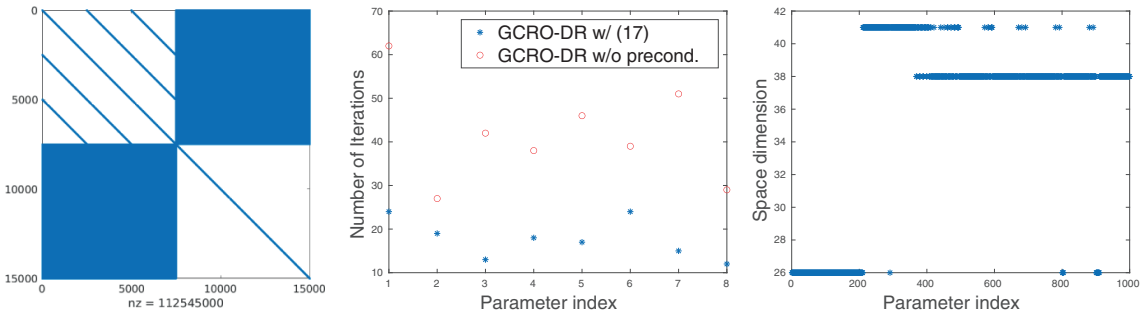

FIGURE 1 | Example 1. Left: Sparsity pattern of $\mathbf{N}^{\mathrm{T}}\mathbf{L}^{-1}\mathbf{M}$. Center: Numbers of iterations performed by GCRO-DR when preconditioned with (17) (blue stars) and when not preconditioned at all (red circles) for all the different parameter selections we tested. Right: Dimension of the space $\mathbf{EK}_m^{\square}(A(\mathrm{v}), c)$ constructed for the solution of each instance of (1).

this example, we get $\dim(\mathbf{EK}_m^{\square}(A_0, P)) = 72$. Even though the latter is slightly larger than the values reported in Figure 1 (right), we remind the reader that we construct $\mathbf{EK}_m^{\square}(A_0, P)$ only once, whereas $\mathbf{EK}_m^{\square}(A(\mathrm{v}), c)$ is built from scratch every time v changes. This aspect can lead to large computational timings, especially when the orthogonalization step in the construction of $\mathbf{EK}_m^{\square}(A(\mathrm{v}), c)$ gets expensive, namely when large spaces are needed to converge. As reported in Figure 1 (right), this does not happen for this example, and $\dim(\mathbf{EK}_m^{\square}(A(\mathrm{v}), c))$ remains rather moderate for all v. This explains why, for this example, our projection technique results in being only slightly faster than a naive application of EKSM: the latter takes 115.76 s, whereas our scheme requires 93.27 s to converge. However, in the next example, coming from damping optimization, we will see that very large subspaces are indeed necessary to compute accurate solutions.

Example 2. We consider a mechanical system of $2d + 1$ masses consisting of two main rows of d masses connected with springs; see, for example [57, 68]. The springs in the first row of masses have stiffness k_1, and those in the second row have stiffness k_2. The first masses, on the left edge, (i.e., masses m_1 and m_{d+1}) are connected to a fixed boundary, while, on the other side of the rows, the last masses (m_d and m_{2d}) are connected to the mass m_{2d+1} which, via a spring with stiffness k_3, is connected to a fixed boundary. An illustration is given in Figure 2.

The model is given by (22) where the mass matrix is $M = \mathrm{diag}(m_1, m_2, \ldots, m_{2d+1})$ and the stiffness matrix K is given by

$$K = \begin{bmatrix} K_{11} & & -\kappa_1 \\ & K_{22} & -\kappa_2 \\ -\kappa_1^{\mathrm{T}} & -\kappa_2^{\mathrm{T}} & k_1 + k_2 + k_3 \end{bmatrix}, \quad K_{ii} = k_i \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}$$

with $\kappa_i = \begin{bmatrix} 0 & \ldots & 0 & k_i \end{bmatrix}$ for $i = 1, 2$.

Each row has d masses, and we consider systems with $2d + 1$ masses, including

$$m_i = \begin{cases} \frac{1}{10}(2d + 1 - 2i), & i = 1, \ldots, 500 \\ \frac{1}{10}(i - 500) + 100, & i = 501, \ldots, 1000 \\ 160, & i = 1000, \ldots, 2d \end{cases}$$
$$m_{2d+1} = 175 \tag{35}$$

The stiffness values are chosen as $k_1 = 40, k_2 = 20$, and $k_3 = 30$. We assume that the internal damping is modeled as a small multiple of the critical damping (25) with $\alpha = 0.04$. We would like to remind the reader that, for a given d, the coefficient matrix $A(\mathrm{v})$ in (1) has dimension $n = 4d + 2$.

We consider viscosity optimization over three dampers ($k = 3$) with viscosities v_1, v_2, and v_3 with their positions encoded in

$$B_l = B_r = \left[e_{i_1}, e_{i_1 + \frac{d}{10}} - e_{i_1 + \frac{d}{10} + d}, e_{i_2} \right], \quad 1 \leqslant i_1 \leqslant d, d + 1 \leqslant i_2 \leqslant 2d \tag{36}$$

where e_i is the ith canonical vector and the indices i_1 and i_2 determine the damping positions. The first damping positions will damp the first row of masses (using a grounded damper). Similarly, the third damps the second row of masses, while the second damper connects both rows of masses of the considered mechanical system.

We will optimize the viscosity parameters with respect to the average total energy measure introduced in Section 3.1. The optimization problems were solved by using MATLAB's built-in `fminsearch` with the starting point $\mathrm{v}^0 = (100, 100, 100)$. The stopping tolerance for this routine was set to 10^{-4}.

The performance of our new approaches will be illustrated on two damping configurations with different features.

a. In the first case, we consider a small dimensional problem with $d = 400$, so we have a system with 801 masses. Here we damp the 9 lowest undamped eigenfrequencies, that is, $s = 9$ in (31). The damping geometry is determined by (36), and 20 different damping positions will be considered. These are determined by the following indices:

$$i_1 \in \{50, 130, 210, 290\}, \text{and } i_2 \in \{460, 540, 620, 700, 780\}$$

Due to the small problem dimension ($n = 1,602$), we employ the SMW+recycling Krylov technique presented in Section 2.1 to solve this problem. In particular, we use the following parameters for the GCRO-DR method. The threshold on the relative residual norm is 10^{-10}, the maximum number of iterations allowed is 300, and the number of eigenvectors s_ℓ used in the recycling technique is $s_\ell = 10$ for all ℓ. In the preconditioner (17), the rank $\overline{p}$ of $\mathcal{N}_1\mathcal{M}_1^{\mathrm{T}}$ and

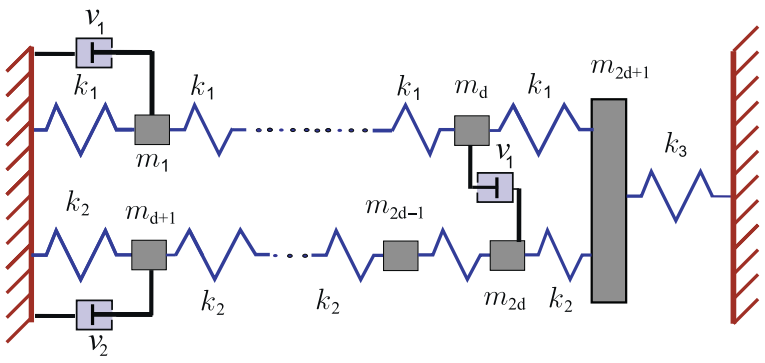

FIGURE 2 | Example 2, illustration of the mechanical system.

$\mathcal{N}_2\mathcal{M}_2^{\mathrm{T}}$ coming from the TSVD of $\mathbf{N}_1^{\mathrm{T}}\mathbf{L}_0^{-1}\mathbf{M}_1$ and $\mathbf{N}_2^{\mathrm{T}}\mathbf{L}_0^{-1}\mathbf{M}_2$, respectively, is 50.

b. In the second case, we increase the problem dimension by considering $d = 1,000$; thus, here we have a system with 2,001 masses and $n = 4002$. In this case, we damp the 21 lowest undamped eigenfrequencies, that is, $s = 21$ in (31). Here we consider 25 damping positions determined by the following indices:

$$i_1 \in \{50, 250, 450, 650, 850\}, \text{and } i_2 \in \{1150, 1350, 1550, 1750, 1950\}$$

We apply the projection framework illustrated in Section 2.2 for the solution of this larger problem. We would like to mention that a Lyapunov equation whose coefficient matrix is of dimension 4,002, as in this case, is usually not considered to be large scale in the matrix equation community. On the other hand, the huge number of Lyapunov equations we need to solve within the optimization procedure makes our novel projection framework very appealing. In Algorithm 1 we employ $\epsilon = 10^{-8}$ and $m_{\max} = 120$. The projected Equations (20) are solved by means of our SMW+recycling Krylov approach, adopting the same parameters as in case (a) above.

We first focus on case (a) described above and in Figure 3 we report the relative errors obtained by our SMW+recycling Krylov method. The relative errors in the optimal viscosities were calculated by $\|\mathrm{v}^* - \mathrm{v}\|/\|\mathrm{v}\|$, where v and v^* denote the optimal viscosity vectors calculated by the SMW+recycling Krylov method and the MATLAB's function `lyap`, respectively. Similarly, the relative errors in the average total energy are calculated by $|\mathrm{trace}(X(\mathrm{v})^*) - \mathrm{trace}(X(\mathrm{v}))|/\mathrm{trace}(X(\mathrm{v})^*)$, where $\mathrm{trace}(X(\mathrm{v})^*)$ is once again the optimal trace for the given configuration obtained by the MATLAB's function `lyap`, and $\mathrm{trace}(X(\mathrm{v}))$ is the optimal trace calculated by our SMW+recycling Krylov scheme.

From the results reported in Figure 3 we can notice that our novel SMW+recycling Krylov approach leads to a solution process able to achieve very small errors in the average total energy. Also, the computed optimal viscosities turn out to be rather accurate, attaining an error of the same order as the threshold used for `fminsearch`. Moreover, our novel SMW+recycling Krylov strategy is very efficient. In particular, in this example, it accelerated the overall optimization process approximately 2.7 times (in the average case). Figure 4 shows a precise acceleration ratio for each considered configuration. The displayed time ratio is the ratio between the total time needed for the direct calculation of the average total energy by using MATLAB's function `lyap` and the total time needed for the viscosity optimization by using the SMW+recycling Krylov method.

We would like to mention that, also for this example, we tried to use also GCRO-DR with no preconditioning within our recycling Krylov technique. In the best case scenarios, plain GCRO-DR managed to converge by performing a larger number of iterations: $\mathcal{O}(10^2)$ iterations to be compared to the $\mathcal{O}(10^1)$ iterations performed in the preconditioned case. Such a large number of iterations remarkably worsened the overall performance in terms of computational time. In some other cases, unpreconditioned GCRO-DR did not achieve the prescribed level of accuracy, thus jeopardizing the convergence of the outer minimization procedure. This shows once again that the preconditioning operator $\mathcal{P}$ described in (17) works well, even though more performing preconditioners can certainly be designed.

We now turn our attention to case (b). As already mentioned, having coefficient matrices of dimension 4,002 is often considered as working in the small-scale setting in the matrix equation literature so, that dense linear algebra solvers may be preferred in this case. In this example, we would like to show that employing a projection framework is largely beneficial also in this scenario. To this end, we compare our fresh projection framework with the SMW+recycling Krylov technique we propose in Section 2.1. Indeed, we showed above that the latter method is able to achieve small errors while accelerating the overall solution process. Figure 5 shows the relative errors achieved by our projection framework with respect to the SMW+recycling

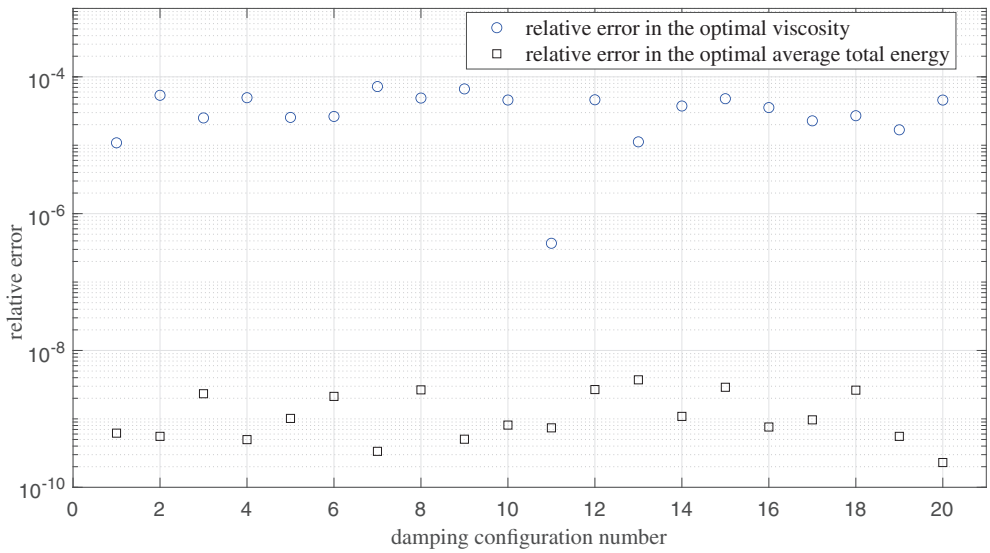

FIGURE 3 | Example 2, the case (a). Relative errors in the total average energy (squares) and in the viscosity (circles) at optimal gains for the SMW+recycling Krylov method.

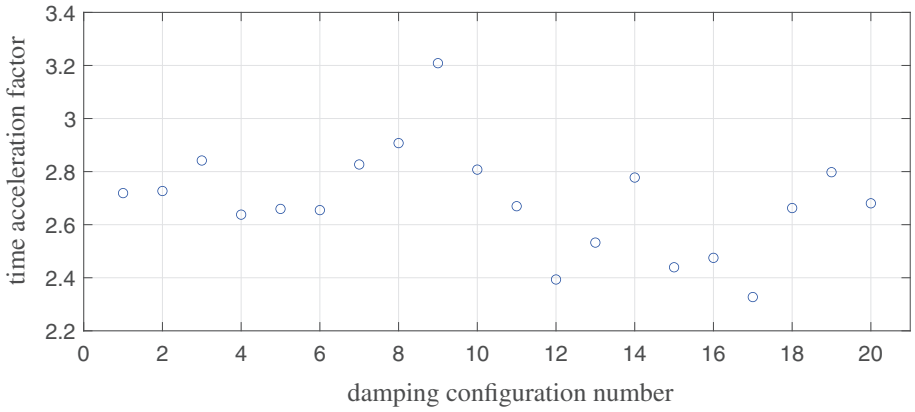

FIGURE 4 | Example 2, case (a). Acceleration factors in the overall minimization procedure attained by employing our novel recycling Krylov approach.

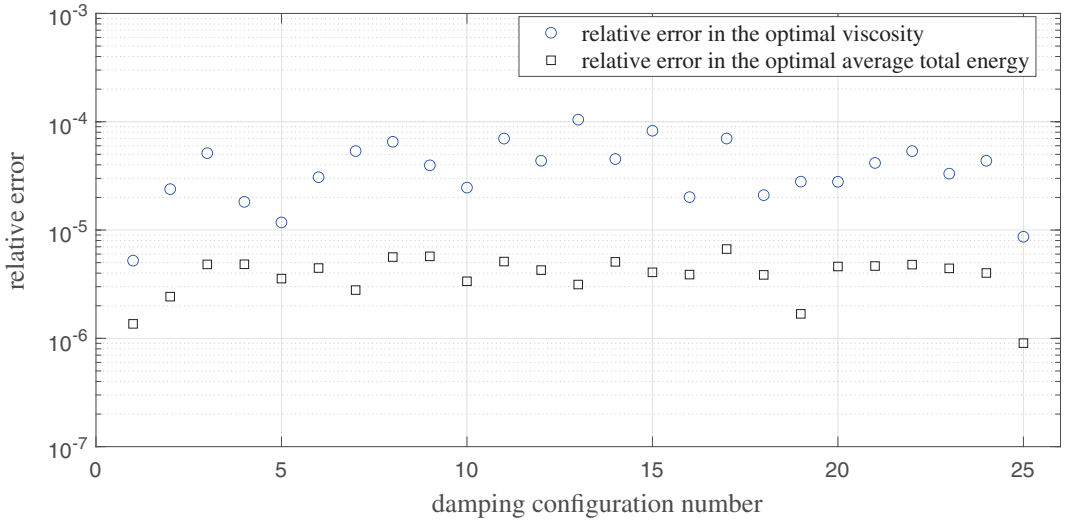

FIGURE 5 | Example 2, case (b). Relative errors in the total average energy (squares) and in the viscosity (circles) at optimal gains for the projection framework.

Krylov method. We can notice that the error in the optimal viscosities is still of the same order of magnitude of the threshold used within `fminsearch`, whereas the relative errors in the optimal average total energy norm are a little higher than the ones reported in Figure 3, even though still satisfactory. Moreover, the projection framework method accelerated the optimization process by approximately 23.3 times, and Figure 6 showcases a precise acceleration time ratio for all the 25 configurations, compared to the SMW+recycling Krylov method. The main reason why our novel projection method performs so well in terms of computational time lies in the fact that we basically use the same approximation subspace for every v_ℓ. In particular, once we construct a sufficiently large $\mathbf{EK}_m^{\square}(A_0, P)$ for the first equation, namely Δ_{m,v_0} meets our accuracy demand, we keep using the same approximation subspace for all the subsequent equations, by expanding it very few times instead of computing a new subspace from scratch. For most of the damping configurations we considered, $\mathbf{EK}_m^{\square}(A_0, P)$ does not need to be expanded at all after its construction during the solution of the first equation. Only for three configurations – $(i_1, i_2) \in \{(250, 1 \quad 750), (450, 1 \quad 550), (450, 1 \quad 950)\}$ – $\mathbf{EK}_m^{\square}(A_0, P)$ needed to be expanded a couple of times during the optimization procedure. In particular, for $(i_1, i_2) = (250, 1 \quad 750)$, the dimension of $\mathbf{EK}_m^{\square}(A_0, P)$ after the solution of the first equation was 708. This space has been expanded twice during the online optimization step, getting a space of dimension 720 and then 732. Similarly, for $(i_1, i_2) = (450, 1 \quad 950)$ we started the online phase with a space of dimension 780, which got expanded twice, getting a space of dimension 792 first, and 804 later. For $(i_1, i_2) = (450, 1 \quad 550)$, $\mathbf{EK}_m^{\square}(A_0, P)$ was expanded only once, passing from a space of dimension 768 to a space of dimension 780. In general, for this example, the dimension of $\mathbf{EK}_m^{\square}(A_0, P)$ averaged over all the configurations is about 749, with a minimum and maximum dimension equal to 708 and 852, respectively.

In the next numerical examples, we are going to illustrate the following two cases of analysis of multi-agent systems that are of interest:

- For each agent i, we choose $H = e_i$. Then we study the influence of the variation of the dynamics of the agent i by calculating the H_2-norm of the corresponding systems. By

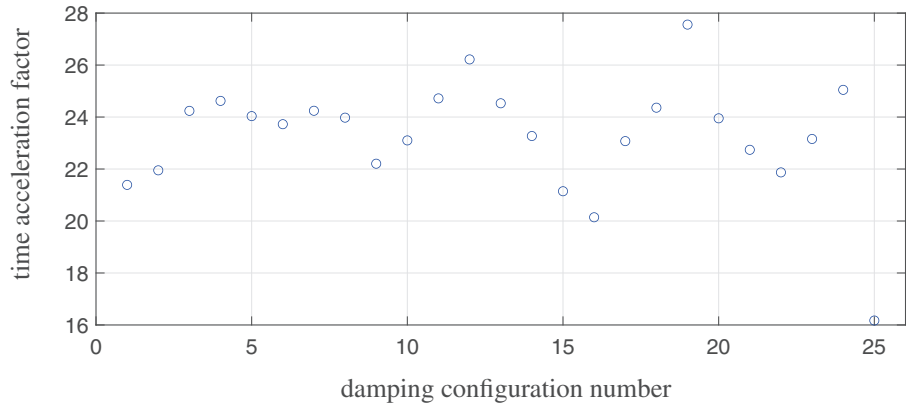

FIGURE 6 | Example 2, case (b). Acceleration factors in the overall minimization procedure attained by employing our new projection framework.

applying this procedure to all agents, we can study the structural importance of each agent in the system.

- We choose $H = I_m$. For a given set of subsets of agents $\{J_1, \ldots, J_q\}$, $J_i \subset \{1, \ldots, m\}$, and a given set of parameters of agents in J_i, we study the influence of these parameters on the dynamics of the overall system. By calculating the corresponding H_2-norms, we can study the influence of these parameters on the dynamics of the whole system when all agents are exposed to independent external disturbances.

See Examples 3 and 4 for further details.

Example 3. For the sake of simplicity, here we investigate agent dynamics that are defined by 2×2 state matrices. We consider $m = 196$ agents. The state matrices of all agents are given by

$$A^{(i)} = \begin{bmatrix} -10 & 5 \\ 5 & -8 \end{bmatrix}, \quad \text{for} \quad i = 1, \ldots, m$$

By following the notation of Section 3.2, the other components of the system are

$$K^{(i)} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad B^{(i)} = (C^{(i)})^{\mathrm{T}} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad \text{for} \quad i = 1, \ldots, m$$

Therefore, the 2×2 blocks of the system matrix $\mathbf{A} \in \mathbb{R}^{2m \times 2m}$ are given by

$$\mathbf{A}_{ij} = \delta_{i,j} A^{(i)} - \ell_{ij} B^{(i)} K^{(i)} C^{(j)}, \quad i, j \in \{1, \ldots, m\}$$

An illustration of the underlying topology can be seen in Figure 7.

We would like to analyze the influence of parameter variations in different agents. This means that for a fixed k-th agent to be altered, we consider a low-rank update of $\mathbf{A}$ aimed at modifying only its k-th diagonal 2×2 block. In particular, given the parameters $\mathrm{v} = (v_1, v_2, v_3)$, the matrix $\mathbf{A}^{(k)}(\mathrm{v})$ corresponds to a low-rank update of $\mathbf{A}$ of the following form

$$\mathbf{A}^{(k)}(\mathrm{v}) = \mathbf{A} - B_l \mathrm{diag}(v_1, v_2, v_2, v_3) B_r^{\mathrm{T}} \tag{37}$$

where $B_l, B_r \in \mathbb{R}^{2m \times 4}$ are determined by the agent index k. It holds,

$$B_l(2(k-1)+1 : 2k, 1 : 4) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

$$B_r(2(k-1)+1 : 2k, 1 : 4) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

whereas all other entries of B_l and B_r are zero. In this example, we considered the influence of the variation of the dynamics of one agent by calculating the H_2-norm of a corresponding system when only this agent is externally disturbed. Thus, in the case of the matrix $\mathbf{A}^{(k)}$ (that corresponds to analyzing the k-th agent), the corresponding matrix $\mathbf{E}$ is given by $\mathbf{E} = e_k \otimes I_n$, for $k = 1, \ldots, m$.

For all different agents, we will analyze the dependence of the system to the parameter variation v_1, v_2, and v_3 of the k-th agent, $k = 1, \ldots, m$. In particular, the low-rank update of the system matrix given by (37) influences the k-th block in the following way. The diagonal elements are altered by the parameters v_1 and v_3, while the off-diagonal elements are symmetrically modified by the parameter v_2. All the parameters v_1, v_2, and v_3 will be varied between -9.9 and 10.1 with step 1.

In this example we will modify all the agents, namely we consider all agent's indices $k = 1, \ldots, m$.

We note that some instances of $\mathbf{A}^{(k)}(\mathrm{v})$ turned out to be non-stable. In particular, we have computed the rightmost eigenvalue λ of $\mathbf{A}^{(k)}(\mathrm{v})$ and discarded those configurations for which $\mathrm{Re}(\lambda) \geqslant 0$. By doing so, for this particular example, the total number of considered parameters is equal to 1,400,328[4]. This means that we needed to calculate the H_2-norm, and thus solve a Lyapunov equation, a significant number of times. In particular, for fixed v_1, v_2, and v_3, we are interested in identifying for which k, $k = 1, \ldots, m$, the H_2-norm of the underlying system over described parameter variations is maximal. This provides an insight on the importance of the considered agent.

Table 2 presents the average computational time needed by different methods to perform the task described above. We report the average required time by our novel solvers, namely the SMW+recycling Krylov method and the projection framework (denoted in the table by SMW+rec. Krylov and proj. framework, respectively). The computational parameters of our SMW+recycling Krylov method are as in Example 2. The

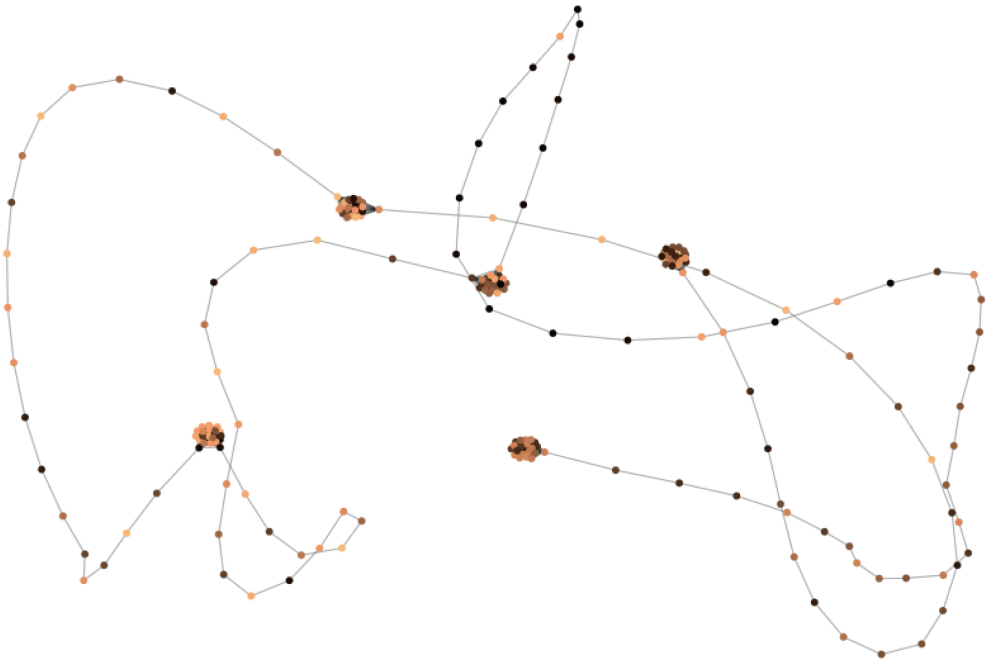

FIGURE 7 | Example 3. Maximal H_2-norm for different agents with respect to the parameter variations we performed. Darker colors correspond to the agents where the maximal H_2-norm is larger.

TABLE 2 | Example 3. Computational time and relative errors.

	Average required time (s)	Average relative error
`lyap`	625.02	—
SMW+rec. Krylov	678.05	3.575×10^{-12}
proj. framework	17.48	3.675×10^{-12}

only exceptions are in the GCRO-DR threshold and the rank $\overline{p}$ of the TSVD $\mathcal{N}_1\mathcal{M}_1^{\mathrm{T}}$ and $\mathcal{N}_2\mathcal{M}_2^{\mathrm{T}}$ used in the preconditioner. Here, we use 10^{-8} and $\overline{p} = 5$, respectively. For Algorithm 1 we used $\epsilon = 10^{-10}$ and $m_{\max} = 200$. The projected equations are solved by our SMW+recycling Krylov method with the same setting we have just described above.

In the same table, we also document the relative errors achieved by the different algorithms. To this end, we considered the results obtained by using `lyap` as *exact*. As we can see, all the approaches achieve a very small relative error. On the other hand, the projection framework outperforms all other approaches in terms of computational time; being one order of magnitude faster than all the other algorithms we tested. Also in the multi-agent system framework, reusing the same subspace $\mathbf{EK}_m^{\square}(A_0, P)$ is key for our projection method to be successful. For this example, the extended Krylov subspace generated by our routine is very small for all the configurations we tested. In particular, the space dimensions range from 8 to 56, with an average value equal to 29.08.

Figure 7 illustrates the topology of the problem and provides information regarding the maximal H_2-norm we computed by varying the parameter of the system. The node colors illustrate this latter aspect. In particular, darker colors correspond to the agents where the maximal H_2-norm turns out to be larger. It can be seen that certain groups of agents result in much larger H_2-norms. Therefore, they are much more important for the considered multi-agent system, as altering those agents may lead to an (almost) unstable system. Thanks to our novel solvers, this kind of analysis can now be accurately carried out at an acceptable cost.

Example 4. In this example, we still consider agent dynamics defined by 2×2 state matrices of dimension m. We aim at analyzing the parameter variation in off-diagonal elements of consecutive agents. The system matrices $A^{(i)}$, $B^{(i)}$, $C^{(i)}$, $K^{(i)}$, and $\mathbf{A}$ are defined as in Example 3 but here we consider $m = 200$.

The Laplacian matrix L of the underlying graph is given by Algorithm 2 and the corresponding matrices B_l and B_r determining the low-rank perturbation are defined in the following way. For a given odd index $1 \leqslant k \leqslant 2m - 3$, $B_l, B_r \in \mathbb{R}^{2m \times 4}$ are

$$B_l(k : k+3, 1 : 4) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_r(k : k+3, 1 : 4) = I_4$$

and all other entries of B_l and B_r are zero.

The modified system matrix will depend on two parameters, v_1 and v_2, and it is defined as

$$\mathbf{A}^{(k)}(\mathrm{v}) = \mathbf{A} - B_l \mathrm{diag}(v_1, v_1, v_2, v_2) B_r^{\mathrm{T}}$$

This means that for a fixed index k, the off-diagonal elements of the diagonal blocks determined by k and $k+1$ are modified by

ALGORITHM 2 | Construction of the Laplacian matrix L in Example 4.

input : number of agents m
output: the Laplacian matrix L

$r = [m/20 + m/50 : m/10,\ m/4 + m/20 : m/4 + m/10,\ m/2 + m/20 + 2 : m/2 + m/20 + m/50, \ldots \quad m/2 + m/10 + 2 : m/2 + m/10 + m/20,\ m - m/10 : m - m/20] \in \mathbb{R}^{i_r}$
$h = [1 : m/20,\ m/10 + m/40 : m/10 + m/20,\ m/2 : m/2 + m/20,\ m - 1] \in \mathbb{R}^{i_h}$
for $i = 1 : m - 2$ **do**
  $L(i, i+1) = -1;\ L(i, m) = -1;$
$L(1, m-1) = -1;\ L(m-1, m) = -1;$
for $i = 1 : i_r$ **do**
  if $r(i) == (m-1)$ **then**
    $L(1, m-1) = 0;$
  else
    $L(r(i), r(i)+1) = 0;$
$L(h, m) = 0;\ L = L + L^{\mathsf{T}};$
for $i = 1 : m$ **do**
  $L(i, i) = -\sum_{j=1}^{m}(L(i, j));$

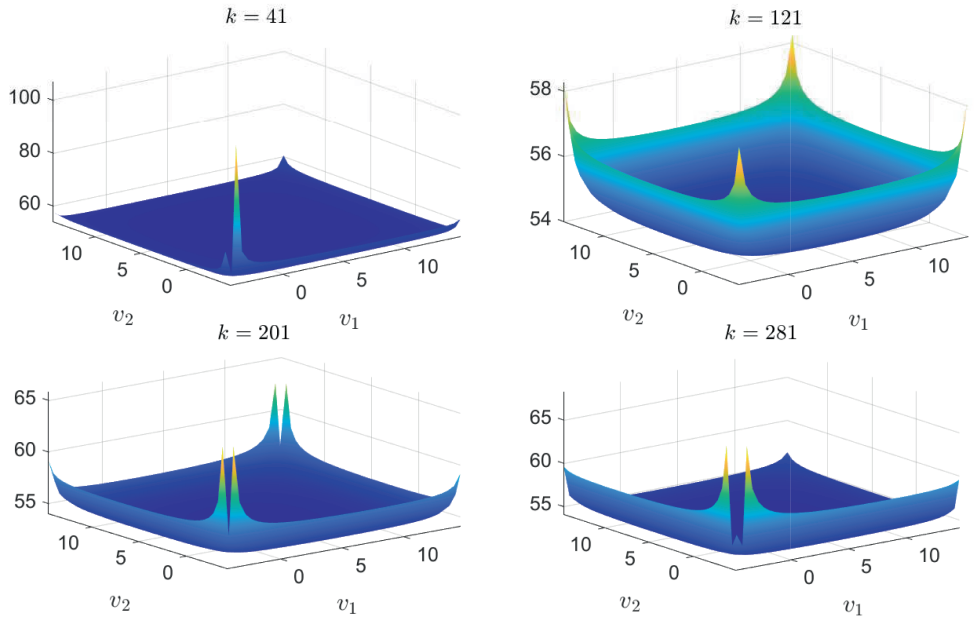

FIGURE 8 | Example 4: magnitude of the H_2-norm.

TABLE 3 | Example 4. Computational time and relative errors.

	Total required time (s)				Average relative error			
k	41	121	201	281	41	121	201	281
`lyap`	165.51	141.36	157.43	164.64	—	—	—	—
SMW+rec. Krylov	257.49	242.93	255.52	265.59	7×10^{-14}	1.03×10^{-13}	2.07×10^{-13}	1.7×10^{-13}
proj. framework	15.23	9.55	10.71	25.31	9.79×10^{-14}	7.64×10^{-14}	2.18×10^{-13}	1.28×10^{-11}

v_1 and v_2, respectively. The parameters v_1 and v_2 vary between -4.9 and 14.6 with step 0.5. As before, we only consider the stable instances of the matrices $\mathbf{A}^{(k)}$ for our purposes.

We will study the behavior of the H_2-norm of the underlying system by varying the parameter $k \in \{61, 141, 221, 301\}$. The total number of parameter configurations $\mathrm{v} = (v_1, v_2)$ for which $\mathbf{A}^k(\mathrm{v})$

is stable is equal to 6 044[5]. Moreover, in this example, we considered the matrix $\mathbf{E} = I_{2m}$ for all cases, meaning that we measured the H_2-norm of the corresponding systems when all agents were independently externally disturbed.

We test the same routines as in Example 3, namely `lyap`, and our novel schemes, with the same setting as before.

We start by reporting the relative errors and the computational timings attained in the computation of the H_2-norms for all the considered parameters; see Table 3. The relative errors are calculated with respect to H_2-norm obtained by using `lyap`.

From the results in Table 3 we can see that our SMW+recycling Krylov scheme always attains very small relative errors. On the other hand, it turns out to be not very competitive in terms of computational timings. A finer tuning of the GCRO-DR parameters and the preconditioning operator may lead to some improvements in the performance of our scheme. Our projection framework outperforms all other approaches in terms of computational timing.

The extended Krylov subspaces constructed by our projection method turn out to be a little larger in this example than the ones in Example 3. In particular, the smallest and largest subspaces have dimensions 12 and 96, respectively, whereas the average space dimension is 49.71.

In Figure 8 we display the magnitude of the H_2-norm by varying v for a fixed k. It can be seen how certain parameter values have a much greater impact on the H_2-norm of the system, thus emphasizing the important role of the corresponding pairs of agents. Once again, thanks to our novel solution processes, such analysis can be carried out in very few seconds on a standard laptop.

5 | Conclusions

We proposed two different, efficient, and accurate methods for solving sequences of parametrized Lyapunov equations. The SMW+recycling Krylov approach is well-suited for small dimensional problems and is able to provide solutions achieving a very small relative errors. The proposed projection framework relies on the extended Krylov subspace method, and it makes use of the aforementioned SMW+recycling Krylov technique to solve the projected equations. Even though this second algorithm is tailored to large-scale problems, our numerical results show that it is extremely competitive also for medium-sized problems, especially if the number of Lyapunov equations to be solved is very large, as it happens in the application settings we studied. We showed that the projection framework is able to speed up the entire solution process by an order of magnitude. Expensive analyses like viscosity optimization for vibrational systems and output synchronization of multi-agent systems are, thus, now affordable.

Acknowledgments

The first author is a member of the INdAM Research Group GNCS. His work was partially supported by the funded project GNCS2023, "Metodi avanzati per la risoluzione di PDEs su griglie strutturate, e non" (CUP_E53C22001930001).

The work of the second author was supported in parts by the Croatian Science Foundation under the project 'Vibration Reduction in Mechanical Systems' (IP-2019-04-6774). Open access publishing facilitated by Universita degli Studi di Bologna, as part of the Wiley - CRUI-CARE agreement.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Endnotes

[1] For the sake of simplicity, we assume the data of our problem to be real. Our algorithms can handle complex data by applying straightforward modifications.

[2] This is the default strategy in [24].

[3] Hereafter, we assume V_m to have full rank. If this is not the case, standard deflation strategies can be adopted; see, for example [29, 30, Section 7.1].

[4] This number is obtained by multiplying all the possible values of v_1, v_2, and v_3, namely 21^3, by all the adopted selections of k, that is, m, and then subtracting the number of unstable cases (414, 828).

[5] As before, this number is obtained by multiplying all the possible values of v_1 and v_2, namely 40^2, by all the adopted selections of k, that is, 4, and then subtracting the number of unstable cases (356).

References

1. N. Truhar, "An Efficient Algorithm for Damper Optimization for Linear Vibrating Systems Using Lyapunov Equation," *Journal of Computational and Applied Mathematics* 172 (2004): 169–182, https://doi.org/10.1016/j.cam.2004.02.005.

2. N. T. Son, P.-Y. Gousenbourger, E. Massart, and T. Stykel, "Balanced Truncation for Parametric Linear Systems Using Interpolation of Gramians: A Comparison of Algebraic and Geometric Approaches," in *Model Reduction of Complex Dynamical Systems*, vol. 171 (Berlin, Germany: International Series of Numerical Mathematics, Birkhäuser/Springer, Cham, 2021), 31–51, https://doi.org/10.1007/978-3-030-72983-7_2.

3. B. Zhou, G. Duan, and Z. Lin, "A Parametric Lyapunov Equation Approach to the Design of Low Gain Feedback," *IEEE Transactions on Automatic Control* 53 (2008): 1548–1554, https://doi.org/10.1109/TAC.2008.921036.

4. H. Kim, H. Shim, and J. H. Seo, "Output Consensus of Heterogeneous Uncertain Linear Multi-Agent Systems," *IEEE Transactions on Automatic Control* 56 (2011): 200–206, https://doi.org/10.1109/TAC.2010.2088710.

5. H. Su, M. Chen, J. Lam, and Z. Lin, "Semi-Global Leader-Following Consensus of Linear Multi-Agent Systems With Input Saturation via Low Gain Feedback," *IEEE Transactions on Circuits and Systems I: Regular Papers* 60 (2013): 1881–1889, https://doi.org/10.1109/TCSI.2012.2226490.

6. R. H. Bartels and G. W. Stewart, "Algorithm 432: Solution of the Matrix Equation AX + XB = C," *Communications of the ACM* 15 (1972): 820–826, https://doi.org/10.1145/361573.361582.

7. I. Kuzmanović and N. Truhar, "Optimization of the Solution of the Parameter-Dependent Sylvester Equation and Applications," *Journal of Computational and Applied Mathematics* 237 (2013): 136–144, https://doi.org/10.1016/j.cam.2012.07.022.

8. D. Kressner, M. Plešinger, and C. Tobler, "A Preconditioned Low-Rank CG Method for Parameter-Dependent Lyapunov Matrix Equations,"

Numerical Linear Algebra with Applications 21 (2014): 666–684, https://doi.org/10.1002/nla.1919.

9. D. Kressner, S. Massei, and L. Robol, "Low-Rank Updates and a Divide-And-Conquer Method for Linear Matrix Equations," *SIAM Journal on Scientific Computing* 41 (2019): A848–A876, https://doi.org/10.1137/17M1161038.

10. M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, "Low-Rank Optimization on the Cone of Positive Semidefinite Matrices," *SIAM Journal on Optimization* 20 (2010): 2327–2351, https://doi.org/10.1137/080731359.

11. J. Przybilla, I. Pontes Duff, and P. Benner, "Semi-Active Damping Optimization of Vibrational Systems Using the Reduced Basis Method," *Advances in Computational Mathematics* 50, no. 3 (2023): 1–28, https://doi.org/10.48550/arXiv.2305.12946.

12. N. T. Son and T. Stykel, "Solving Parameter-Dependent Lyapunov Equations Using the Reduced Basis Method With Application to Parametric Model Order Reduction," *SIAM Journal on Matrix Analysis and Applications* 38 (2017): 478–504, https://doi.org/10.1137/15M1027097.

13. M. Köhler, *Approximate Solution of Non-Symmetric Generalized Eigenvalue Problems and Linear Matrix Equation on HPC Platforms* (Magdeburg, Germany: Dissertation, Otto-von-Guericke-Universität, 2021).

14. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. (Baltimore: Johns Hopkins University Press, 2013).

15. Y. Hao and V. Simoncini, "The Sherman–Morrison–Woodbury Formula for Generalized Linear Matrix Equations and Applications," *Numerical Linear Algebra with Applications* 28 (2021): e2384, https://doi.org/10.1002/nla.2384.

16. I. Kuzmanović and N. Truhar, "Sherman–Morrison–Woodbury Formula for Sylvester and T-Sylvester Equations With Applications," *International Journal of Computer Mathematics* 90 (2013): 306–324, https://doi.org/10.1080/00207160.2012.716154.

17. T. Damm, "Direct Methods and ADI-Preconditioned Krylov Subspace Methods for Generalized Lyapunov Equations," *Numerical Linear Algebra with Applications* 15 (2008): 853–871, https://doi.org/10.1002/nla.603.

18. S. Massei, D. Palitta, and L. Robol, "Solving Rank-Structured Sylvester and Lyapunov Equations," *SIAM Journal on Matrix Analysis and Applications* 39 (2018): 1564–1590, https://doi.org/10.1137/17M1157155.

19. P. Benner, J.-R. Li, and T. Penzl, "Numerical Solution of Large-Scale Lyapunov Equations, Riccati Equations, and Linear-Quadratic Optimal Control Problems," *Numerical Linear Algebra with Applications* 15 (2008): 755–777, https://doi.org/10.1002/nla.622.

20. T. Penzl, "Lyapack Users Guide, Tech. Report SFB393/00-33, Sonderforschungsbereich 393 Numerische Simulation auf massiv parallelen Rechnern," TU Chemnitz, 09107 Chemnitz, Germany (2000), http://www.tu-chemnitz.de/sfb393/sfb00pr.html.

21. A. Gaul, *Recycling Krylov Subspace Methods for Sequences of Linear Systems – Analysis and Applications* (Dissertation: Technischen Universität Berlin, 2014), https://doi.org/10.13140/2.1.4015.3284.

22. M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, "Recycling Krylov Subspaces for Sequences of Linear Systems," *SIAM Journal on Scientific Computing* 28 (2006): 1651–1674, https://doi.org/10.1137/040607277.

23. K. M. Soodhalter, E. Sturler, and M. E. Kilmer, "A Survey of Subspace Recycling Iterative Methods," *GAMM-Mitteilungen* 43 (2020): e202000016, https://doi.org/10.1002/gamm.202000016.

24. M. L. Parks, K. M. Soodhalter, and D. B. Szyld, "Block GCRO-DR: A version of the recycled GMRES method using block Krylov subspaces and harmonic Ritz vectors," April (2016), https://doi.org/10.5281/zenodo.48836.

25. M. L. Parks, K. M. Soodhalter, and D. B. Szyld, "A block recycled GMRES method with investigations into aspects of solver performance," math.NA. (2016), http://arxiv.org/abs/1604.01713.

26. K. Veselić, *Damped Oscillations of Linear Systems: A mathematical introduction*, vol. 2023 (Berlin, Heidelberg: Springer, 2011), https://doi.org/10.1007/978-3-642-21335-9.

27. V. Simoncini, "A New Iterative Method for Solving Large-Scale Lyapunov Matrix Equations," *SIAM Journal on Scientific Computing* 29 (2007): 1268–1288, https://doi.org/10.1137/06066120X.

28. L. Knizhnerman and V. Simoncini, "Convergence Analysis of the Extended Krylov Subspace Method for the Lyapunov Equation," *Numerische Mathematik* 118 (2011): 567–586, https://doi.org/10.1007/s00211-011-0366-3.

29. S. Birk, *Deflated Shifted Block Krylov Subspace Methods for Hermitian Positive Definite Matrices* (Dissertation, Fakultät für Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, 2015).

30. M. H. Gutknecht, *Block Krylov Space Methods for Linear Systems With Multiple Right-Hand Sides: An Introduction*, eds. A. H. Siddiqi, I. S. Duff, and O. Christensen (New Delhi: Anamaya, 2007), 420–447, https://people.math.ethz.ch/~mhg/pub/delhipap.pdf.

31. N. Lang, H. Mena, and J. Saak, "On the Benefits of the LDL^T Factorization for Large-Scale Differential Matrix Equation Solvers," *Linear Algebra and Its Applications* 480 (2015): 44–71, https://doi.org/10.1016/j.laa.2015.04.006.

32. V. Simoncini, "Computational Methods for Linear Matrix Equations," *SIAM Review* 58 (2016): 377–441, https://doi.org/10.1137/130912839.

33. D. Kressner, K. Lund, S. Massei, and D. Palitta, "Compress-and-Restart Block Krylov Subspace Methods for Sylvester Matrix Equations," *Numerical Linear Algebra with Applications* 28 (2021): e2339, https://doi.org/10.1002/nla.2339.

34. D. E. Beskos and B. A. Boley, "Critical Damping in Linear Discrete Dynamic Systems," *ASME Journal of Applied Mechanics* 47 (1980): 627–630, https://doi.org/10.1115/1.3153744.

35. I. Kuzmanović, Z. Tomljanović, and N. Truhar, "Optimization of Material With Modal Damping," *Applied Mathematics and Computation* 218 (2012): 7326–7338, https://doi.org/10.1016/j.amc.2012.01.011.

36. C. Beards, *Structural Vibration: Analysis and Damping* (New York, Toronto: Elsevier, 1996).

37. C. Du and L. Xie, *Modeling and Control of Vibration in Mechanical Systems* (Boca Raton: CRC press, 2016).

38. G. Genta, *Vibration Dynamics and Control* (New York, NY: Springer, 2009), https://doi.org/10.1007/978-0-387-79580-5.

39. P. Müller and W. Schiehlen, *Linear Vibrations* (Dordrecht: Martinus Nijhoff Publishers, 1985).

40. I. Takewaki, *Building Control With Passive Dampers: Optimal Performance-Based Design for Earthquakes* (United States: John Wiley and Sons Ltd, 2009).

41. L. Zuo and S. A. Nayfeh, "Optimization of the Individual Stiffness and Damping Parameters in Multiple-Tuned-Mass-Damper Systems," *Journal of Vibration and Acoustics* 127 (2005): 77–83, https://doi.org/10.1115/1.1855929.

42. S. J. Cox, "Designing for Optimal Energy Absorption I: Lumped Parameter Systems," *ASME Journal of Vibration and Acoustics* 120 (1998): 339–345.

43. P. Freitas and P. Lancaster, "On the Optimal Value of the Spectral Abscissa for a System of Linear Oscillators," *SIAM Journal on Matrix Analysis and Applications* 21 (1999): 195–208, https://doi.org/10.1137/S0895479897331850.

44. N. Gräbner, V. Mehrmann, S. Quraishi, C. Schröder, and U. von Wagner, "Numerical Methods for Parametric Model Reduction in the

Simulation of Disk Brake Squeal," *Zeitschrift für Angewandte Mathematik und Mechanik* 96 (2016): 1388–1405, https://doi.org/10.1002/zamm.201500217.

45. J. Moro and J. Egana, "Directional Algorithams for Frequency Isolation Problem in Undamped Vibrational Systems," *Mechanical Systems and Signal Processing* 75 (2016): 11–26.

46. F. Tisseur and K. Meerbergen, "The Quadratic Eigenvalue Problem," *SIAM Review* 43 (2001): 235–286, https://doi.org/10.1137/S0036144500381988.

47. J.-H. Wehner, D. Jekel, R. Sampaio, and P. Hagedorn, *Damping Optimization in Simplified and Realistic Disc Brakes* (Cham: SpringerBriefs in Applied Sciences and Technology, Springer, 2018), https://doi.org/10.1007/978-3-319-62713-7.

48. K. Brabender, *Optimale Dämpfung von linearen Schwingungssystemen*, PhD thesis (Hagen: Fernuniversität, 1998).

49. S. Cox, I. Nakić, A. Rittmann, and K. Veselić, "Lyapunov Optimization of a Damped System," *Systems & Control Letters* 53 (2004): 187–194, https://doi.org/10.1016/j.sysconle.2004.04.004.

50. I. Nakić, "Minimization of the Trace of the Solution of Lyapunov Equation Connected With Damped Vibrational Systems," *Mathematical Communications* 18 (2013): 219–229.

51. N. Truhar, Z. Tomljanović, and M. Puvača, "An Efficient Approximation for Optimal Damping in Mechanical Systems," *International Journal of Numerical Analysis and Modeling* 14 (2017): 201–217.

52. N. Truhar and K. Veselić, "An Efficient Method for Estimating the Optimal dampers' Viscosity for Linear Vibrating Systems Using Lyapunov Equation," *SIAM Journal on Matrix Analysis and Applications* 31 (2009): 18–39, https://doi.org/10.1137/070683052.

53. Z. Tomljanović, N. Truhar, and K. Veselić, "Optimizing a Damped System - A Case Study," *International Journal of Computer Mathematics* 88 (2011): 1533–1545, https://doi.org/10.1080/00207160.2010.521547.

54. K. Veselić, "On Linear Vibrational Systems With One Dimensional Damping," *Applicable Analysis* 29 (1988): 1–18, https://doi.org/10.1080/00036818808839770.

55. K. Veselić, "On Linear Vibrational Systems With One Dimensional Damping II," *Integral Equations and Operator Theory* 13 (1990): 883–897, https://doi.org/10.1007/BF01198923.

56. P. Benner, Z. Tomljanović, and N. Truhar, "Dimension Reduction for Damping Optimization in Linear Vibrating Systems," *Zeitschrift für Angewandte Mathematik und Mechanik* 91 (2011): 179–191, https://doi.org/10.1002/zamm.201000077.

57. P. Benner, Z. Tomljanović, and N. Truhar, "Optimal Damping of Selected Eigenfrequencies Using Dimension Reduction," *Numerical Linear Algebra with Applications* 20 (2013): 1–17, https://doi.org/10.1002/nla.833.

58. D. Nojavanzadeh, Z. Liu, A. Saberi, and A. Stoorvogel, "Scale-Free Protocol Design for H_∞ Almost Output and Regulated Output Synchronization of Heterogeneous Multi-Agent Systems," in *2021 40th Chinese Control Conference (CCC)* (New York, NY: IEEE, 2021), 5056–5061, https://doi.org/10.23919/CCC52363.2021.9550620.

59. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks, Princeton Ser. Appl. Math* (Princeton, NJ: Princeton University Press, 2010), https://doi.org/10.1515/9781400835355.

60. C. Wang, Z. Zuo, J. Wang, and Z. Ding, *Robust Cooperative Control of Multi-Agent Systems: A Prediction and Observation Prospective* (Boca Raton: CRC Press, 2021).

61. W. Yu, G. Wen, G. Chen, and J. Cao, *Distributed Cooperative Control of Multi-Agent Systems* (Singapore: John Wiley & Sons, 2017).

62. M. R. Davoodi, K. Khorasani, H. A. Talebi, and H. R. Momeni, "Distributed Fault Detection and Isolation Filter Design for a Network of Heterogeneous Multiagent Systems," *IEEE Transactions on Control Systems Technology* 22 (2014): 1061–1069, https://doi.org/10.1109/TCST.2013.2264507.

63. J. Jiao, H. L. Trentelman, and M. K. Camlibel, "H2 Suboptimal Output Synchronization of Heterogeneous Multi-Agent Systems," *Systems & Control Letters* 149 (2021): 1–8, https://doi.org/10.1016/j.sysconle.2021.104872.

64. K. Zhou and J. C. Doyle, *Essentials of Robust Control*, vol. 104 (NJ: Prentice hall Upper Saddle River, 1998).

65. I. Nakić, D. Tolić, Z. Tomljanović, and I. Palunko, "Numerically Efficient H_∞ Analysis of Cooperative Multi-Agent Systems," *Journal of The Franklin Institute* 359 (2022): 9110–9128, https://doi.org/10.1016/j.jfranklin.2022.09.013.

66. I. Palunko, D. Tolić, and V. Prkačin, "Learning Near-Optimal Broadcasting Intervals in Decentralized Multi-Agent Systems Using Online Least-Square Policy Iteration," *IET Control Theory & Applications* 15 (2021): 1054–1067, https://doi.org/10.1049/cth2.12102.

67. W. Ren and R. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control - Theory and Applications* (London: Springer, 2008) In: Communications and control engineering.

68. C. Beattie, S. Gugercin, and Z. Tomljanović, "Sampling-Free Model Reduction of Systems With Low-Rank Parameterization," *Advances in Computational Mathematics* 46 (2020): 1–34, https://doi.org/10.1007/s10444-020-09825-8.