Structural and Multidisciplinary Optimization (2023) 66:85
https://doi.org/10.1007/s00158-023-03528-6

RESEARCH PAPER

Static and dynamic topology optimization: an innovative unifying approach

Paolo Venini[1] · Marco Pingaro[2]

Received: 24 October 2022 / Revised: 14 January 2023 / Accepted: 14 February 2023 / Published online: 27 March 2023
© The Author(s) 2023

Abstract

This paper presents a topology optimization approach that is innovative with respect to two distinct matters. First of all the proposed formulation is capable to handle static and dynamic topology optimization with virtually no modifications. Secondly, the approach is inherently a multi-input multi-output one, i.e., multiple objectives can be pursued in the presence of multiple loads. The input-to-output transfer matrix, say $\boldsymbol{G}$, is the key ingredient that governs the algebraic mapping between applied loads and structural response. In statics $\boldsymbol{G}$ depends on the design variables only, whereas it depends on the frequency variable as well in the dynamic case. The Singular Value Decomposition (SVD) of $\boldsymbol{G}$ represents then the core of the proposed approach. Singular values are shown to be the gains of the input/output mapping and are used to compute proper norms of $\boldsymbol{G}$ that represent the goal functions to be minimized. Singular vectors provide at no extra cost the plant directions, i.e., the load combination factors that stress the structure the most. Numerical examples are discussed in much detail and open issues object of ongoing investigations are highlighted. A full Matlab code handling the static topology optimization problem is provided as an online Appendix to the manuscript. Its extension to the dynamic case may be gathered following the formulation proposed in Sect. 5.

Keywords Topology Optimization · Statics · Dynamics · Singular Value Decomposition (SVD) · Matlab

1 Introduction and motivation

Topology optimization is nowadays a mature discipline from both engineering and computational viewpoints. A systematic view about principles, computational methods, and applications was given in the classic book (Bendsøe and Sigmund 2003) that dates back some twenty years ago. The vast majority of contributions in both static and dynamic regimes aim at the minimization of (possibly slightly different definitions of) the structural compliance in the presence of further goals and/or constraints.

With no aim of completeness, as far as static response is concerned stress constraints were investigated in Bruggi and Duysinx (2012), Bruggi and Venini (2008), Giraldo-Londono et al. (2022), and Collet et al. (2017) focused on fatigue response, design of compliant mechanisms was tackled in Emmendoerfter et al. (2022), whereas Ferrari and Sigmund (2020b) and Gersborg-Hansen et al. (2006) addressed heat conduction and buckling, respectively. The development of fast computational methods was pursued in Andreassen et al. (2011) and Ferrari and Sigmund (2020a), whereas the advantages of mesh adaptivity were exploited in Salazar de Troya and Tortorelli (2018), just to mention a few. The above list is by no means complete and reference is made to the references therein and to the review paper (Sigmund and Maute 2013) for a comprehensive list of contributions and applications.

When it comes to dynamic response, a first main subdivision should be made between time-domain and frequency-domain approaches. As for time-domain approaches, Min et al. (1999) should be cited as a pioneering contribution,

Responsible Editor: Jianbin Du

Paolo Venini and Marco Pingaro have contributed equally to this work.

✉ Paolo Venini
paolo.venini@unipv.it

[1] Department of Civil Engineering and Architecture, University of Pavia, Via Ferrata, 3, Pavia 27100, Italy

[2] Department of Structural and Geotechnical Engineering, University of Rome La Sapienza, Via Gramsci 53, Rome 00197, Italy

Giraldo-Londono and Paulino (2021) represents a milestone in that a general approach is presented along with a Matlab implementation that fits a more general modular environment, whereas (Jensen et al. 2014) faces and solves the fundamental issue of sensitivity computation. Switching to frequency-domain methods, Olhoff and Du (2016) presents an incremental approach for the minimization of the dynamic compliance in the presence of prescribed low or high value of the excitation frequency, Yang and Li (2014) minimizes the dynamic compliance of structures at resonance frequencies in thermal environments, whereas the frequent case of harmonic excitation is investigated in Liu et al. (2015).

Given the scenario above, this paper introduces an innovative formulation for the topology optimization of Multi-Input (i.e., multi-loaded) Multi-Output (i.e., multi-objective) structures. The proposed framework includes classic compliance optimization as a representative case but is general enough to allow extensions to alternative definitions of compliance as well as to the adoption of other merit functions that may be more appropriate for specific applications. The main idea behind the proposed approach is to write explicitly the input/output transfer matrix, say $\boldsymbol{G}$, that maps the acting loads to the structural response. Using the adopted format, the structural response vector (that more often than not goes under the name of output vector) can be any linear combination of the state vector, i.e., it may include any nodal displacement of the discretized structure. Interestingly enough, the structural compliance (that is the goal function used in nearly all the applications of the literature) may inherently be written as a linear combination of those structural displacements where a load is applied (and in fact the loads themselves are the coefficients of such linear combination as shown in the sequel of the paper). Any optimal design pursued herein is the one that minimizes a proper measure of "*amplitude*" of the transfer matrix $\boldsymbol{G}$. From an engineering perspective, minimizing such amplitude amounts to minimize the effects of the loads onto the structure. It is therefore clear that a proper "tool" to assess the amplitude of the transfer matrix is the key ingredient to fully define the proposed approach. The Singular Value Decomposition (SVD) (Brunton and Kutz 2019; Strang 2019) is shown to be such a tool and more in that not only it does allow to compute a few matrix norms that represent the needed amplitude but also shed some light onto the system response in static and dynamic regime as well. The fact that such norms are uniquely defined in terms of singular values of $\boldsymbol{G}$ simplifies dramatically the computation of the sensitivities that are reduced to the evaluation of the gradients of the singular values themselves with respect to the design variables.

The paper is organized as follows. Section 2 introduces the properties of the singular value decomposition focusing on its engineering interpretation. The fact that input and output singular vectors are mapped onto each other with a gain that is the associated singular value is the reason why the proposed approach is in a sense the most natural to address static and dynamic topology optimization problems, including the celebrated minimum compliance one. The section is concluded by an explicit derivation of the transfer matrix in the frequent case of a displacement-based FEM discretization. The topology optimization problem is presented in Sect. 3 along with a semi-analytic derivation of the sensitivities of the transfer matrix entries and of the singular values with respect to the design parameters. Such sensitivities are needed to apply any gradient-based minimization scheme including the method of moving asymptotes (MMA) (Svanberg 1987) that is used herein in a Matlab version made available by Krister Svanberg. The specific choice of the output vector that fully defines the objectives of the design is made in Sect. 4 wherein attention is paid to the compliance or better a few generalized versions of it. The extension to the dynamic problem is analyzed in Sect. 5. The focus therein is not on the (minimal) modifications needed to handle the dynamic case but is conversely on the unifying aspects that static and dynamic problems share. A few convenient computational remedies that take care of the lengthy processes behind the expensive evaluation of system matrix norms are highlighted and framed within the Matlab environment. Section 6 presents the results of a few numerical investigations along with comparisons with similar designs obtained via more classical methods (when available). The focus is on reporting general lessons that may be learned by proper comparisons between all the proposed norms in terms of effectiveness and robustness. Concluding remarks and need for ongoing investigations are left to Sect. 7, whereas the full Matlab code that solves the static topology optimization problem is given in online Appendix 1. The code is provided for completeness and is open for extensions and improvements.

2 Singular Value Decomposition of the input/output transfer matrix

2.1 General properties of the SVD

Upon recalling that a (generally complex) matrix $\boldsymbol{L}$ is unitary if

$$\boldsymbol{L}^H = \boldsymbol{L}^{-1}, \tag{1}$$

where a superposed H denotes conjugate transposition, we observe that any complex $l \times m$ matrix $\boldsymbol{G}$ admits a singular value decomposition (Strang 2019) that reads

$$\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H, \tag{2}$$

where

- $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary $l \times l$ and $m \times m$ matrices, respectively;
- $\boldsymbol{\Sigma}$ is a $l \times m$ matrix that admits the representations

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 \\ \boldsymbol{0} \end{bmatrix}, \text{ if } l \geq m \quad \text{or} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{0} \end{bmatrix}, \text{ if } l \leq m,$$

where

$$\boldsymbol{\Sigma}_1 = \text{ diag } \{\sigma_1, \sigma_2, \dots \sigma_r\}; \ r = \min(l, m), \tag{3}$$

in which r is the rank of $\boldsymbol{G}$ and

$$\overline{\sigma} \equiv \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \equiv \underline{\sigma} > 0. \tag{4}$$

A few developments below are based on the fact that the unitary matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ form orthonormal bases for the output and input spaces of $\boldsymbol{G}$, respectively.

2.1.1 The main property

Equation (2) provides the definition itself of singular value decomposition but should be further elaborated to gain a deeper understanding of the proposed design approach. To this goal, both sides of Eq. (2) are right-multiplied by $\boldsymbol{V}$ so as to obtain (thanks to the unitary property of $\boldsymbol{V}$)

$$\boldsymbol{GV} = \boldsymbol{U\Sigma},$$

that componentwise reads

$$\boldsymbol{G}\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k. \tag{5}$$

Equation (5) is the first of the two milestones on which the proposed design approach rests (the second one is concerned with the concept of matrix norms based on singular values that is tackled in the next paragraph). The column vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$, say $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$, respectively, represent the output and input directions of the system and are dual to each other, i.e., if an input vector (load) in the direction of $\boldsymbol{v}_k$ is considered, the output shall be in the direction of the associated singular vector $\boldsymbol{u}_k$. Since $||\boldsymbol{v}_k||_2 = ||\boldsymbol{u}_k||_2 = 1$, one has

$$\sigma_k(\boldsymbol{G}) = ||\boldsymbol{G}\boldsymbol{v}_k||_2 = \frac{||\boldsymbol{G}\boldsymbol{v}_k||_2}{||\boldsymbol{v}_k||_2}. \tag{6}$$

From an engineering point of view, Eq. (6) is of the utmost importance for two reasons:

- The kth singular value σ_k is the gain of the matrix $\boldsymbol{G}$ in the direction of the k−th input–output singular pair;
- Being singular values sorted in descending order, see Eq. (4), the triple $(\sigma_1, \boldsymbol{v}_1, \boldsymbol{u}_1)$, respectively, define the maximum gain of the response matrix $\boldsymbol{G}$ and the associated input and output directions.

2.2 Singular values and matrix norms

Paragraph 2.1 has introduced the singular value decomposition of the response matrix as an engineering tool for a deeper understanding of the input–output mapping of the structure to be designed. Furthermore, singular values were shown to be the structural gains coupling each singular vector pair. Given this scenario, a natural goal function, say $\mathscr{F}$, should enjoy the following two properties:

1. $\mathscr{F}$ should be a proper norm of the response matrix $\boldsymbol{G}$ (and such norm should preferably be induced by an associated vector norm);
2. The norm $\mathscr{F}$ should be uniquely defined by the singular values (gains) σ_k.

Three possible such norms are as follows:

- $||\boldsymbol{G}||_2 = \sigma_1$ (the so-called 2-norm) that is in fact the supremum of the ratio in (6). The 2-norm satisfies the multiplicative property

$$||G_1 G_2||_2 \leq ||G_1||_2 \cdot ||G_2||_2 \tag{7}$$

  which is a desirable property that shows that the 2-norm is a vector-induced one. As shown later in much detail, it is appropriate to note that in the dynamic case singular values depend on the frequency ω, i.e., $\sigma_1 = \sigma_1(\omega)$, and the induced norm is defined as $\sup_\omega = \sigma_1(\omega)$. Since a sup operator is involved, in dynamics such norm shall be named H_∞ norm but one should always recall that the underlying norm is a 2-norm;
- $||\boldsymbol{G}||_N = \sigma_1 + \sigma_2 + \dots + \sigma_r$ that is often referred to as the Nuclear norm. All plant's directions concur to the definition of the goal function and therefore the frequency range wherein the design is effective is expected to be wider than the H_∞–norm case that however performs better as far as maximum peak reduction is concerned;
- $||\boldsymbol{G}||_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots \sigma_r^2}$, i.e., the so-called Frobenius norm. Although neither is the Frobenius norm an induced one nor it enjoys the multiplicative property (7), numerical results to follow show that the Frobenius norm works properly and is a viable choice when the primary goal is the robustness of the response.

To conclude this section with, the following key features of the singular value decomposition are worth being emphasized:

- the singular values give direct info on the input/output gains of the system;

- the structure directions $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ are mutually orthogonal and represent the input–output directions of the system;
- unlike eigenvalue-based approaches, the SVD applies inherently to non-square structures. Most response matrices $\boldsymbol{G}$ are in fact non-square since the number of inputs and outputs are usually different from each other. (There is actually no reason why they should be the same).

2.3 The steady-state input/output response matrix *G*

Aim of this paragraph is to derive the abstract framework for the static state and output equations that fits the proposed SVD-based design approach. To start off, equations governing the static response of the structure under optimization are written. For the sake of simplicity, reference is made hereinafter to a displacement-based finite element discretization, but mixed stress–displacement approaches based on the Hellinger–Reissner functional or three-field (stress, strain, displacement) formulations arising from the Hu–Washizu functional might be considered as well. A classical displacement-based equilibrium equation reads

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{B}\boldsymbol{f}, \tag{8}$$

where $\boldsymbol{K}$ is the $n \times n$ stiffness matrix, $\boldsymbol{x}$ is the $n \times 1$ column vector of the unknown displacements, $\boldsymbol{B}$ is an $n \times r$ load matrix, and $\boldsymbol{f}$ is a $r \times 1$ Boolean column vector that distributes the loads to the appropriate degrees of freedom. It should be noticed that constrained displacements and dual reactive forces have been ruled out from Eq. (8) so that n denotes the number of free degrees of freedom of the problem.

The output equation is then used to define the goals of the design. Each row of the $n_o \times 1$ output vector z represents a physical quantity of interest for the designer. When compliance minimization is pursued $n_o = 1$ so that z is actually a scalar quantity. The output equation is eventually written as follows:

$$z = \boldsymbol{C}\boldsymbol{x}, \tag{9}$$

where $\boldsymbol{C}$ is the $n_o \times n$ output matrix that is responsible for the definition of the quantities of interest for the designer.

Upon combining Eqs. (8) and (9), i.e.,

$$\begin{cases} \boldsymbol{K}\boldsymbol{x} &= \boldsymbol{B}\boldsymbol{f} \\ z &= \boldsymbol{C}\boldsymbol{x}, \end{cases}$$

one gets the input–output mapping that reads

$$z = \boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{B}\boldsymbol{f} = \boldsymbol{G}\boldsymbol{f}, \tag{10}$$

that is given a classic system representation in Fig. 1 where the dependence of $\boldsymbol{G}$ on the design variable vector $\boldsymbol{p}$ has been made explicit.

The $n_o \times r$ steady-state (static) transfer matrix eventually reads

$$\boldsymbol{G} = \boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{B}, \tag{11}$$

where one should notice that in the frequent case of a single load case ($r = 1$) and a single scalar output ($n_o = 1$), as it happens, e.g., for compliance minimization of a structure acted upon by a single load, $\boldsymbol{G}$ is actually a scalar quantity.

3 *G*-based topology optimization

The typical $\boldsymbol{G}$-based optimal design problem eventually reads

$$\begin{cases} \min\limits_{\boldsymbol{p}} \mathscr{F}(\boldsymbol{p}) \;\; = \|\boldsymbol{G}(\boldsymbol{p})\| \\ \text{s.t.} \qquad \boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{C}\boldsymbol{K}^{-1}(\boldsymbol{p})\boldsymbol{B} \\ \qquad\qquad V(\boldsymbol{p}) \leq V_{\max} \\ \qquad\qquad \boldsymbol{p}_{\min} \leq \boldsymbol{p} \leq \boldsymbol{p}_{\max} \end{cases}, \tag{12}$$

where

- $(12)_1$ defines the function $\mathscr{F}(\boldsymbol{p})$ to be minimized. The matrix norm being used for each specific problem should be specified (see paragraph 2.2);
- $(12)_2$ is the definition itself of the input/output transfer matrix wherein the dependence on the vector of the design variables $\boldsymbol{p}$ has been made explicit;
- $(12)_3$ is the standard constraint on the maximum volume of the structure. From a computational viewpoint, it may be useful to impose also a lower bound to the overall volume, i.e.,

  $$V_{\min} \leq V(\boldsymbol{p}) \leq V_{\max}.$$

  Although not compulsory, in the very first iterations of the optimization process a lower bound on the volume

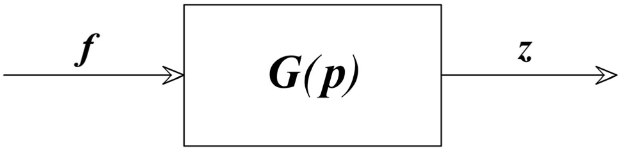

Fig. 1 Block representation of the input–output mapping

may help the numerical solver avoid unfeasible “super-light” solutions in the design space;
- The inequalities on the design variables in (12)$_4$ are to be intended componentwise. As usual, each design variable is bounded between zero (void) and one (full material) within a standard SIMP idealization (Bendsøe and Sigmund 1999).

The optimization problem (12) is solved numerically using the method of moving asymptotes (MMA) (Svanberg 1987). To this goal the user must provide the current values of the goal function and the constraints, along with their sensitivities with respect to the design variables. When the objective function to be minimized is a matrix norm that depends on its singular values, using the chain rule one should therefore provide a semi-analytical computation of the sensitivity of the transfer matrix and of the singular values with respect to any design variable, say p_k. This is done below and implemented in the Matlab code in online Appendix 1.

3.1 Sensitivity of the input/output transfer matrix

In the most general case assume the input/output matrix $\boldsymbol{G}$ depends on the design variable vector $\boldsymbol{p}$ through all its three factors, i.e.,

$$\boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{C}(\boldsymbol{p})\boldsymbol{K}^{-1}(\boldsymbol{p})\boldsymbol{B}(\boldsymbol{p}). \tag{13}$$

In fact, when, for example, seismic forces are considered, one may show that the matrix $\boldsymbol{B}$ does depend on the design variable vector $\boldsymbol{p}$ (Venini and Ceresa 2018) and that is why the dependence of all the factors in Eq. (13) on $\boldsymbol{p}$ should in general be considered. Upon recalling that the derivative of the inverse reads

$$\frac{\partial \boldsymbol{K}^{-1}}{\partial p_k} = -\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial p_k}\boldsymbol{K}^{-1}, \tag{14}$$

using the product rule, the derivative of $\boldsymbol{G}(\boldsymbol{p})$ with respect to the kth component p_k of the design variable vector writes

$$\begin{aligned}\frac{\partial \boldsymbol{G}(\boldsymbol{p})}{\partial p_k} &= \frac{\partial \boldsymbol{C}}{\partial p_k}\boldsymbol{K}^{-1}\boldsymbol{B} + \boldsymbol{C}\frac{\partial \boldsymbol{K}^{-1}}{\partial p_k}\boldsymbol{B} \\ &\quad + \boldsymbol{C}\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{B}}{\partial p_k},\end{aligned}, \tag{15}$$

where the derivative of the stiffness matrix $\boldsymbol{K}(\boldsymbol{p})$ with respect to the design variable p_k that appears in Eq. (14) is computed semianalytically (Andreassen et al. 2011). When the dependence of $\boldsymbol{G}$ on $\boldsymbol{p}$ is through $\boldsymbol{K}$ only, Eq. (15) reduces to

$$\begin{aligned}\frac{\partial \boldsymbol{G}(\boldsymbol{p})}{\partial p_k} &= \boldsymbol{C}\frac{\partial \boldsymbol{K}^{-1}}{\partial p_k}\boldsymbol{B} = \\ &= -\boldsymbol{C}\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial p_k}\boldsymbol{K}^{-1}\boldsymbol{B}\end{aligned}. \tag{16}$$

3.2 Sensitivity of the singular values

Under certain hypotheses of regularity (that we certainly supposed to be met hereinafter) direct differentiation of Eq. (2) allows to arrive at the following formula for the sensitivity of the generic h-th singular value σ_h (see also Strang 2019):

$$\frac{\partial \sigma_h}{\partial p_k} = \mathscr{R}\left(\boldsymbol{u}_h^H \frac{\partial \boldsymbol{G}(\boldsymbol{p})}{\partial p_k}\boldsymbol{v}_h\right), \tag{17}$$

where $\mathscr{R}$ indicates the real part of a (possibly) complex number.

3.3 Sensitivity of the norm of the input/output transfer matrix

Since 2, Nuclear and Frobenius norms of $\boldsymbol{G}$ are uniquely defined by its singular values, Eq. (17) coupled to a proper usage of the chain rule allows the computation of any norm sensitivity in (12)$_1$. The sensitivity of any of the three norms of interest eventually reads

$$\frac{\partial ||\boldsymbol{G}||}{\partial p_k} = \sum_{h=1}^{r}\frac{\partial ||\boldsymbol{G}||}{\partial \sigma_h}\frac{\partial \sigma_h}{\partial p_k}, \tag{18}$$

where the second factor at the right-hand-side is given in Eq. (17), whereas the first factor is as follows:

$$\begin{aligned}\frac{\partial ||\boldsymbol{G}||_2}{\partial \sigma_h} &= \frac{\partial \sigma_1}{\partial \sigma_h} = \delta_{1h} \quad \text{(Kronecker delta)} \\ \frac{\partial ||\boldsymbol{G}||_N}{\partial \sigma_h} &= \frac{\partial(\sigma_1 + \ldots + \sigma_r)}{\partial \sigma_h} = 1 \\ \frac{\partial ||\boldsymbol{G}||_F}{\partial \sigma_h} &= \frac{\sigma_h}{\sqrt{\sigma_1^2 + \ldots + \sigma_r^2}}\end{aligned}. \tag{19}$$

4 Compliance definition (revisited and extended) and more

As already observed most of the times topology optimization aims at the minimization of the compliance. Therefore, even though the proposed approach may handle many different goal functions, a specific derivation of the compliance minimization problem seems to be appropriate. For the sake of simplicity, a lumped parameters system is referred to next. However, the adopted formulation is general in that it applies to any system such that

- $\boldsymbol{x}$ is the state vector grouping all the generalized displacements emerging from a suitable finite element discretization;
- $\boldsymbol{s} \subset \boldsymbol{x}$ is the subset of the displacements that is loaded by the force vector $\boldsymbol{f}$ that is the dual of $\boldsymbol{s}$ in the virtual work sense.

4.1 Notations

Reference is made to Fig. 2 where the state vector reads

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{20}$$

$\boldsymbol{f} = \{f_1, f_2\}$ is the set of applied loads and $\boldsymbol{s} = \{x_1, x_2\} \subset \boldsymbol{x}$ is the set of displacements that is dual to $\boldsymbol{f}$ in virtual work sense.

The optimization problem associated to the system in Fig. 2 is far from being unique in that different choices of load cases and combinations on the input side, see Eq. (8), and design objectives on the output side give rise to a wide variety of engineering problems of interest as discussed below. Each problem fits the framework of Eq. (12) and is fully characterized by the stiffness matrix $\boldsymbol{K}$, the load matrix $\boldsymbol{B}$, and the output matrix $\boldsymbol{C}$. For the case at hand $n = 3$ and the 3×3 stiffness matrix is shared by all possible problems. The designer should conversely select the two matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ to derive the optimization setting that fits better the actual engineering problem to be solved.

4.2 The SISO case: single load case and standard compliance minimization

In this case, i.e., the standard compliance minimization in the presence of a single load case, $n_o = 1$, $r = 1$ and one writes

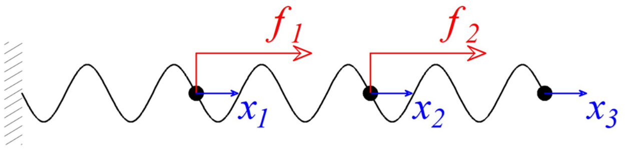

Fig. 2 System, loads, and displacements

$$\boldsymbol{B} = \begin{bmatrix} f_1 \\ f_2 \\ 0 \end{bmatrix}, \quad \boldsymbol{f} = 1, \quad \boldsymbol{C} = \begin{bmatrix} f_1 & f_2 & 0 \end{bmatrix} = \boldsymbol{B}^T, \tag{21}$$

where it should be noticed that the output vector $\boldsymbol{z}$ reduces to a scalar and reads:

$$z = \boldsymbol{Cx} = \begin{bmatrix} f_1 & f_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = f_1 x_1 + f_2 x_2 \equiv \mathscr{C}, \tag{22}$$

i.e., the classical compliance (Andreassen et al. 2011). In this SISO case $\boldsymbol{G}$ is a scalar as is the "matrix" of singular values that reads $\boldsymbol{\Sigma} = \sigma_1$. It should be noted that in the SISO case the three cited norms (2, Nuclear and Frobenius) coincide.

4.3 The MISO case: two load cases and standard compliance minimization

This case is non standard in that $r = 2$, i.e., two independent load cases are considered, whereas $n_o = 1$, i.e., the output is a scalar compliance. One may write

$$\boldsymbol{B} = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \\ 0 & 0 \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} f_1 & f_2 & 0 \end{bmatrix}, \tag{23}$$

where the output vector is formally the same as the SISO one but now the overall compliance $\mathscr{C}$ is the sum of two independent ones, say $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively, due to the loads f_1 and f_2, i.e.,

$$z = \begin{bmatrix} f_1 & f_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = f_1 x_1 + f_2 x_2 = \mathscr{C}_1 + \mathscr{C}_2 = \mathscr{C}. \tag{24}$$

This resembles the multiple load formulation of Sect.5.2 in the celebrated paper (Andreassen et al. 2011) by Sigmund and co-workers. A further property enjoyed by the MISO case should be highlighted. Having two inputs and a single output, the singular input matrix $\boldsymbol{V}$ and the singular value matrix $\boldsymbol{\Sigma}$ may be, respectively, written as

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \end{bmatrix}. \tag{25}$$

The only non-vanishing singular value σ_1 gives the plant gain, whereas the associated first right (input) singular vector $\boldsymbol{v}_1$ gives the factors of the linear combination of the two loads f_1 and f_2 that stresses the structure the most, i.e.,

$$\boldsymbol{v}_{11} f_1 + \boldsymbol{v}_{12} f_2. \tag{26}$$

On the opposite side, the second right singular vector $\boldsymbol{v}_2$ is associated to the vanishing singular value $\sigma_2 = 0$. Therefore, the second input singular vector $\boldsymbol{v}_2$ gives the coefficients of the linear combination

$$\boldsymbol{v}_{21} f_1 + \boldsymbol{v}_{22} f_2, \tag{27}$$

to which a null compliance is associated. The crucial role played by (26) as far as the optimal topology is concerned should be highlighted: as long as the two loads f_1 and f_2 are independent of one another, i.e., when they belong to different load cases so that $r = 2$, the proposed approach provides the designer with an optimal solution that is based on the automatic selection of the worst possible load combination that is given by the components of $\boldsymbol{v}_1$ and characterized by a gain that is σ_1. No role is conversely played by the load combination (27). Being the system a single output one, i.e., $\sigma_2 = 0$ in (25), the rank of $\boldsymbol{G}$ is $r = 1$ and therefore the plant direction $\boldsymbol{v}_2$ is annihilated or, in engineering words, the relevant compliance is zero whenever the two loads f_1 and f_2 are combined according to (27).

4.4 The SIMO case: single load case and minimization of the compliance vector norm

The peculiarity of this (and next) case is its output z. The compliance associated to each loaded point, say $\mathscr{C}_k = f_k u_k$, is no longer added to define a unique scalar compliance $\mathscr{C} = \sum_k \mathscr{C}_k$, but it conversely defines an independent entry of the compliance vector z. For the problem at hand $r = 1$ and $n_o = 2$ and one writes

$$\boldsymbol{B} = \begin{bmatrix} f_1 \\ f_2 \\ 0 \end{bmatrix}, \quad \boldsymbol{f} = 1, \quad \boldsymbol{C} = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \end{bmatrix}, \tag{28}$$

such that the output vector reads

$$z = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} f_1 x_1 \\ f_2 x_2 \end{bmatrix} \equiv \begin{bmatrix} \mathscr{C}_1 \\ \mathscr{C}_2 \end{bmatrix}. \tag{29}$$

The rectangular matrix of singular values reads

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 \\ 0 \end{bmatrix}, \tag{30}$$

that means that $\mathscr{C}_1$ and $\mathscr{C}_2$ are automatically combined according to the components of the first output vector $\boldsymbol{u}_1$, i.e.,

$$u_{11} \mathscr{C}_1 + u_{12} \mathscr{C}_2 \tag{31}$$

and such combination is the one that determines the final optimal topology to which the maximum gain σ_1 is associated. On the other side, the combination

$$u_{21} \mathscr{C}_1 + u_{22} \mathscr{C}_2 \tag{32}$$

is associated to a null gain $\sigma_2 = 0$ and is therefore annihilated by the computations. As a matter of fact (31) is the SIMO dual of the MISO input combination (26) in that they both are the combinations, input and output, respectively, that govern and determine the optimal topologies. Likewise, (32) and (27) are dual to each other in that none of them plays any role toward the computation of the final optimal topologies being associated to $\sigma_2 = 0$.

4.5 The MIMO general case: two load cases and minimization of the compliance vector norm

This is the way more general formulation. One has two inputs and two outputs, i.e., $r = 2$ and $n_o = 2$, thus generating a 2×2 transfer matrix $\boldsymbol{G}$. As for load and output matrices one may write

$$\boldsymbol{B} = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \\ 0 & 0 \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \end{bmatrix} = \boldsymbol{B}^T, \tag{33}$$

the output vector z being the same as in Eq. (29). In this case ($2 = r > 1$ and $2 = n_o > 1$), two non-vanishing singular values show up and one may write

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}, \tag{34}$$

where σ_1 and σ_2 are, respectively, associated to the structure directions $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ that maximize and minimize the plant gains. In the MIMO case, a comparison between the choice of the 2-norm on the one side versus the Nuclear and Frobenius norms on the other is appropriate. Being $r = n_o = 2 > 1$, no plant direction is annihilated (unlike previous cases) and both the triples $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ and $(\sigma_2, \boldsymbol{u}_2, \boldsymbol{v}_2)$ contribute to the output response. Being however $\sigma_1 > \sigma_2$, the first triple plays a more important role than the second. That said, the 2-norm is focused on the first triple only, i.e., the one that maximizes the plant gain, whereas using the nuclear and Frobenius norms all plant directions enter the formulation, including those characterized by smaller gains (σ_2 for the case at hand).

5 Dynamic topology optimization fits the proposed formulation

By definition itself, the proposed topology optimization approach of Eq. (12) may be applied to any system whenever the following properties are met:

1. A transfer matrix $\boldsymbol{G}(\boldsymbol{p})$ is available that governs the input–output mapping of which the dependence on the design variable vector $\boldsymbol{p}$ is known;
2. A suitable norm of such matrix $\boldsymbol{G}(\boldsymbol{p})$ can be computed along with its sensitivities with respect to the design variables.

When these two properties hold true, the proposed formulation is abstract enough to allow its applicability, no matter the specific problem at hand and in particular no matter whether the system is static or dynamic. For the latter to fit the proposed formulation, a Laplace (Fourier) domain setting is adopted wherein the transfer matrix $\boldsymbol{G}$ is readily available in an overall algebraic framework. In the dynamic case, though, the transfer matrix is no longer constant but depends on the Laplace variable s and eventually on the frequency ω (and this matter shall affect the definitions themselves of matrix norms).

5.1 The dynamic transfer function

The more convenient way to arrive at the Laplace-domain formulation of the dynamic problem is to start with the time-domain state space form and take its Laplace transform. The state space form reads

$$\begin{cases} \boldsymbol{E}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{f} \\ z &= \boldsymbol{C}\boldsymbol{x} \end{cases}, \tag{35}$$

where a superposed dot indicates time differentiation and

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}, \ \boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K} & -\boldsymbol{D} \end{bmatrix}, \ \boldsymbol{x} = \begin{bmatrix} \boldsymbol{u} \\ \dot{\boldsymbol{u}} \end{bmatrix}, \tag{36}$$

in which

- $\boldsymbol{K}$, $\boldsymbol{D}$, $\boldsymbol{M}$, and $\boldsymbol{I}$ are the stiffness, damping, mass, and identity matrices, respectively. In what follows a Rayleigh damping is considered such that

  $$\boldsymbol{D} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}, \tag{37}$$

  where α and β are positive constants (small enough not to overdamp the system). However, the formulation is not limited to Rayleigh damping and any viscous damping could be adopted;
- $\boldsymbol{x}$ is the state vector that piles the displacement and velocity vectors $\boldsymbol{u}$ and $\dot{\boldsymbol{u}}$, respectively;
- z is the output vector that is defined through a proper choice of the output matrix $\boldsymbol{C}$.

Taking the Laplace transform of $(35)_1$ and $(35)_2$ and ruling out the Laplace transform of the state vector $\boldsymbol{x}$ one gets the Laplace-domain input/output relationship and the desired transfer matrix $\boldsymbol{G}(s)$, i.e.,

$$\begin{array}{c} \boldsymbol{Z}(s) = \left[\boldsymbol{C}(s\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}\right]\boldsymbol{F} \\ \Downarrow \\ \boldsymbol{G}(s) = \boldsymbol{C}(s\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B} \end{array}. \tag{38}$$

Upon choosing $s = i\omega$ one gets the frequency response matrix that is the dynamic counterpart of the static one in $(12)_2$:

$$\boldsymbol{G}(i\omega;\boldsymbol{p}) = \boldsymbol{C}(i\omega\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}. \tag{39}$$

The dependence of $\boldsymbol{G}(i\omega;\boldsymbol{p})$ on the frequency $\omega \in (0, \infty)$ calls for a generalization of the three norms previously introduced as discussed below.

5.2 Matrix norm definitions in the dynamic case

The norms in Sect. 2.2 are extended to the dynamic case in which $\boldsymbol{G}$ is no longer constant but depends on the frequency ω. This is done through an operation that is conceptually trivial but computationally extremely expensive: the supremum over the frequency axis in the H_∞ and Nuclear norm cases, a suitable integral average in the H_2 case. One may write

- H_∞ norm: $||\boldsymbol{G}||_\infty = \sup_\omega \sigma_1(\omega)$ where again the slight change of notation should be noticed. In statics this was named a 2-norm in view of the fact that it was defined as a ratio between the 2-norm of the output and input vectors. In dynamics, such norm is named H_∞ since a supremum is taken over the frequency axis;
- nuclear norm: $||\boldsymbol{G}||_N = \sup_\omega \sigma_1(\omega) + \ldots + \sigma_r(\omega)$;
- the H_2-norm is the generalization of the algebraic Frobenius norm to dynamical systems. Unlike the H_∞-norm, however, such generalization is not performed using a sup operator, such as

  $$\sup_\omega \sqrt{\sigma_1^2(\omega) + \ldots + \sigma_r^2(\omega)},$$

  but via an integral average that reads

$$||\boldsymbol{G}||_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty}\sum_{i=1}^{r}\sigma_i^2(\omega)\mathrm{d}\omega}. \tag{40}$$

However, Eq. (40) is neither practical for computing the H_2 norm nor its sensitivity. A more viable approach is then recalled in Sect. 5.3 below.

5.3 Computing the H_2 norm and its sensitivity

The first step to compute $||\boldsymbol{G}||_2$ in the dynamic case is to evaluate the contrallability Gramian $\boldsymbol{W}$ that solves the generalized Lyapunov equation:

$$\boldsymbol{AWE} + \boldsymbol{EWA}^H + \boldsymbol{BB}^H = \boldsymbol{0}. \tag{41}$$

Then, one may show that the following relation holds true (Francis 1987):

$$||\boldsymbol{G}||_2 = \sqrt{\text{trace}\ (\boldsymbol{CWC}^H)}. \tag{42}$$

5.3.1 Semi-analytic H_2-norm sensitivity

As for the derivative of $||\boldsymbol{G}||_2$ with respect to the design parameter p_k, one may write by direct differentiation of (42):

$$\frac{\partial ||\boldsymbol{G}||_2}{\partial p_k} = \frac{1}{2||\boldsymbol{G}||_2}\ \text{trace}\ \left[\boldsymbol{C}\frac{\partial \boldsymbol{W}}{\partial p_k}\boldsymbol{C}^H\right], \tag{43}$$

in which the derivative of the Gramian $\boldsymbol{W}$ with respect to p_k may be computed by solving the generalized Lyapunov equation one gets by direct differentiation of (41):

$$\boldsymbol{AW'E} + \boldsymbol{EW'A}^H + \boldsymbol{S} = \boldsymbol{0}, \tag{44}$$

where a superposed prime denotes differentiation with respect to p_k and the pseudo-load $\boldsymbol{S}$ is given as follows:

$$\boldsymbol{S} = \boldsymbol{A'WE} + \boldsymbol{AWE'} + \boldsymbol{E'WA}^H + \boldsymbol{EWA}^{H\prime}. \tag{45}$$

5.3.2 Finite difference H_2-norm sensitivity

The semi-analytic sensitivity of Eq. (43) calls for the solution of an additional generalized Lyapunov equation, the pseudo-load of which is quite a complex one, i.e., $\boldsymbol{S}$ in (45). A fast solver for the generalized Lyapunov equation was coded in the Matlab add-on Saak et al. (2022) that performs at its best when the $\boldsymbol{B}$ matrix is a "thin" rectangular one, a property that is not enjoyed by $\boldsymbol{S}$ in (45). It is therefore more convenient to compute the H_2-norm sensitivity using a finite difference approximation that also involves the solution of a generalized Lyapunov equation. The $\boldsymbol{B}$ matrix of the latter, though, is the same as the one of the actual system and is therefore ideally suited for the code in Saak et al. (2022) to work full throttle.

5.4 The dynamic topology optimization problem

The dynamic topology optimization problem eventually writes

$$\begin{cases} \min\limits_{\boldsymbol{p}} \mathscr{F}(\boldsymbol{p}) & = ||\boldsymbol{G}(i\omega,\boldsymbol{p})|| \\ \text{s.t.} & \qquad \boldsymbol{G}(i\omega;\boldsymbol{p}) = \boldsymbol{C}(i\omega\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B} \\ & \qquad V(\boldsymbol{p}) \le V_{\max} \\ & \qquad \boldsymbol{p}_{\min} \le \boldsymbol{p} \le \boldsymbol{p}_{\max} \end{cases}, \tag{46}$$

that is formally the same as the static one in (12), provided the respective definitions of matrix norms (Sect 2.2 versus Sect 5.2) and transfer matrices (Eq. 11 versus Eq. 39). A few considerations on how to compute the norms and their sensitivities when the frequency response matrix depends on ω are in order. The scope here is to show how these norms can be computed using, for example, Matlab, whereas details concerning the algorithms themselves are beyond the scopes of this paper and may be found in Bruisma and Steinbuch (1990) among others.

5.4.1 On the numerical computation of $||\boldsymbol{G}(i\boldsymbol{\omega}, \boldsymbol{p})||$

Computing any of the three norms of $\boldsymbol{G}(i\omega;\boldsymbol{p})$ (for fixed $\boldsymbol{p}$) represents a formidable computational task. The H_∞ norm is by far the one that has received more attention by the scientific community, mainly because of its relevance in the framework of robust control of closed-loop systems (Francis 1987). The Matlab function `norm` (or better its version made available in the add-on `sss Toolbox` (MORLab 2017) that works on sparse matrices) may be used for this purpose: provided as inputs a suitable version of the transfer function $\boldsymbol{G}$ and the flag `inf`, the function `norm` returns the H_∞-norm of $\boldsymbol{G}$ and the frequency at which the norm is attained. The latter frequency is needed to compute the sensitivities as shown below. Using the same function `norm` with the flag 2, the Frobenius norm is retrieved. However, when the system size is 10^5 or more as is the case of the examples below, all available norm functions become dramatically slow and the very many calls required to achieve convergence within an optimization session make their usage not convenient if not impractical. The way out adopted for the simulations to come is a sub-optimal strategy described below that applies to the Nuclear norm as well for which ad hoc methods are not available.

5.4.2 A viable sub-optimal approach for computing the H_∞ and the Nuclear norms of a large scale dynamic system

Given a dynamic system with transfer matrix $\boldsymbol{G}(i\omega)$, the proposed method for a sub-optimal computation of $||\boldsymbol{G}(i\omega)||_\infty$ and $||\boldsymbol{G}(i\omega)||_N$ works as follows:

1. the frequency range of interest, say $I_\omega = (\omega_1, \omega_2)$, is determined and sampled (uniformly or not);
2. a call to the Matlab function `sigma` is made that returns all the singular values of the dynamic system for all frequencies in I_ω;
3. $||\boldsymbol{G}(i\omega)||_\infty$ and $||\boldsymbol{G}(i\omega)||_N$ are, respectively, computed as follows:

$$||\boldsymbol{G}||_\infty = \max_{\omega \in I_\omega} \sigma_1(\omega),$$

$$||\boldsymbol{G}||_N = \max_{\omega \in I_\omega} \left[\sigma_1(\omega) + \sigma_2(\omega) + \ldots \sigma_r(\omega)\right].$$

The method is only sub-optimal in that the first singular value and the sum of all of them might assume larger values at frequency points that do not belong to the set I_ω, especially if the adopted partition of the frequency axis is a coarse one. One may of course refine the approach using an adaptive sampling strategy that would allow a closer and closer computation of the actual value of the norm. For the sake of this paper, this option is not however exploited but is left for near-future extensions.

5.4.3 On the computation of the sensitivity of $||G(\boldsymbol{i\omega}, \boldsymbol{p})||$

If one compares the static and dynamic definitions of $\boldsymbol{G}$ that are, respectively, given in Eqs. $(12)_2$ and $(46)_2$, it is straightforward to realize that the static sensitivity formulas in (15) and (16) still hold true in the dynamic case provided that

- $\boldsymbol{K}$ is substituted by $i\omega\boldsymbol{E} - \boldsymbol{A}$;
- in such formulas, ω is no longer a variable but takes on the value at which the norm is attained.

6 Numerical results

A few representative numerical results are presented hereafter that are cast in the framework of Sects. 3 and 5 for the static and dynamic cases, respectively.

6.1 Software and third party add-ons

The codes with which all the numerical results were obtained are written in Matlab. The full code that solves the static problem is presented in online Appendix 1 and requires the Matlab implementation of the Method of Moving Asymptotes (MMA) made available by its creator Krister Svanberg, Svanberg (1987). The following add-ons were used for the dynamic code:

- Sparse dynamic models (state space and/or transfer functions) are not handled in plain Matlab. The `sss Toolbox` (MORLab 2017) allows this useful option and extends to sparse systems nearly all functions that require full storage in Matlab;
- Fast matrix computations are a pre-requisite of the proposed code. The `Matrix Equation Sparse Solver (M-M.E.S.S) Toolbox` (Saak et al. 2022) is an ideal tool in this respect, e.g., for solving the generalized matrix Lyapunov Equation (41).

6.2 Problem 1-Statics

6.2.1 Preliminary considerations

The four possible formulations (SISO, MISO, SIMO, and MIMO) are exploited along with the three proposed norms (2, Nuclear and Frobenius). It should be outlined that in the first three cases (SISO, MISO, and SIMO) $r = 1$, i.e., a single singular value is non-trivial so that the three norms happen to be the same and so do the relevant optimal topologies. Therefore, for each of the cases SISO, MISO, and SIMO a single optimal topology shall be shown that is generated using any of the three norms. In the MIMO case, though, two singular values are non-trivial and therefore the optimal designs do depend on the norm being used. Therefore, for the latter MIMO case each design is presented.

6.2.2 General data

The geometry and the loads of the static topology optimization problem are shown in Fig. 3. Thanks to the symmetry of the structure the half-model in Fig. 3 is used in the computations using the following numerical data:

- Young modulus: $E_0 = 1$, $E_{\min} = 10^{-9}$,
- Poisson ratio: $\nu = 0.3$,
- Number of elements in vertical (Y)-direction: 2^6;
- Number of elements in horizontal (X)-direction: 3×2^6;
- Side length: $L = 2^6$ (by this choice a uniform mesh of square elements of unit size is generated);
- $P = Q = 1$ (the load time dependence in Fig. 3 is for use in the dynamic example below);
- Volume fraction $= 0.5$;
- Filter type: density, see Andreassen et al. (2011);
- SIMP penalization factor: 3.

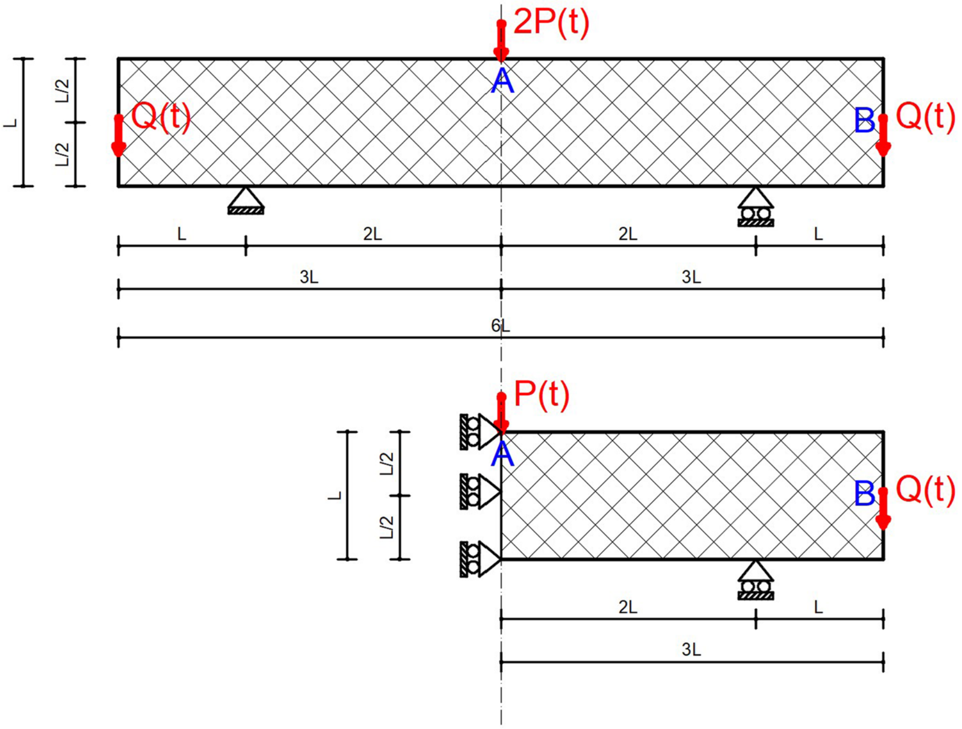

Fig. 3 Numerical example: full and half structure using symmetry

Table 1 SISO system results

	$\|\|\mathscr{C}\|\|_2$	$\|\|\mathscr{C}\|\|_N$	$\|\|\mathscr{C}\|\|_F$
Norm Values	58.71	58.71	58.71
Elapsed Time (s)	124.2	107.58	131.1

6.2.3 SISO formulation

Using a SISO formulation the input is given by the two forces P and Q that belong to the same load case and the output is the overall compliance $\mathscr{C}$. As shown in Table 1, the norms at convergence coincide (as they should), whereas the computational times do not. The CPU time is limited and smaller than two minutes in all cases, see the convergence curves in Fig. 4. The optimal topology is shown in Fig. 5 that is the same as the one given by the 88-line code by Sigmund and co-workers Andreassen et al. (2011). This does not come as a surprise since the problem being solved is the same, even though the underlying formulation is different. As for the input and output plant directions, in the case of a SISO representation both $\boldsymbol{V}$ and $\boldsymbol{U}$ degenerate to the (scalar) unity, whereas the singular value is the norm itself.

6.2.4 MISO formulation

In the MISO formulation, the two loads P and Q are allowed to vary independently of each other within two different load cases. However, a single output compliance is considered that is the sum of the two compliances. Table 2 shows norm values and CPU times needed to perform the computations. The convergence curves are shown in Fig. 10 that exhibits a “nearly” monotonic convergence for all the three norms. The optimal topology (shared by all the three norms) is shown in Fig. 7. A look at the topologies in Figs. 5 and 7 shows that, though similar, the two designs are different. One may wonder whether this represents a violation of the superposition principle according to which as long as the response is linear the output of the sum ($\mathscr{C}$ due to P and Q acting upon the structure simultaneously) is the sum of the outputs ($\mathscr{C} = \mathscr{C}_P + \mathscr{C}_Q$). Of course, no violations, since the structure is not given but being defined iteratively, the updates on the design variables at each iteration need not to be the same and so are the final designs (Fig. 6) .

The input plant direction is given by the singular input vector at convergence that reads

$$\boldsymbol{V} = \begin{bmatrix} 0.727 \\ 0.686 \end{bmatrix}. \tag{47}$$

However, for MISO systems, the concept of plant direction is not as important as one may show that the gain neither depends on the input magnitude nor on its direction (Skogestad and Postlethwaite 1996), i.e., $\boldsymbol{V}$. As shown below, the concept of input direction becomes conversely crucial in the case of MIMO systems.

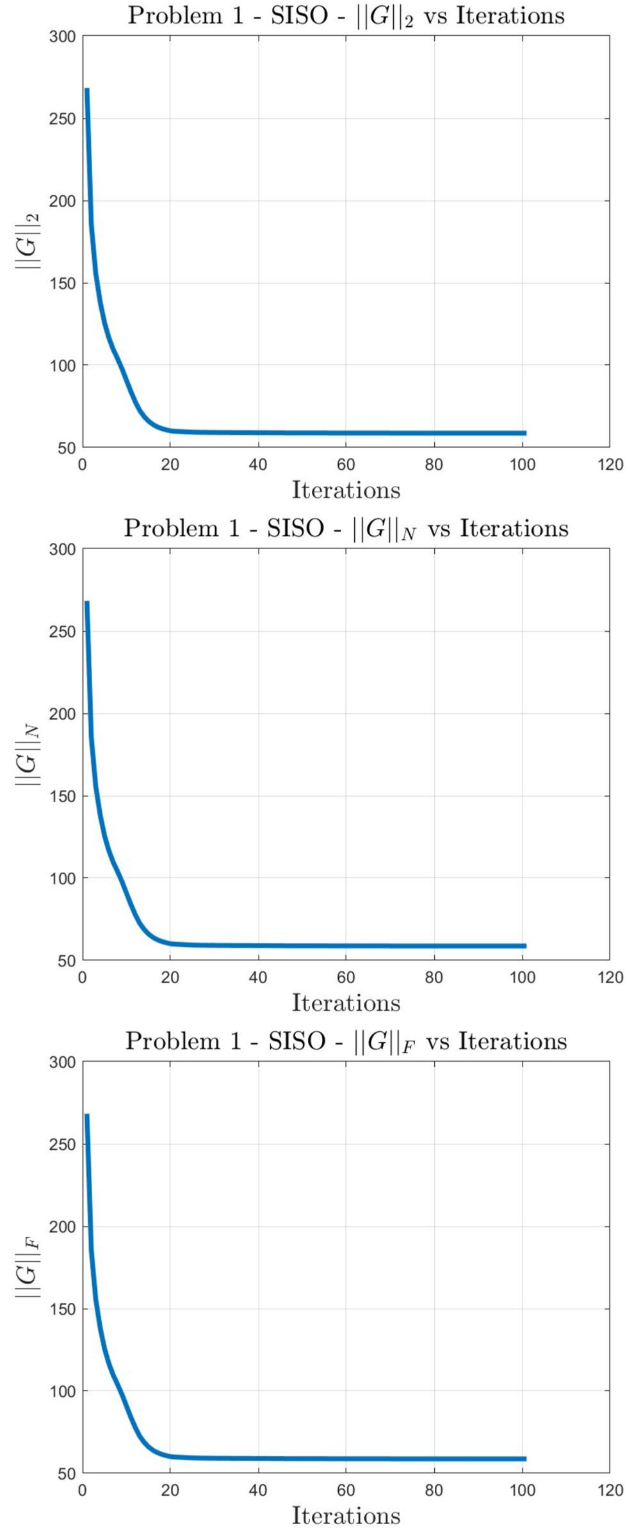

Fig. 4 SISO formulation—Convergence curves: (Up) 2-norm, (Mid) N-norm, and (Bottom) F-norm

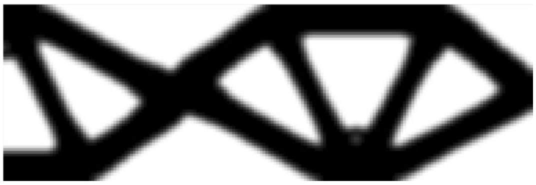

Fig. 5 SISO formulation—Optimal topology: 2-norm, N-norm, and F-norm

Table 2 MISO system results

	$\|\|\mathscr{C}\|\|_2$	$\|\|\mathscr{C}\|\|_N$	$\|\|\mathscr{C}\|\|_F$
Norm Values	42.08	42.08	42.08
Elapsed Time (s)	93.7	111.1	114.4

6.2.5 SIMO formulation

In the SIMO formulation, the two forces P and Q belong to the same load case but the overall compliance is split to form a 2D output vector $z = \left[\mathscr{C}_1 \;\; \mathscr{C}_2\right]^T$. Table 3 shows norm values and CPU time needed to achieve convergence.

Figures 8 and 9 show the convergence paths for the three norms and the optimal topology, respectively.

The optimal topology happens to be the same as the MISO case and interestingly enough the singular output vector reads

$$\boldsymbol{U} = \begin{bmatrix} 0.727 \\ 0.686 \end{bmatrix}, \tag{48}$$

that is the same as the input vector of the SIMO case.

6.2.6 MIMO formulation

As anticipated, the MIMO case is the more general one in that not only are the two forces P and Q free to vary independently of each other within two load cases, but also each compliance $\mathscr{C}_P$ and $\mathscr{C}_Q$ represents an entry of the 2D output vector z. The concepts of plant directions, orthogonality of input and output singular vectors, and singular values seen as plant gains assume the deepest meaning in the MIMO case. Norm values and CPU time needed to achieve convergence are displayed in Table 4.

The optimal topologies obtained using the three norms are shown in Fig. 11. For the sake of completeness, Fig. 12 shows the optimal topology one gets using Sigmund's 88-line code with the multiload option. The four designs are similar but different being based on different optimality conditions. The triples $(\boldsymbol{V}, \boldsymbol{U}, \boldsymbol{\Sigma})$ are reported next for the four designs at convergence:

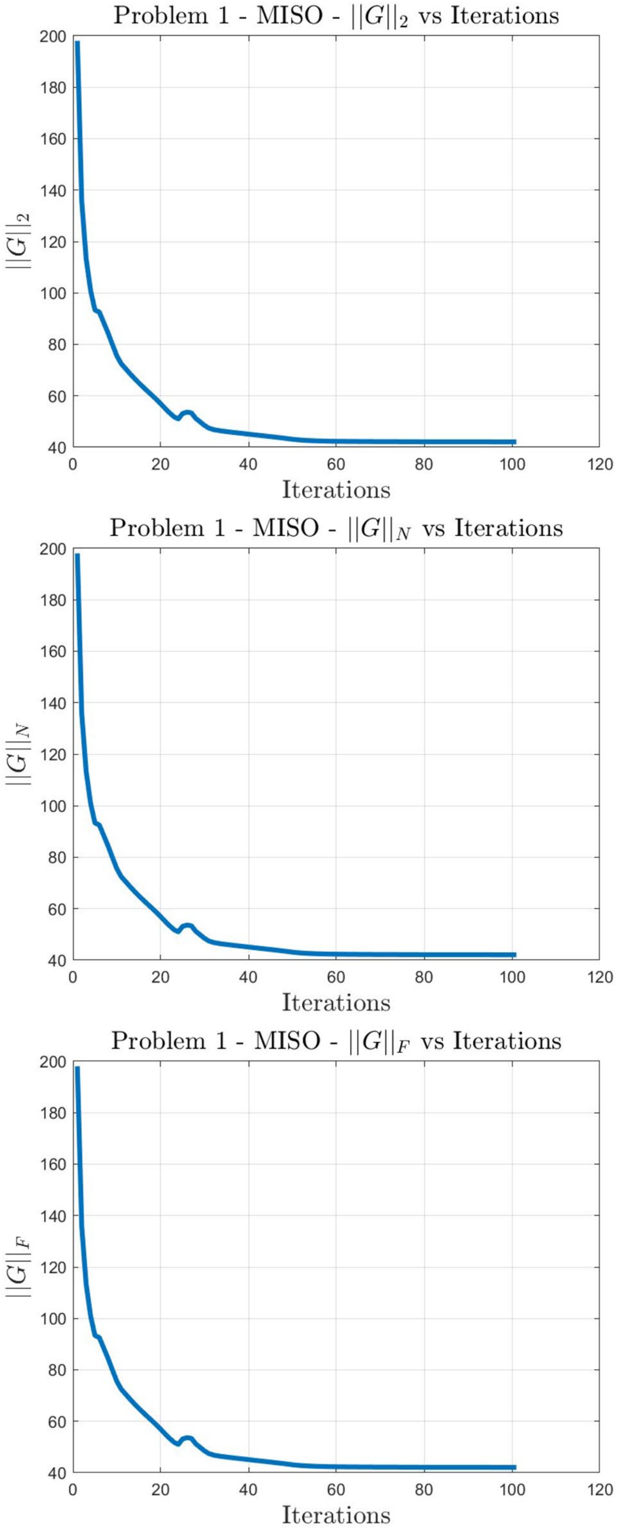

Fig. 6 MISO formulation—Convergence curves: (Up) 2-norm, (Mid) N-norm, and (Bottom) F-norm

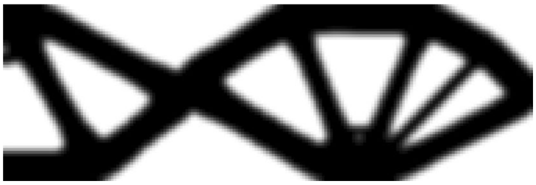

Fig. 7 MISO formulation—Optimal topology: 2-norm, N-norm, and F-norm

Table 3 SIMO system results

	$\|\|\mathscr{C}\|\|_2$	$\|\|\mathscr{C}\|\|_N$	$\|\|\mathscr{C}\|\|_F$
Norm Values	42.08	42.08	42.08
Elapsed Time (s)	93.7	64.18	81.07

- 2-Norm

$$\boldsymbol{V} = \begin{bmatrix} -0.755 & 0.666 \\ 0.666 & 0.755 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} -0.755 & 0.666 \\ 0.666 & 0.755 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 94.16 & 0 \\ 0 & 54.72 \end{bmatrix} \tag{49}$$

- Nuclear norm

$$\boldsymbol{V} = \begin{bmatrix} -0.784 & 0.621 \\ 0.621 & 0.784 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} -0.784 & 0.621 \\ 0.621 & 0.784 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 98.92 & 0 \\ 0 & 38.22 \end{bmatrix} \tag{50}$$

- Frobenius norm

$$\boldsymbol{V} = \begin{bmatrix} -0.771 & 0.636 \\ 0.636 & 0.771 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} -0.771 & 0.636 \\ 0.636 & 0.771 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 96.81 & 0 \\ 0 & 41.57 \end{bmatrix} \tag{51}$$

- 88-line Sigmund's code

$$\boldsymbol{V} = \begin{bmatrix} -0.786 & 0.618 \\ 0.618 & 0.786 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} -0.786 & 0.618 \\ 0.618 & 0.786 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 100.28 & 0 \\ 0 & 38.06 \end{bmatrix} \tag{52}$$

As for the dominating plant direction, it is important to highlight that the first input singular vector $\boldsymbol{v}_1$ (first column of $\boldsymbol{V}$ in Eqs. (49), (50), (51), and (52)) has two components that are opposite in sign and share a similar amplitude. This was

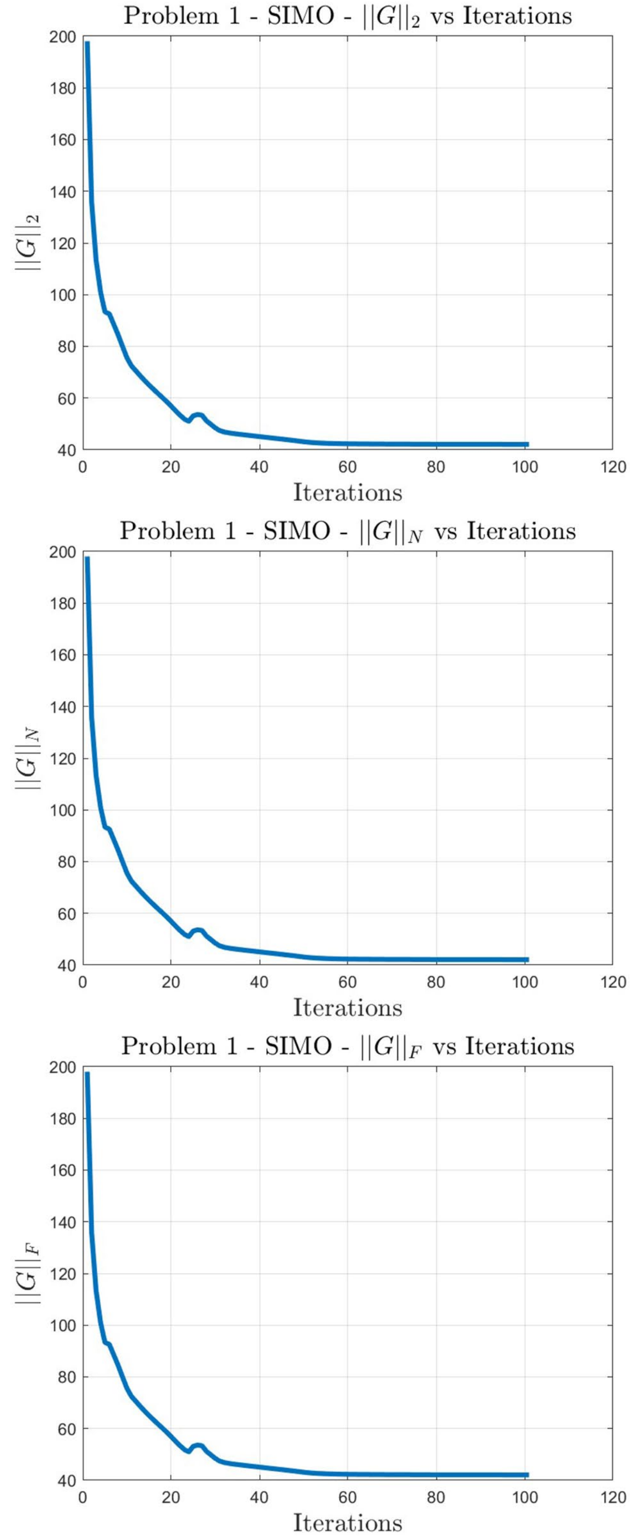

Fig. 8 SIMO formulation—Convergence curves: (Up) 2-norm, (Mid) N-norm, and (Bottom) F-norm

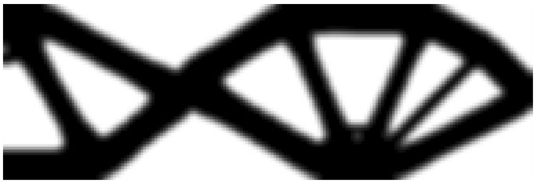

Fig. 9 SIMO formulation—Optimal topology: 2-norm, N-norm, and F-norm

Table 4 MIMO system results

	$\|\|\mathscr{G}\|\|_2$	$\|\|\mathscr{G}\|\|_N$	$\|\|\mathscr{G}\|\|_F$
Norm Values	94.16	137.14	105.36
Elapsed Time (s)	156.47	171.41	177.35

expected because of the cantilevered nature of the beam and the position of the two loads (P and Q, respectively, inside and outside the beam span). As for the performance of the proposed designs, an insight may be gathered by looking at Table 5 that grades the presented designs in terms of the numerical values of the adopted norms. The following considerations are worth being made:

- as expected each design wins its own competition, i.e., any norm is minimized by the formulation that explicitly aimed at its minimization (the relevant values are written in bold on the diagonal of Table 5);
- minimizing the 2-norm, i.e., the maximum gain σ_1 of the system in the direction of the first input and output singular vectors, causes a high penalization on the other plant directions. One may in fact see that for the three cases Nuclear, Frobenius, and Sigmund, the second (and last) singular value is, respectively, equal to 38.22, 41.57, and 38.06, i.e., values that are smaller than half the first gain σ_1. Not so for the 2-norm case for which $\sigma_2 = 54.72$, i.e., a remarkably larger value. As a general guideline, the lesson learned is that using a norm that involves the maximum singular value only is computationally more efficient and inherently leads to the reduction of the maximum gain (which is an amazing result). However, neglecting the effect of smaller singular values may lead to a system response that is poorer overall, especially when σ_2 is close to σ_1;
- for each of the four designs, a measure of overall performance (the smaller the better) may be computed as the SRSS (square root of the sum of the squares) of the differences between scored norm values and best norm values (eventually scored by other approaches). One gets the results shown in Table 6 that again shows that the 2-norm is an extreme one: the maximum gain is reduced the most among all possible design choices but the price

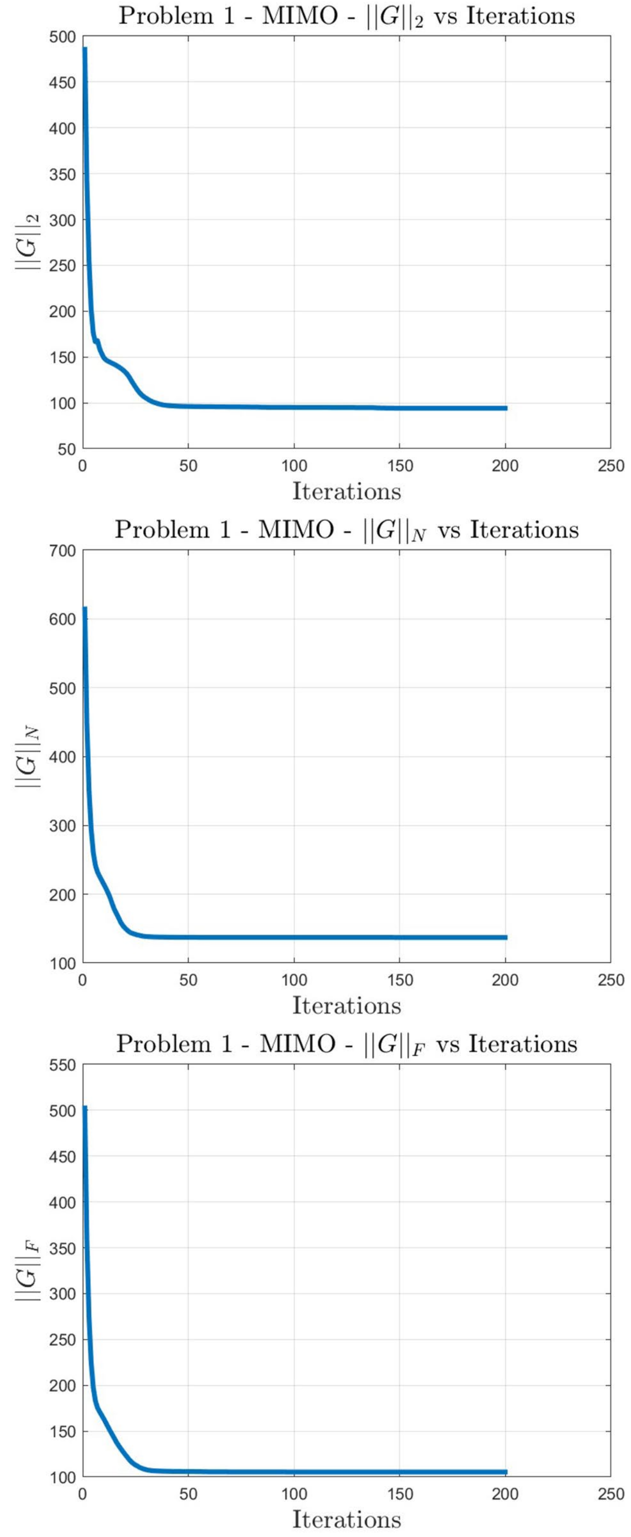

Fig. 10 MIMO formulation—Convergence curves: (Up) 2-norm, (Mid) N-norm, and (Bottom) F-norm

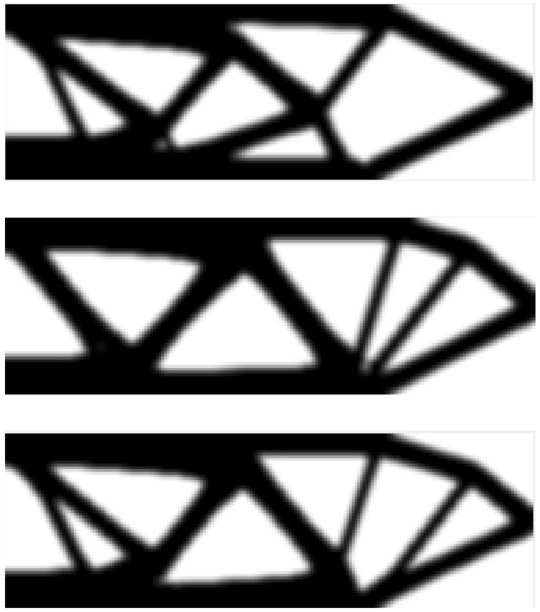

Fig. 11 MIMO formulation—Optimal topologies: (Top) 2-norm, (Mid) N-norm, and (Bottom) F-norm

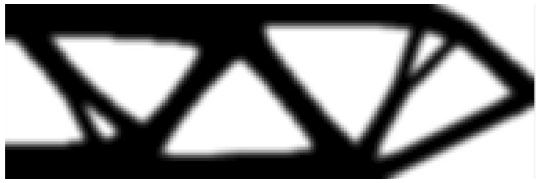

Fig. 12 Sigmund's 88-line code solution using the multiload case option

Table 5 Norm competition between the four available designs

	$\|\|\mathscr{C}\|\|_2$	$\|\|\mathscr{C}\|\|_N$	$\|\|\mathscr{C}\|\|_F$
2-norm design	**94.16**	148.88	108.91
Nuclear norm design	98.92	**137.14**	106.05
Frobenius norm design	96.81	138.38	**105.36**
88-line code based	100.28	138.34	107.26

Table 6 Overall performance of the designs (the smaller, the better)

	Performance index
2-norm design	12.26
Nuclear norm design	4.81
Frobenius norm design	2.93
88-line code based	6.52

to be paid is a weaker behavior with respect to other possible merit functions. Eventually, Sigmund's code design (that is not based on an explicit norm minimization as other designs do) is shown to be quite a robust one in that it performs nicely with respect to all the proposed norms.

6.3 Problem 2-Statics - High-Rise Building

6.3.1 Preliminary considerations

The five-story high-rise building shown in Fig. 13 is investigated next. The objective is to arrive at an optimal bracing structure capable to withstand horizontal environmental actions. This kind of application of topology optimization has been investigated for some fifteen years (see Beghini et al. 2012 as one of the most successful strategies within this framework). This problem is properly suited to test the proposed approach in that its multi-output version is a five outputs challenging one. The following four different designs are pursued:

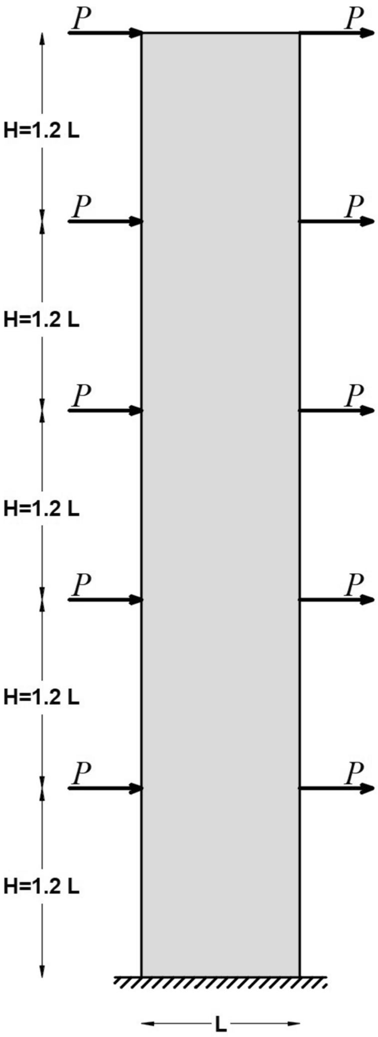

Fig. 13 Five-story high-rise building domain

1. Single-Input Single-Output (SISO);
2. Multi-Input Multi-Output (MIMO), 2-norm;
3. Multi-Input Multi-Output (MIMO), Nuclear norm;
4. Multi-Input Multi-Output (MIMO), Frobenius norm.

The main geometry, mesh, and mechanical parameters are as follows:

- number of stories = 5;
- number of elements in horizontal direction = $2^6 = 64$;
- number of elements per story in vertical direction = $\texttt{floor}(1.2 \times 2^6) = 77$;
- maximum volume fraction = 0.40;
- SIMP exponent = 3.0;
- filter minimum length = 2.25;
- filter type: density.

Convergence curves for all the designs are displayed in Fig. 14 that show a steep monotonic behavior of the MMA-based approach. All the optimal topologies are shown in Fig. 15. The SISO-based bracing system is shown to be quite a stiff one at the base (and is so up to roughly mid-rise). MIMO designs are in general characterized by a more homogeneous material distribution as far as the thickness of the diagonal braced is concerned. From a manufacturability standpoint, the MIMO design using a nuclear norm-based goal function appears to be superior with respect to other designs in that a lower and, so to speak, "cleaner" bracing layout shows up. The performance of three MIMO designs is assessed in Table 7 where all the three norms are computed.

6.4 Problem 3 - Dynamics - MIMO Formulation

6.4.1 Preliminary considerations

Attention is focused on the MIMO formulation only that is the one that exhibits a more general and structurally richer behavior. As for the norms, the H_∞ and the Nuclear norms are considered since they depend on local evaluations of the singular values in such a way that the sub-optimal approach introduced in Sect. 5.4.2 may be used. On the other side, though conceptually applicable, the usage of the H_2-norm has not been exploited because of the prohibitive computational cost of the relevant sensitivities, no matter the function or approximation being used.

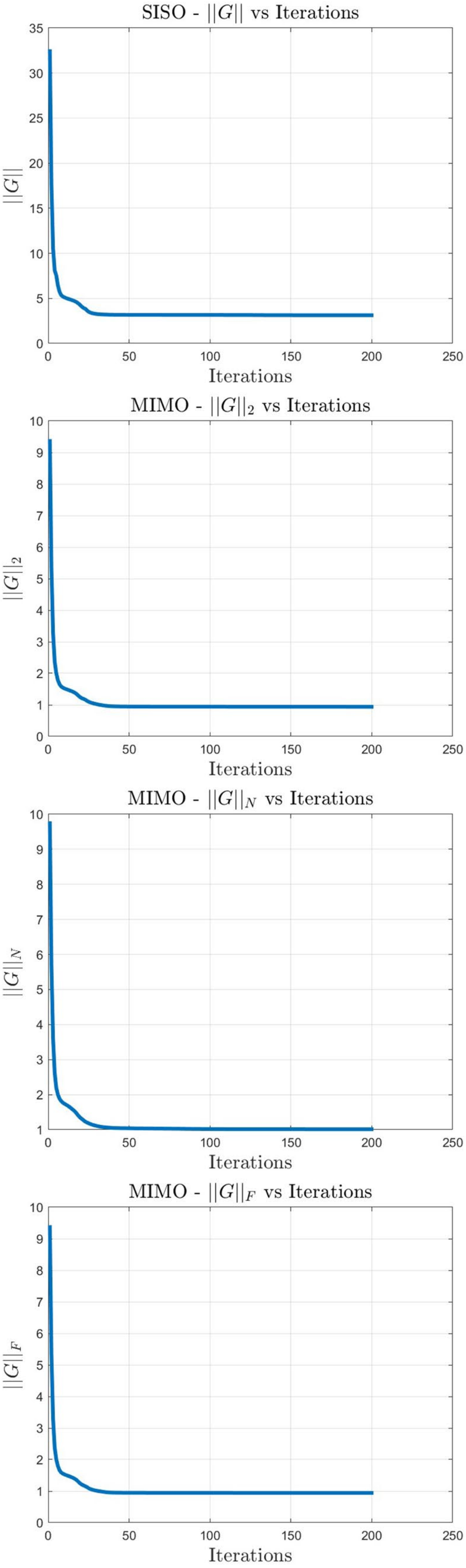

Fig. 14 Convergence curves: in top-down order: SISO, MIMO (2-norm), MIMO (Nuclear norm), and MIMO (Frobenius norm)

6.4.2 General data

Geometry and loads are the same as Problem 1. As for the Young modulus $E_0 = 1000$ and $E_{\min} = 10^{-9}$ are used, whereas the mass density parameters are chosen as $\rho_0 = 10^{-3}$ and $\rho_{\min} = 10^{-6}$ and a Rayleigh damping matrix is introduced that reads

$$\boldsymbol{D} = 0.04\boldsymbol{M} + 0.03\boldsymbol{K}. \tag{53}$$

The number of elements in vertical and horizontal directions are given as 2^5 and 3×2^5, respectively.

6.4.3 H_∞-norm formulation

The optimal topology one gets by minimizing the H_∞-norm of the MIMO transfer matrix $\boldsymbol{G}(i\omega)$ is shown in Fig. 16, whereas Fig. 17 shows the path of the MMA scheme toward convergence. The chattering pattern displayed by the convergence curve is not due to numerical difficulties encountered along the path by the optimization routine. It is conversely due to the fact that, as explained below in more detail, the first singular value $\sigma_1(\omega)$ exhibits two very close points that are competing to realize the overall maximum, i.e., the H_∞ -norm. The design switches back and forth between the two points along an overall converging path. Eventually the two designs coalesce and convergence is achieved.

Figures 18 and 19 shade some light on the complexity of the dynamic response of a Multi-Input Multi-Output system (even the simplest of such type, i.e., a 2-input 2-output structure as the one under investigation) and allow a deeper and rationale understanding of the overall optimization process. Remarkable features emerging from Fig. 18 are as follows:

- Toward the evaluation of $||\boldsymbol{G}(i\omega)||_\infty$, i.e., $\sup_\omega \sigma_1(\omega)$, a competition between two frequencies takes place to assess the one at which the norm itself is attained. This should be considered a general outcome of minimizing the H_∞-norm. In fact such an approach is focused on minimizing the supremum of σ_1 (the maximum of the first singular value) over the frequency axis. Design changes are therefore sharply localized in the frequency range where the current maximum is being attained, whereas no care is taken of the dynamic response at frequencies far apart at which the magnitude of the response is allowed to increase. When such an increase leads to a response amplitude that falls closer and closer to the one where the H_∞-norm is currently attained the competition starts. In general, one can conclude that designs based on the minimization of the H_∞-norm are likely to be multimodal ones as a result of this inherent feature;

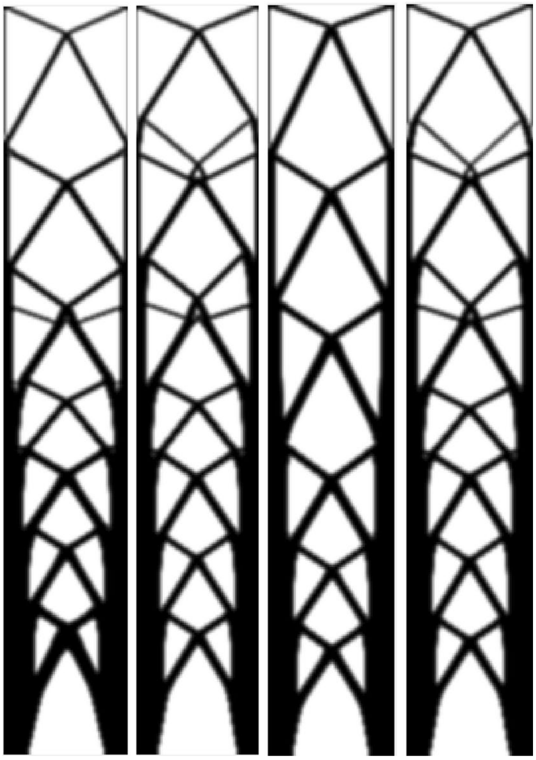

Fig. 15 Optimal topologies: left to right: SISO, MIMO (2-norm), MIMO (Nuclear norm), and MIMO (Frobenius norm)

Table 7 Norm competition between the three available designs

	$\|\|\mathscr{C}\|\|_2$	$\|\|\mathscr{C}\|\|_N$	$\|\|\mathscr{C}\|\|_F$
2-norm design	**0.909**	1.055	0.943
Nuclear norm design	0.926	**1.013**	0.940
Frobenius norm design	0.939	1.048	**0.921**

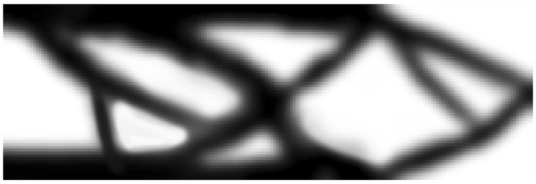

Fig. 16 min H_∞-norm formulation - Optimal topology

– it is well known that classical eigenvalue optimization may be affected by the eigenvalue crossing phenomenon, i.e., a crossing between different eigenvalue trajectories in the design space that leads to the annoying non-differentiability of the min – max eigenvalue problem. A similar behavior shows up in Fig. 18 where a singular value crossing takes place at 4.0 rad/s. Such crossing takes place at a minimum point so that the proposed procedure is not affected by the phenomenon. However this may not always be the case and an issue may arise in general so that further investigations in the near future are planned on this topic.

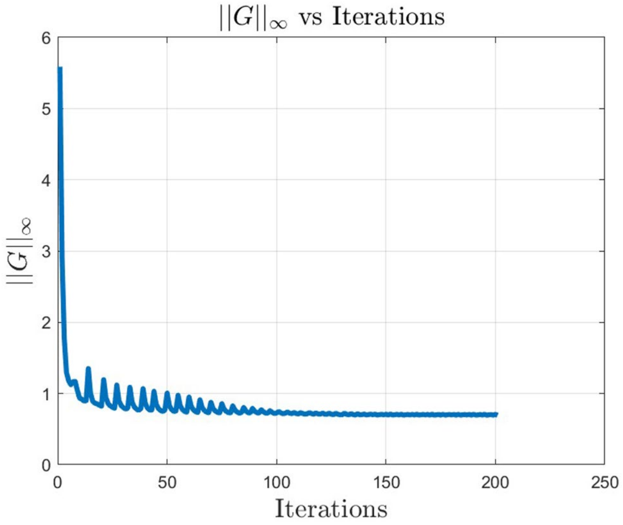

Fig. 17 MIMO formulation - H_∞-norm convergence curve

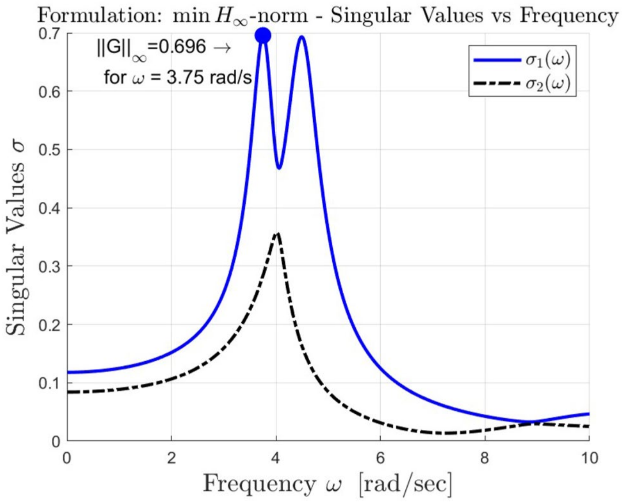

Fig. 18 H_∞-norm formulation - Variation of the two singular values $\sigma_1(\omega)$ and $\sigma_2(\omega)$

Table 8 H_∞-norm formulation. Key points of the frequency response amplitude plot (see Fig. 19)

Point	Frequency [rad/s]	Amplitude
P_{11}	3.76	0.645
P_{12}	3.77	0.152
P_{21}	3.77	0.152
P_{22}	4.49	0.689

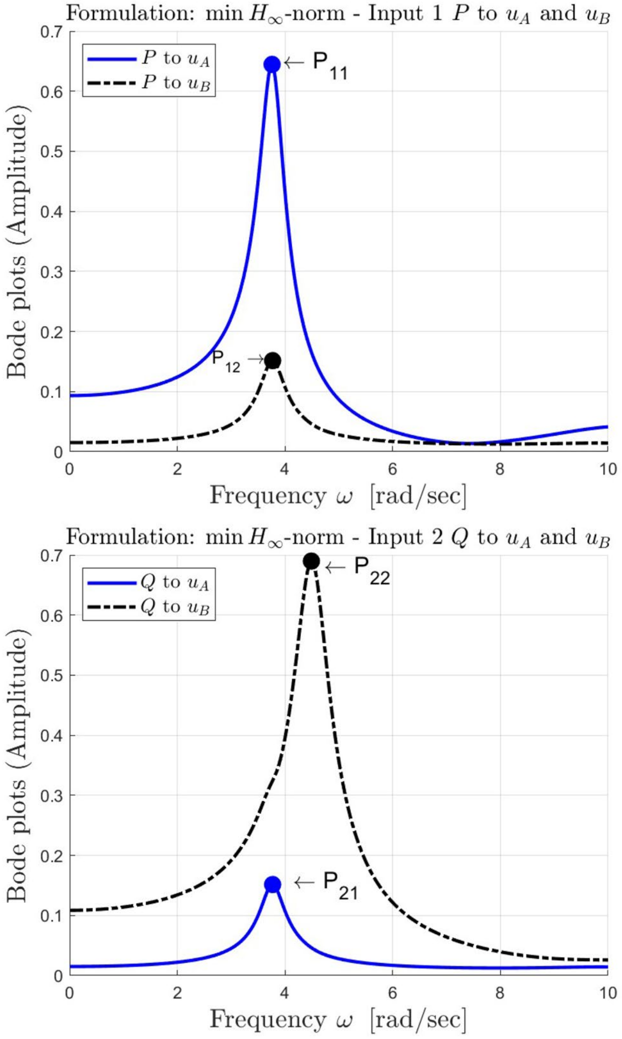

Fig. 19 H_∞-norm formulation - Frequency response amplitudes—Top: from P (input 1) to outputs 1 (u_A) and 2 (u_B)—Bottom: from Q (input 2) to outputs 1 (u_A) and 2 (u_B)

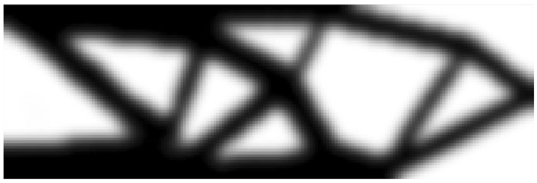

Fig. 20 min Nuclear norm formulation—Optimal topology

Switching to the amplitude of the frequency response, reference is made to Fig. 19 where the points in Table 8 are highlighted. First of all one should observe that the two points P_{12} and P_{21} govern the mutual compliance, i.e., the (amplitude of the) displacement of B when the loaded point is A and viceversa. The fact that $P_{12} = P_{21}$ is a version of the Maxwell–Betti reciprocal work theorem applied to the dynamic case in a harmonic load framework. Interestingly enough, not only are the two amplitudes the same, but also the frequencies at which maximum points are attained coincide. As for the overall system dynamics, the singular value crossing at $\omega = 4$ rad/s and the competition between the two maximum points find their motivation and confirmation in Fig. 19 where one may see that the amplitude of the compliance of the two loads (points P_{11} and P_{22}) is quite similar as are the relevant frequencies $\omega_{P_{11}} = 3.76$ rad/s and $\omega_{P_{22}} = 4.49$ rad/s. From a quantitative point of view, the singular value decomposition of $\boldsymbol{G}(i\omega)$ for all the frequencies above read as follows:

1. $\omega = \omega_{P_{11}} = 3.76$ rad/s

$$\boldsymbol{U}_{11} = \begin{bmatrix} -0.058 + 0.940i & -0.198 + 0.271i \\ 0.119 - 0.313i & -0.7556 + 0.563i \end{bmatrix} \tag{54}$$

$$\boldsymbol{\Sigma}_{11} = \begin{bmatrix} 0.695 & 0 \\ 0 & 0.286 \end{bmatrix} \tag{55}$$

$$\boldsymbol{V}_{11} = \begin{bmatrix} -0.942 + 0.000i & -0.335 + 0.000i \\ 0.320 - 0.099i & -0.900 + 0.278i \end{bmatrix} \tag{56}$$

2. $\omega = \omega_{P_{22}} = 4.49$ rad/s

$$\boldsymbol{U}_{22} = \begin{bmatrix} 0.059 - 0.045i & 0.914 + 0.391i \\ -0.878 - 0.473i & 0.001 + 0.074i \end{bmatrix} \tag{57}$$

$$\boldsymbol{\Sigma}_{22} = \begin{bmatrix} 0.6932 & 0 \\ 0 & 0.165 \end{bmatrix} \tag{58}$$

$$\boldsymbol{V}_{22} = \begin{bmatrix} -0.175 + 0.000i & 0.985 + 0.000i \\ 0.932 - 0.319i & 0.166 - 0.057i \end{bmatrix} \tag{59}$$

3. $\omega = \omega_{P_{12}} = \omega = \omega_{P_{21}} = 3.77$ rad/s.

$$\boldsymbol{U}_{12} = \begin{bmatrix} -0.017 + 0.942i & -0.190 + 0.276i \\ 0.109 - 0.317i & -0.747 + 0.5748i \end{bmatrix} , \tag{60}$$

$$\boldsymbol{\Sigma}_{12} = \begin{bmatrix} 0.694 & 0 \\ 0 & 0.289 \end{bmatrix} , \tag{61}$$

$$\boldsymbol{V}_{12} = \begin{bmatrix} -0.942 + 0.000i & -0.335 + 0.000i \\ 0.319 - 0.103i & -0.897 + 0.289i \end{bmatrix} . \tag{62}$$

6.4.4 Nuclear norm formulation

The optimal topology one gets by minimizing the Nuclear norm of the MIMO transfer function $\boldsymbol{G}(i\omega)$ and the

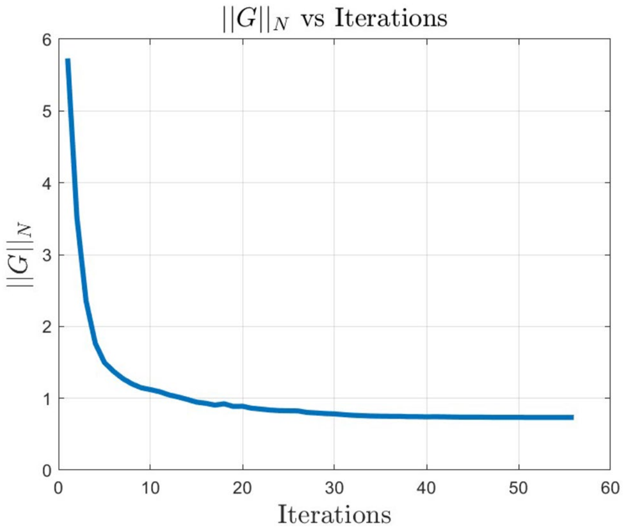

Fig. 21 MIMO formulation— Nuclear norm convergence curve

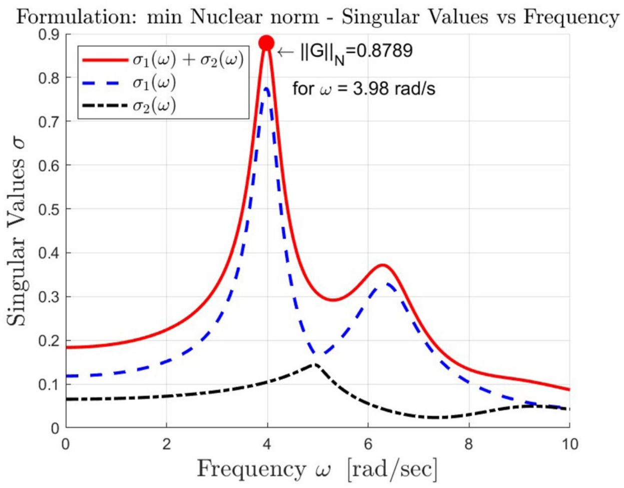

Fig. 22 Nuclear norm formulation - Variation of the two singular values $\sigma_1(\omega)$ and $\sigma_2(\omega)$

convergence curve are shown in Figs. 20 and 21, respectively. In Fig. 21 no oscillations show up and this is a first indicator of a structural response dominated by a single mode. This is confirmed by the singular value curves displayed in Fig. 22 wherein an innocuous low-amplitude crossing takes place at about $\omega = 5$ rad/s. In fact the second gain (maximum of the second singular value) is smaller that half the larger gain that takes place at $\omega = 3.98$ rad/s that therefore governs the entire system dynamics.

Figure 22 shows the variations of the singular values $\sigma_1(\omega)$ and $\sigma_2(\omega)$ and their sum $\sigma_1(\omega) + \sigma_2(\omega)$ with respect to the frequency ω.

Outstanding features emerging from Fig. 22 are as follows:

- the Nuclear norm, i.e., the maximum value of $\sigma_1(\omega) + \sigma_2(\omega)$ is 0.8789 and is attained at $\omega = 3.98$. Using a goal function that depends on both the singular values causes design updates to spread their effect over a wider frequency range, at least as wide as $(\omega_{P_{11}}, \omega_{P_{22}})$, see also Table 9. This seems to be the main reason why the amplitude of the second singular value is kept small and a single-mode design is ended up with, unlike the H_∞ -norm-based design that was a multi-modal one;
- Not only is the dynamic response dominated by a single mode but also a similar behavior is exhibited by the two inputs. Upon comparing the upper and lower amplitude curves in Fig. 23, one concludes that the plant is dominated by the first input, i.e., P, whereas the second input Q has a nearly negligible effect.

To summarize with, (at least for the problem at hand) using the nuclear norm leads to a single-mode design wherein one load determines much of the structural response, whereas the other is nearly annihilated by the design itself. These features are not shared by the H_∞ -norm-based design that puts all its efforts against the overall peak response.

The single modality of the structural response is further confirmed by the singular value decomposition of $\boldsymbol{G}$ for $\omega = \omega_{P_{11}}$, $\omega = \omega_{P_{22}}$, and $\omega = \omega_{P_{12}}$. In all cases, the second (and last) gain σ_2 is much smaller than the largest gain σ_1 as shown below:

1. $\omega = \omega_{P_{11}} = 3.97$ rad/s

$$\boldsymbol{U}_{11} = \begin{bmatrix} -0.104 + 0.837i & -0.519 + 0.140i \\ 0.091 - 0.5300i & -0.824 + 0.182i \end{bmatrix} , \tag{63}$$

$$\boldsymbol{\Sigma}_{11} = \begin{bmatrix} 0.645 & 0 \\ 0 & 0.087 \end{bmatrix} , \tag{64}$$

$$\boldsymbol{V}_{11} = \begin{bmatrix} -0.843 + 0.000i & -0.537 + 0.000i \\ 0.537 - 0.025i & -0.843 + 0.040i \end{bmatrix} , \tag{65}$$

2. $\omega = \omega_{P_{22}} = 4.01$ rad/s

Table 9 Nuclear norm formulation. Key points of the frequency response amplitude plot (see Fig. 23)

Point	Frequency [rad/s]	Amplitude
P_N	3.98	0.879
P_{11}	3.97	0.563
P_{12}	4.01	0.339
P_{21}	4.01	0.339
P_{22}	6.36	0.301

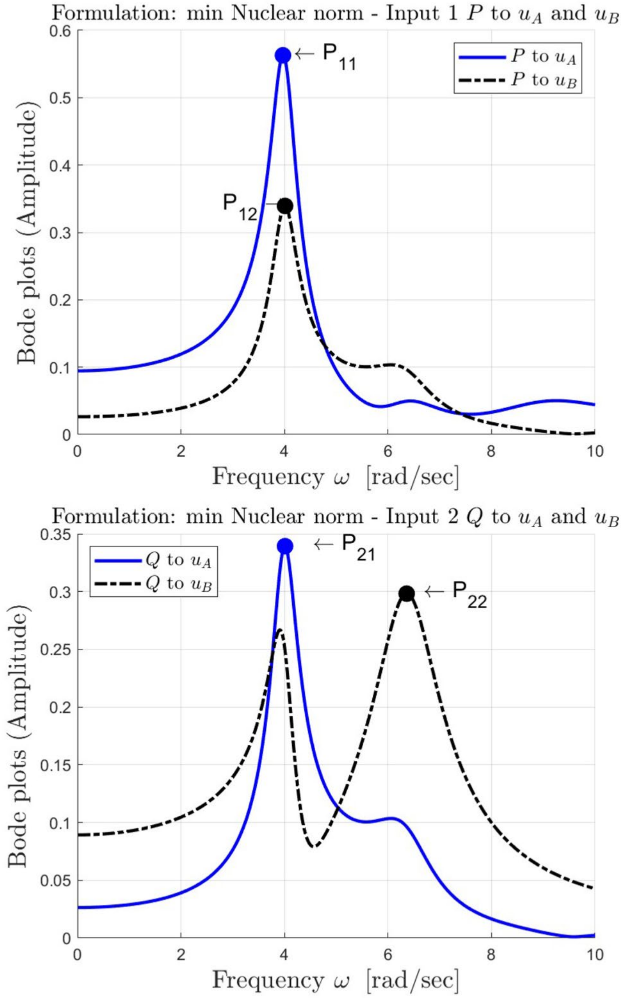

Fig. 23 Nuclear norm formulation - Frequency response amplitudes—Top: from P (input 1) to outputs 1 (u_A) and 2 (u_B)—Bottom: from Q (input 2) to outputs 1 (u_A) and 2 (u_B)

$$\boldsymbol{U}_{22} = \begin{bmatrix} -0.130 + 0.279i & 0.839 + 0.449i \\ -0.217 + 0.927i & -0.295 - 0.087i \end{bmatrix}, \tag{66}$$

$$\boldsymbol{\Sigma}_{22} = \begin{bmatrix} 0.274 & 0 \\ 0 & 0.034 \end{bmatrix}, \tag{67}$$

$$\boldsymbol{V}_{22} = \begin{bmatrix} -0.307 + 0.000i & -0.952 + 0.000i \\ -0.931 - 0.195i & 0.301 + 0.063i \end{bmatrix}, \tag{68}$$

3. $\omega = \omega_{P_{12}} = \omega = \omega_{P_{21}} = 3.97,$ rad/s.

$$\boldsymbol{U}_{12} = \begin{bmatrix} 0.026 + 0.845i & -0.515 + 0.144i \\ 0.010 - 0.535i & -0.824 + 0.187i \end{bmatrix}, \tag{69}$$

$$\boldsymbol{\Sigma}_{12} = \begin{bmatrix} 0.641 & 0 \\ 0 & 0.088 \end{bmatrix}, \tag{70}$$

$$\boldsymbol{V}_{12} = \begin{bmatrix} -0.845 + 0.000i & -0.535 + 0.000i \\ 0.534 - 0.027i & -0.844 + 0.043i \end{bmatrix}. \tag{71}$$

7 Conclusions and need for further investigations

A novel unifying approach has been presented that allows to handle static and dynamic topology optimization using the very same formulation (and computer code). The singular value decomposition of the input–output transfer matrix $\boldsymbol{G}$ along with the selection of suitable matrix norms (defined uniquely in terms of its singular values) to be used as goal functions represent the key ingredients of the proposed approach wherein static and dynamic frameworks are identical. Computing $\boldsymbol{G}$, its norms and its sensitivities in the static and dynamic cases are the main focus of the paper that is completed by in-depth numerical investigations wherein general features of the resulting designs are highlighted. The full Matlab code that solves the static problem is provided in online Appendix 1 and may be readily extended to cover the dynamic case by properly implementing the details given in Sect. 5.

Ongoing investigations are concerned with the following topics that are mainly focused on the improvement of the dynamic topology optimization approach:

- The number of states of the dynamic systems investigated herein is about twenty thousands, whereas just few poles govern the dynamic response of the system. Using reduced order models seems therefore the more natural way to cope with dynamic accuracy and reduction of CPU time;
- Using reduced order models opens the way to the potential issue of computing the sensitivities. In fact reduced order models are in general black boxes that do not allow design variables at the element level to hold their physical meaning. A possible way out is the adoption of parametric reduced order models that are currently being investigated;
- The H_∞-norm approach was shown to lead to multimodal solutions characterized by a competition between candidate design points at different frequencies. The numerical minimization scheme is therefore likely to be trapped in between the two points exhibiting an overall chattering behavior. The adoption of proper filters that allow a fast choice among the two points is the remedy being exploited to solve such problem;

- There are such applications as, for example, building engineering wherein structures should be designed to withstand static and dynamic load combinations at the same time. The former are often determined by permanent and variable vertical loads, whereas earthquakes represent the driving loads of the latter. It is therefore planned to extend the proposed approach using goal functions that are convex combinations of the purely static and dynamic ones;
- Finally a few words of the robustness issue (both in statics and dynamics). It is planned to adopt some non-probabilistic uncertainty models similar to what is done in the framework of robust control of uncertain systems. By doing so, one operates on a class of systems (and not on a single one) in a worst-case scenario framework;

To conclude with, it is planned to use some Application Programming Interface (API) to link the Matlab code to a general purpose finite element software so as to apply the proposed approach to (nearly) real-world structural and industrial applications.

Appendix A Matlab code

A.1 Brief explanation of the code

Other than the method of moving asymptotes for which reference is made to Svanberg (1987), the code consists of four files that are briefly commented on below. In principle, only lines where the user is supposed to provide an input (via a specific choice of a proper parameter) are commented. The code itself is thoroughly commented to ease the use and the overall understanding of the reader. Furthermore, data structures and the like are based on (Andreassen et al. 2011) to which the reader and the user are addressed for further details.

A.1.1 main

- Lines 15-18: `ITYPE`: a single line should be left uncommented to select the problem under investigation (SISO, MISO, SIMO, MIMO);
- Lines 20-22: `INORM`: a single line should be left uncommented to select the matrix norm to be used (2, Nuclear, Frobenius).

A.1.2 initialize_mma

This file and the following one are standard files that are needed to run the MMA Svanberg (1987). Parameters to be set by the user are as follows:

- Line 25: `maxoutit`: maximum number of iterations of the MMA;
- Line 26: `kkttol`: tolerance to exit the MMA.

A.1.3 start_mma

This file and the previous one are standard files that are needed to run the MMA Svanberg (1987). The file calls the function `compliance_stat` that returns goal function, constraints and relevant gradients. A verbose mode is set (but can be changed by the user) that at each iteration provides the following:

- Line 32: current values of iteration, goal function and volume constraint;
- Line 33: the current topology is plotted.

A.1.4 compliance_stat

This function is the core of the proposed approach. No action is expected to be taken by the user. However, a few comments are given below (in addition to the ones reported in the file itself).

- Lines 19-31: matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ (respectively named `Bstate` and `Cstate` in the code) are computed once for all at the first iteration and boundary conditions applied;
- Lines 33-35: the stiffness matrix is updated;
- Lines 37-39: the factors $\boldsymbol{C}\boldsymbol{K}^{-1}$ and $\boldsymbol{K}^{-1}\boldsymbol{B}$ in Equation (16) are computed;
- Lines 41-49: the transfer matrix $\boldsymbol{G}$ in Equation (11) is determined, its Singular Value Decomposition computed, Equation (2), and relevant singular values sorted in descending order (and consequently the associated left and right singular vectors);
- Lines 51-53: the factors $\boldsymbol{u}_h^H\boldsymbol{C}\boldsymbol{K}^{-1}$ and $\boldsymbol{K}^{-1}\boldsymbol{B}\boldsymbol{v}_h$ are computed as needed by plugging Equation (16) into Equation (17);
- Lines 55-64: the goal function is computed that is the norm $||\boldsymbol{G}||$ selected by the user via `INORM`;
- Lines 66-87: the gradient of the goal function is computed by plugging into the chain rule in Equation (18) the formulas in Equations (19) and (17). One should notice that the `parfor` command (parallel for) available in Matlab is used to speed up the computations;
- Lines 89-end: volume constraints are computed and relevant filters applied (see again (Andreassen et al. 2011)).

```
%% MAIN PROGRAM

clear all
close all
clc

global ITYPE INORM

%% ITYPE GOVERNS THE SYSTEM TYPE (SEE BELOW)
%% INORM GOVERNS THE NORM TYPE (SEE BELOW)

tic

%% PROBLEM TYPE
%ITYPE = 1; %SISO
%ITYPE = 2; %MISO
%ITYPE = 3; %SIMO
ITYPE = 4;  %MIMO

%% NORM TYPE
%INORM = 1; %2 NORM
INORM = 2; %NUCLEAR NORM
%INORM = 3; %FROBENIUS NORM

%% MAIN PARAMETERS
nely = 2^6;       %number of elements in x direction
nelx = 3*nely;    %number of elements in y direction
volfrac = 0.5;    %volume fraction available for material
penal = 3.0;      %SIMP penalization factor
rmin = 3.;        %filter distance
ft = 2;           %filter type (1=sensitivity, 2=density)

%% OTHER PARAMETERS
LX = nelx;              %x-size-global
LY = nely;              %y-size-global
NUMEL = nelx*nely;      %number of elements
dX = LX/nelx;           %x-size-element
dY = LY/nely;           %y-size-element
area = dX*dY;           %element area
V0 = LX*LY;             %maximum volume (thickness=1 implicitly)

%% MATERIAL PROPERTIES
E0 = 1;                 %Nominal Young modulus
Emin = 1e-9;                %Minimum (void) Young modulus
nu = 0.3;               %Poisson ratio (constant)

%% PREPARE FINITE ELEMENT ANALYSIS
A11 = [12  3 -6 -3;  3 12  3  0; -6  3 12 -3; -3  0 -3 12];
A12 = [-6 -3  0  3; -3 -6 -3 -6;  0 -3 -6  3;  3 -6  3 -6];
B11 = [-4  3 -2  9;  3 -4 -9  4; -2 -9 -4 -3;  9  4 -3 -4];
B12 = [ 2 -3  4 -9; -3  2  9 -2;  4  9  2  3; -9 -2  3  2];
KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);

%% TOWARD THE GLOBAL STIFFNESS MATRIX
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);

%% DEFINE LOADS AND SUPPORTS (EXAMPLE 1)
if ITYPE == 1
    F = sparse([2 2*nelx*(nely+1)+nely+2]',[1 1]',[-1 -1]',2*(nely+1)*(nelx+1),1);
elseif ITYPE == 2
    F = sparse([2 2*nelx*(nely+1)+nely+2]',[1 2]',[-1 -1]',2*(nely+1)*(nelx+1),2);
```

```
elseif ITYPE == 3
    F = sparse([2 2*nelx*(nely+1)+nely+2]',[1 1]',[-1 -1]',2*(nely+1)*(nelx+1),1);
elseif ITYPE == 4
    F = sparse([2 2*nelx*(nely+1)+nely+2]',[1 2]',[-1 -1]',2*(nely+1)*(nelx+1),2);
else
    'System type unknown'
    exit
end
na = (nely+1)*(2*nely+1);
fixeddofs = sort(union(1:2:2*(nely+1),[2*na-2*(nely+1) 2*na 2*na+2*(nely+1)]));

% if the mesh is fine, use more cosntrained elements
if nely > 2^3
    fixeddofs = sort(unique(union(fixeddofs,[2*na-4*(nely+1) 2*na+4*(nely+1)])));
end
alldofs = 1:2*(nely+1)*(nelx+1);
freedofs  = setdiff(alldofs,fixeddofs);

%% PREPARE FILTER
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx
  for j1 = 1:nely
    e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
      for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
      end
    end
  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);

%% INITIALIZE 1: UNIFORM INITIAL GUESS (but could be the last design from a previous run or else)
x = repmat(volfrac,nely,nelx);
xPhys = x;

%% INITIALIZE 2: MMA PARAMETERS
initialize_mma;

%% START MMA AND GET THE OPTIMAL DESIGN
start_mma;

toc
```

```
%% initialize_mma.m
%% see Svanberg's note for the meaning of the following parameters

global outeriter

m = 2;                  %number of constraints
n = NUMEL;              %number of design variables
epsimin = 1e-6;
eeen     = ones(n,1);
eeem     = ones(m,1);
zeron    = zeros(n,1);
zerom    = zeros(m,1);
xval     = x(:);
xold1    = xval;
xold2    = xval;
xmin     = 10^(-6)*ones(size(xval));
xmax     = ones(size(xval));
low      = xmin;
upp      = xmax;
c        = 10^4*eeem;
d        = zerom;
```

```
a0       = 1;
a        = zerom;
outeriter = 0;
maxoutit  = 200;
kkttol   = 5*10^(-4);
```

```
%% start_mma.m
%% ITERATION "0"
if outeriter < 0.5
    [f0val,df0dx,fval,dfdx,xPhys] = compliance_stat(xval,H,Hs,E0,Emin,nelx,nely,...
        penal,ft,V0,area,volfrac,KE,iK,jK,F,freedofs,edofMat);
    xsave = xval;
    nsave = f0val;
    goalsave = f0val;
end

%% ITERATIONS START
kktnorm = kkttol+10;
outit = 0;
while kktnorm > kkttol & outit < maxoutit
    outit   = outit+1;
    outeriter = outeriter+1;
    %% The MMA subproblem is solved at the point xval:
    [xmma,ymma,zmma,lam,xsi,eta,mu,zet,s,low,upp] = ...
        mmasub(m,n,outeriter,xval,xmin,xmax,xold1,xold2, ...
        f0val,df0dx,fval,dfdx,low,upp,a0,a,c,d);
    %% Some vectors are updated:
    xold2 = xold1;
    xold1 = xval;
    xval = xmma;
    [f0val,df0dx,fval,dfdx,xPhys] = compliance_stat(xval,H,Hs,E0,Emin,nelx,nely,...
        penal,ft,V0,area,volfrac,KE,iK,jK,F,freedofs,edofMat);
    %% The residual vector of the KKT conditions is calculated:
    [residu,kktnorm,residumax] = ...
        kktcheck(m,n,xmma,ymma,zmma,lam,xsi,eta,mu,zet,s, ...
        xmin,xmax,df0dx,fval,dfdx,a0,a,c,d);
    %% print results
    fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f\n',outit,f0val,mean(xval(:)));
    xPhysPrint = reshape(xPhys,nely,nelx);
    %% plot densities
    colormap(gray); imagesc(1-xPhysPrint); caxis([0 1]); axis equal; axis off; drawnow;
end
```

```
function [c,dcc,v,dvv,xPhys] = compliance_stat(x,H,Hs,E0,Emin,nelx,nely,penal,ft,V0,area,volfrac,...
    KE,iK,jK,F,freedofs,edofMat)

global outeriter Bstate Cstate ITYPE INORM
%  c     = objective function
% dc     = objective function gradient
%  v     = volume (constraint)
% dv     = volume gradient

%% filtering densities
if ft == 1
    xPhys = x(:);
    xPhys = reshape(xPhys,nely,nelx);
elseif ft == 2
    xPhys(:) = (H*x(:))./Hs;
    xPhys = reshape(xPhys,nely,nelx);
end

%% matrices to be computed only at the first iteration
if outeriter < 0.5
    Bstate = F;
    if ITYPE == 1 || ITYPE == 4
        Cstate = F';
    elseif ITYPE == 2
        Cstate = [sparse([2 2*nelx*(nely+1)+nely+2]',[1 1]',[-1 -1]',2*(nely+1)*(nelx+1),1)]';
    elseif ITYPE == 3
        Cstate = [sparse([2 2*nelx*(nely+1)+nely+2]',[1 2]',[-1 -1]',2*(nely+1)*(nelx+1),2)]';
    end
    Bstate = Bstate(freedofs,:);
    Cstate = Cstate(:,freedofs);
end

%% updated stiffness matrix
sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),64*nelx*nely,1);
K = sparse(iK,jK,sK); K = (K+K')/2;

%% factors for sensitivities 1
FSX = Cstate/K(freedofs,freedofs);
FDX = K(freedofs,freedofs)\Bstate;

%% transfer matrix
G = Cstate*FDX;

%% svd
[UU,SS,VV] = svds(G,max(size(G)));
[SS,II] = sort(diag(SS),'descend');
SS = diag(SS);
UU = UU(:,II);
VV = VV(:,II);

%% factors for sensitivities 2
FSX2 = UU'*FSX;
FDX2 = FDX*VV;

%% objective function
if INORM == 1
    c = SS(1,1);
elseif INORM == 2
    c = sum(diag(SS));
elseif INORM == 3
    c = norm(diag(SS));
else
    'wrong norm case'
end

% 'grad'
NormGrad = zeros(nelx*nely,1);

parfor jj = 1:nelx*nely
    %% stiffness matrix derivative
    kDer = sparse(size(K,1),size(K,2));
    kDer(edofMat(jj,:),edofMat(jj,:)) = penal*xPhys(jj)^(penal-1)*(E0-Emin)*KE;
    kDer = kDer(freedofs,freedofs);
    if INORM == 1
        NormGrad(jj) = - real(FSX2(1,:)*kDer*FDX2(:,1));
    elseif INORM == 2
        for kk = 1:length(diag(SS))
            NormGrad(jj) = NormGrad(jj) - real(FSX2(kk,:)*kDer*FDX2(:,kk));
        end
    elseif INORM == 3
        for kk = 1:length(diag(SS))
            NormGrad(jj) = NormGrad(jj) - 2*SS(kk,kk)*real(FSX2(kk,:)*kDer*FDX2(:,kk));
        end
        NormGrad(jj) = NormGrad(jj)/(2*norm(diag(SS)));
    end
end
dc = reshape(NormGrad,nely,nelx);

%% constraints
v1  = sum(xPhys(:))*area/V0-volfrac;
v2  = -sum(xPhys(:))*area/V0 + volfrac - volfrac/30;
v = [v1;v2];
dv1 = ones(nely,nelx)*area/V0;
dv2 = -ones(nely,nelx)*area/V0;

%% filtering/modification of sensitivities
if ft == 1
    dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
elseif ft == 2
    dc(:) = H*(dc(:)./Hs);
    dv1(:) = H*(dv1(:)./Hs);
    dv2(:) = H*(dv2(:)./Hs);
end
dcc = dc(:);
dvv = [dv1(:)';dv2(:)'];
```

Acknowledgements This research is supported in part by the Grant of the Municipality of Pavia No. F739 (Advanced methods for seismic risk index evaluation).

Funding Open access funding provided by Università degli Studi di Pavia within the CRUI-CARE Agreement.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Replication of results The full Matlab code solving the static topology optimization problem is reported in online Appendix 1. The user may straightforwardly extend the static code to cover the dynamic case following the algorithmic and computational indications given in Sect. 5.

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

References

Andreassen Erik, Clausen Anders, Schevenels Mattias, Lazarov Boyan S, Sigmund Ole (2011) Efficient topology optimization in matlab using 88 lines of code. Struct Multidisc Optim 43:1–16

Beghini A, Baker WF, Paulino GH (2012) Topology optimization for braced frames: Combining continuum and beam/column elements. Eng Struct 37:106–124

Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9–10):635–654

Bendsøe MP, Sigmund O (2003) Topology optimization: Theory, methods and applications. Springer, Berlin

Bruggi M, Duysinx P (2012) Topology optimization for minimum weight with compliance and stress constraints. Struct Multidisc Optim 46:369–384

Bruggi M, Venini P (2008) A mixed fem approach to stress-constrained topology optimization. Int. J. Numer. Methods Engrg 73:1693–1714

Bruisma NA, Steinbuch M (1990) A fast algorithm to compute the h_∞ -norm of a transfer function matrix. System and Control Letters 14:287–293

Brunton SL, Kutz JN (2019) Data-Driven Science and Engineering - Machine Learning. Dynamical Systems and Control. Cambrige University Press, Cambridge

Collet M, Bruggi M, Duysinx P (2017) Topology optimization for minimum weight with compliance and simplified nominal stress constraints for fatigue resistance. Struct Multidisc Optim 55(3):7839–855

Emmendoerfter H Jr, Maute K, Fancello EA, Nelli Silva EC (2022) A level set-based optimized design of multi-material compliant mechanisms considering stress constraints. Comput Methods Appl Mech Eng 391:114556

Ferrari F, Sigmund O (2020) A new generation 99 line matlab code for compliance topology optimization and its extension to 3d. Struct Multidisc Optim 62:2211–2228

Ferrari F, Sigmund O (2020) Towards solving large-scale topology optimization problems with buckling constraints at the cost of linear analyses. Comput Methods Appl Mech Eng 363:112911

Francis BA (1987) A Course in H_∞ Control Theory, vol 88. Springer-Verlag, Berlin, Heidelberg

Gersborg-Hansen A, Bendsøe MP, Sigmund O (2006) Topology optimization of heat conduction problems using the finite volume method. Struct Multidisc Optim 31:251–259

Giraldo-Londono O, Paulino GH (2021) Polydyna: a matlab implementation for topology optimization of structures subjected to dynamic loads. Struct Multidisc Optim 64:957–990

Giraldo-Londono O, Russ JB, Aguiló MA, Paulino GH (2022) Limiting the first principal stress in topology optimization: a local and consistent approach. Structural and Multidisciplinary Optimization, **65**

Jensen JS, Nakshatrala PB, Tortorelli DA (2014) On the consistency of adjoint sensitivity analysis for structural optimization of linear dynamic problems. Struct Multidisc Optim 49:831–837

Liu H, Zhang W, Gao T (2015) A comparative study of dynamic analysis methods for structural topology optimization under harmonic force excitations. Struct Multidisc Optim 51:1321–1333

Min S, Kikuchi N, Park YC, Kim S, Chang S (1999) Optimal topology design of structures under dynamic loads. Struct Multidisc Optim 17:208–218

MORLab @ Chair of Automatic Control. sss Toolbox version 2.0. TUM School of Engineering and Design, Technical University of Munich, (2017)

Olhoff N, Du J (2016) Generalized incremental frequency method for topological design of continuum structures for minimum dynamic compliance subject to forced vibration at a prescribed low or high value of the excitation frequency. Struct Multidisc Optim 54:1113–1141

Saak J, Köhler M, Benner P (February 2022) M-M.E.S.S.-2.1 – the matrix equations sparse solvers library. see also: https://www.mpi-magdeburg.mpg.de/projects/mess

Salazar de Troya MA, Tortorelli DA (2018) Adaptive mesh refinement in stress-constrained topology optimization. Struct Multidisc Optim 58:2369–2386

Sigmund O, Maute K (2013) Topology optimization approaches a comparative review. Struct Multidisc Optim 48:1031–1055

Skogestad S, Postlethwaite I (1996) Multivariable Feedback Control - Analysis and Design. John Wiley and Sons, Baffins Lane, Chichester

Strang G (2019) Linear Algebra and Learning from Data. Cambridge Press, Wellesley, Wellesley

Svanberg K (1987) The method of moving asymptotes - a new method for structural optimization. Int J Numer Meth Eng 24:359–373

Venini P, Ceresa P (2018) A rational H_∞-norm-based approach for the optimal design of seismically excited reinforced concrete frames. Earthquake Eng Struct Dynam 47:1522–1543

Yang X, Li Y (2014) Structural topology optimization of dynamic compliance at resonance frequency in thermal environments. Struct Multidisc Optim 49:81–91

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