Computers and Structures 298 (2024) 107362

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

A new rational approach to multi-input multi-output 3D topology optimization

P. Venini

Department of Civil Engineering and Architecture, University of Pavia, via Ferrata, 1, I27100 Pavia, Italy

ARTICLE INFO

Keywords:
Topology optimization
Singular value decomposition (SVD)
Multiple loads
Multiple goals
Matlab

ABSTRACT

A new 3D topology optimization approach is presented that is based on the singular value decomposition of the input/output transfer matrix of the system. To start with, the input and output vectors, i.e. the acting loads and the quantities of interest for the designer, are chosen and the input-output transfer matrix is derived. Such matrix, say $\boldsymbol{G}(\boldsymbol{p})$, depends on the vector of the design variables $\boldsymbol{p}$ (the densities at the element level). The singular value decomposition of $\boldsymbol{G}(\boldsymbol{p})$ is the core of the proposed approach. It provides singular values as well as left and right singular vectors. Singular values are shown to uniquely define a few matrix norms that can be conveniently computed and used as goal functions to be minimized. Left and right singular vectors respectively represent the principal input/output pairs of the system whose gain is the associated singular value. Numerical optimization is pursued via the method of moving asymptotes (MMA) [1] that calls for the semi-analytic computations of objective functions and constraints. The results of a few 3D numerical investigations are presented and discussed in much detail. An in-house Matlab code developed for the sake of this paper, and based on the ones presented in [2] and [3], is provided in full as an Appendix to the paper.

1. Introduction and motivation

Three dimensional topology optimization has received an enormous thrust by the additive manufacturing technology that is enabling the actual realization of systems that were simply not even conceivable just a few decades ago. Within this framework, the availability of a reliable three-dimensional topology optimization approach is mandatory for the success of the entire procedure. By combining the well-established density-based method and the Non-uniform rational basis spline (NURBS) [4] develops a 3D topology optimization method that is applied to bridge-like and cantilevered structures. NURBS-based approaches are also investigated in [5] where a Matlab implementation is also discussed. As far as material design is concerned, [6] presents a 3D approach for the design of auxetic metamaterials along with a few 3D-printed optimal topologies, whereas [7] is focused on implementation details for the design of cellular materials. Tailoring anisotropic three-dimensional metamaterials is the crucial goal pursued in [8] where Nurbs are used to satisfy minimum length scale constraints, whereas [9] presents the solution of 2D and 3D benchmark problems within the framework of stress-based topology optimization. Ultra-light architected materials are presented in [10] where a continuous density distribution is used via the Shepard function in a Nurbs environment, whereas the delicate task of transforming an optimal topology into a beam-like structure is tackled in [11].

When it comes to loaded systems and structures, the concept of input-output transfer matrix has been probably overlooked, or at least not fully exploited, by the mechanical and civil engineering communities. The state equation (that in statics is nearly always an equilibrium equation) typically involves a huge number of variables, e.g. the generalized displacements arising from a finite-element discretization. The designer may not be specifically interested in any of them (or just very few are of actual interest) but their computation is needed to determine the overall state of the system. Given this massive state equation that typically acts as a constraint, there are actually two ways to state the optimization problem. What is typically done is to pursue the minimization of the structural compliance that is the work of the external loads that, thanks to the principle of virtual works, is the same as the internal work. This latter identity is very convenient for updating of the design variables, see e.g. [12]. An alternative approach is proposed hereinafter that mimics a procedure that is typical in such areas as feedback control and automatics. The approach consists of two main steps [13]:

E-mail address: paolo.venini@unipv.it.

https://doi.org/10.1016/j.compstruc.2024.107362
Received 15 November 2023; Accepted 21 March 2024
Available online 9 April 2024

1. an output vector is determined that writes as a linear combination of the state vector entries. The classic compliance of topology optimization may be shown to fall into this category;
2. by so doing and upon ruling out the state vector, a direct input-output transfer matrix is this way determined that depends on the design variables that more often than not are the material densities at the element level.

The input-output transfer matrix is in general a rectangular one (the number of load cases is not the same as the number of the outputs, i.e. the quantities of interest for the designer) that has the appealing feature of governing the direct mapping between external disturbances and desired response. The singular value decomposition (SVD) [14] is then shown to be the ideal tool to extract from the input-output transfer matrix all the info of engineering interest that may drive the overall design process. For the sake of completeness, it should be said that the SVD was previously used in [15] to enhance design sensitivities and identify the components (modes) that contribute the most to the overall structural sensitivity. As shown in the sequel of the paper, left and right singular vectors represent mutually orthogonal basis vectors that map the disturbance onto the structural response. To each singular pair of input and output singular vectors, a singular value is associated that is the gain of the system over that specific channel. Interestingly enough, all the matrix norms of engineering interest are uniquely defined in terms of the relevant singular values. This observation is a key one that defines the optimization approach proposed herein that consists in the minimization of proper norms of the input-output transfer matrix. From an engineering point of view this therefore amounts to finding the design variables that minimize the system gains, i.e. the potentially huge effects of the disturbances on the structural response that should conversely be as small as possible. Peculiar attention is paid to the issue of result replication along the guidelines given in [16], among others. For this reason, the full Matlab code that has originated the numerical examples to follow is given in Appendix A.

The paper is organized as follows. Section 2 presents an input/output formulation of the classical equilibrium equation. Though rather standard in some engineering areas, this format is crucial to arrive at the methodology to be developed hereinafter. A detailed derivation of the input-output transfer matrix is left to Section 3 wherein a Lagrange multiplier technique is introduced to handle non-parallelepiped domains such as the one arising in the framework of exoskeleton structural design. Section 4 addresses the basics of the singular value decomposition whereas reference is made to [14] and [17] for a rigorous presentation of this crucial topic. Alongside a few purely algebraic properties, emphasis is on the engineering significance of the singular value decomposition as a unique tool capable to drive the design toward the minimization of the effects of the loads onto the structure under design. Though applicable to a wide variety of systems and goals, minimizing the structural compliance is the main objective of this paper as is the one of most of the papers in the optimization literature. Section 5 is therefore dedicated to a few definitions of compliance that emerge from the proposed formulation and extend the classical definition of compliance itself. The topology optimization problem is set up in Section 6 where the (non standard) transfer matrix framework is developed in much detail. Semi-analytic formulas for the sensitivities of the transfer matrix are also derived that are needed to compute the sensitivities of the overall goal function, i.e. a suitable transfer-matrix norm. Representative results based on the application of the proposed approach are presented in Section 7. First of all, a variant of the MBB problem on a parallelepiped domain is investigated in much detail. This MIMO problem allows a full understanding of the proposed approach in that two distinct singular values arise naturally along with their respective left and right singular vectors. A few different designs are presented that are optimal with respect to one of the norms previously introduced. The solutions are compared with each other but also with the design one gets using the celebrated paper [2] and the Matlab code therein.

Designing the bracing system of a 3D tall-building is the goal of the second numerical investigation. Solutions of engineering interest are presented along with considerations on which norm and system idealization (single or multi-input and output) should be used to address such kind of problems.

A third numerical example is concerned with the classical L-shaped problem that is interesting *per se*, but also in view of the fact that it allows to draw some conclusions on the effect of the selected filter onto the optimal topology at convergence.

The full Matlab code that solves the 3D topology optimization problem is given in Appendix A.

2. An input-output (re)formulation of the classical equilibrium equation

We shall hereinafter refer to a standard displacement based finite-element discretization of a linear elastic structure. The classical equilibrium equation reads (see e.g. [12] among many more):

$$\boldsymbol{K}\boldsymbol{U} = \boldsymbol{F}, \tag{1}$$

where as usual $\boldsymbol{K}$, $\boldsymbol{U}$ and $\boldsymbol{F}$ are the stiffness matrix and the displacement and load vector, respectively. A few slight but crucial notation modifications and general rearrangements are now needed to arrive at the desired formulation. Besides using $\boldsymbol{x}$ (rather than $\boldsymbol{U}$) to denote the $n \times 1$ displacement vector (n being the number of unconstrained generalized displacements of the structure), the following two changes are made to derive the input-output version of the equilibrium equation (1):

1. the load vector at the right-side of Equation (1) is written as the product $\boldsymbol{B}\boldsymbol{f}$, where $\boldsymbol{B}$ is an $n \times n_i$ load matrix (each column of which takes care of an independent load case, n_i being the number of load cases, i.e. the number of inputs to the structure) and $\boldsymbol{f}$ is an $n_i \times 1$ Boolean vector that represents the actual input to the system. By so doing, the equilibrium equation (1) is replaced by the following one:

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{B}\boldsymbol{f}. \tag{2}$$

   Equation (2) should however be complemented by the so called output equation that defines the components of the $n_o \times 1$ output vector $\boldsymbol{z}$, i.e. the quantities that are of interest for the designer.
2. The output equation reads:

$$\boldsymbol{z} = \boldsymbol{C}\boldsymbol{x}, \tag{3}$$

   i.e. the output vector is written as a proper linear combination of the displacement (state) vector $\boldsymbol{x}$.
   As for the dimension and the meaning of $\boldsymbol{C}$, a few considerations are given next, and reference is also made to Section 5 for the specific choice of using the compliance as output of the system. The number of columns of $\boldsymbol{C}$ is the same as the number of rows of the state vector $\boldsymbol{x}$, i.e. n, as needed by the standard matrix-vector product rule in Equation (3). Each row of $\boldsymbol{C}$ represents an objective that the designer aims to control. Frequent choices include the following:
   - a specific displacement of the state vector $\boldsymbol{x}$, e.g. the horizontal displacement of the top floor of a shear-building structure, in which case the relevant row of $\boldsymbol{C}$ is full of zeroes but for a 1 in the position that identifies that specific displacement;
   - the stress and/or strain at a desired location, that may always be written as a linear combination of the components of the state vector $\boldsymbol{x}$;
   - the compliance, that is treated in much detail in Section 5.

   The number of rows of $\boldsymbol{C}$ is therefore the same as the number of goals of the design and each row takes care of a specific objective. One further note should be considered: when the rows of $\boldsymbol{C}$ share the same physical dimensions, e.g. all displacements, no further

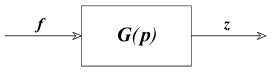

Fig. 1. Block representation of the input-output mapping.

care has to be taken. When physical dimensions do not coincide, e.g. when both displacements and stresses are considered, a normalization of the components of the output vector $\boldsymbol{z}$ is mandatory, this being the only way to make the outputs comparable.

Eventually, Equations (2) and (3) are coupled so as to rule out the displacement vector $\boldsymbol{x}$ and get a direct input($\boldsymbol{f}$) - output($\boldsymbol{z}$) mapping that reads

$$\boldsymbol{z} = \boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{B}\boldsymbol{f}, \tag{4}$$

that defines the input-output transfer matrix $\boldsymbol{G}$ as

$$\boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{C}\boldsymbol{K}^{-1}(\boldsymbol{p})\boldsymbol{B}, \tag{5}$$

where the vector $\boldsymbol{p}$ of the design variables has been introduced on which the stiffness matrix $\boldsymbol{K}$ and consequently the input-output transfer matrix $\boldsymbol{G}$ depend. Vector $\boldsymbol{p}$ typically groups the element densities within a SIMP [18] interpolation scheme. A typical graphical representation of Equation (4) is given in Fig. 1. It is clear, actually by the definition itself, that the input-output transfer matrix $\boldsymbol{G}$ governs the overall response of the structure to be designed. That said, the remainder of the paper introduces an innovative approach that shapes the 3D transfer matrix $\boldsymbol{G}$ so as to determine a structure that fulfills the multiple expectations of the designer. Such an approach relies on the following main concepts:

1. suitable norms of $\boldsymbol{G}$ are to be determined such that their minimization represents a proper mathematical translation of the engineering objectives that the designer is pursuing;
2. the singular value decomposition of $\boldsymbol{G}$ is the main ingredient here for a few reasons:
   - the singular values of $\boldsymbol{G}$ have a clear engineering meaning in terms of structural response that goes far beyond their algebraic definition;
   - the singular values of $\boldsymbol{G}$ uniquely define all the needed norms that allow to grade all possible designs and choose the optimal one;
   - the left and right singular vectors of $\boldsymbol{G}$, far from being purely algebraic quantities, are shown to govern and determine the input-output principal plant directions, each of which is shown to be associated to a gain (the so called blow-up factor using Strang's terminology, [14]) that is the associated singular value.

In what follows a brief recall on the Singular Value Decomposition (SVD) of a possibly complex rectangular matrix is given, basically following [14]. Based on this general algebraic formulation, SVD's concepts are rephrased in an engineering, topology optimization-oriented framework. Eventually a few numerical studies are presented that are concerned with 3D applications of classical as well as (to this author's best knowledge) new applications of structural engineering interest.

3. The factors $\boldsymbol{K}$, $\boldsymbol{C}$, and $\boldsymbol{B}$ that define the transfer matrix $\boldsymbol{G}$

The three matrix factors $\boldsymbol{K}$, $\boldsymbol{C}$, and $\boldsymbol{B}$ that define the transfer matrix $\boldsymbol{G}$ (Equation (5)) are described in some detail in this paragraph. Details are given on both the engineering formulation and Matlab implementation as well.

3.1. FEM discretization and the stiffness matrix $\boldsymbol{K}$

As for the stiffness matrix $\boldsymbol{K}$, the eight-node hexahedral isoparametric element is used for the finite-element discretization. Reference

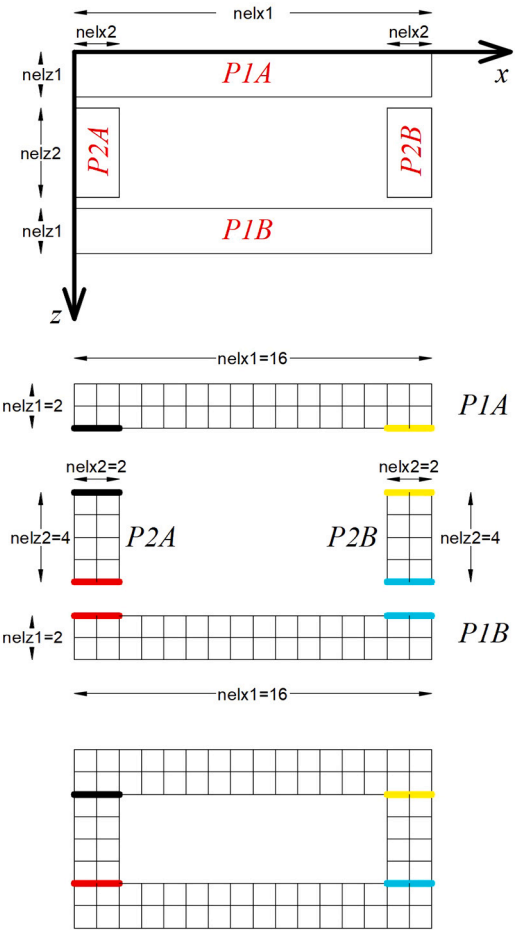

Fig. 2. High-rise building - Plan view of the four walls and Lagrange multipliers.

is respectively made to [19] and [2] for an analysis of the finite element and details concerning its implementation using Matlab. Node and element numbering, dof's management, data structures and plotting functions are borrowed from those presented in [2] (that is turn based on [12]), and extended when needed to fulfill the specific goals of this paper. It seems worth saying that composite (i.e. non parallelepiped) domains are addressed using a standard Lagrange-multiplier technique whereby the equilibrium equation (2) is replaced by its mixed counterpart that reads:

$$\begin{pmatrix} \boldsymbol{K} & \boldsymbol{H} \\ \boldsymbol{H}^T & \boldsymbol{0} \end{pmatrix} \begin{bmatrix} \boldsymbol{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}\boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix}. \tag{6}$$

As shown in the Matlab code in Appendix A, when a sparse storage of the involved matrices is adopted, using such an approach for imposing internal node compatibility does not call for any substantial code modification but for the fact that the dimension of the state vector ($[\boldsymbol{x}\ \lambda]^T$ versus $\boldsymbol{x}$) changes and so does that of the output matrix $\boldsymbol{C}$. In particular, each line of the vector equation $\boldsymbol{H}^t\boldsymbol{x} = \boldsymbol{0}$ is of type $x_k - x_h = 0$, i.e. it is used to restore compatibility between the degrees-of-freedom of adjacent nodes that formally belong to different and mutually orthogonal walls as shown in Fig. 2. The latter figure also shows the meaning of the size parameters `nelx1`, `nelx2`, `nelz1` and `nelz2` that are used in the Matlab code to define the mesh size in the horizontal plane. A further parameter `nely1` = `nely2` = `nely` is used to define the elevation of the high-rise buildings along with `nstoreys` that is the variable retaining the number of storeys (see Fig. 3).

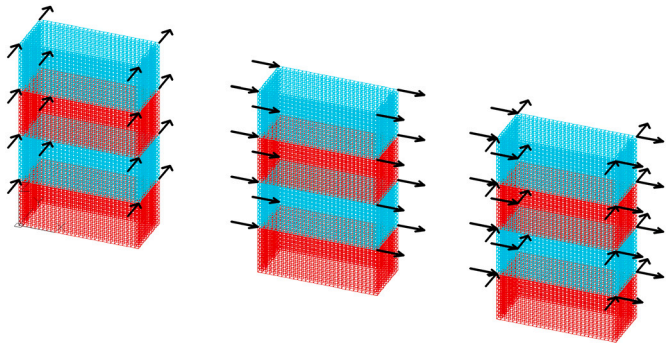

Fig. 3. High-rise building - 3D view, load cases and load combination.

3.2. The vector output equation and matrix $\boldsymbol{C}$

One of the main features of the proposed approach is the possibility to address multiple design objectives. Each goal should be written as a linear combination of the state-vector $\boldsymbol{x}$ thanks to a suitable selection of (each row of) the matrix $\boldsymbol{C}$. In this respect a few considerations seem to be appropriate:

- When the system is actually a Multi-Output one ($n_o > 1$) and the outputs do not share the physical dimensions, e.g. the entries of the output vector $\boldsymbol{z}$ include displacements and stresses, a normalization procedure is compulsory to allow the numerical solver a fair evaluation of all the outputs toward a meaningful minimization of the norm of the transfer matrix $\boldsymbol{G}$. A normalization of the input-output mapping may also be useful to ease the task of the numerical solver as well [1];
- When the goal is the compliance, or better one of its possible variants as discussed in Section 5, the main ingredient is the product of any acting load, say f_k, times its dual displacement x_k so as to define the external work $\mathscr{C}_k = f_k u_k$ that shows how any compliance problem fits the proposed formulation since any compliance is actually the linear combination of the displacement (state) vector wherein the applied loads are the coefficients of the linear combination.

3.3. The system input and matrix $\boldsymbol{B}$

The role played by matrix $\boldsymbol{B}$ in defining the system input has already been outlined in Section 2 and more follows in the paragraph below. It should however be further recalled the importance of splitting the load vector $\boldsymbol{F}$ into the product of two factors $\boldsymbol{Bf}$ which is quite a standard procedure in such areas as automatics but not so in civil and mechanical engineering. As a matter of fact, the vector $\boldsymbol{f}$ is the actual input to the system. It has as many rows as there are load cases (n_i) and its size has nothing to do with the system size, as it should be for an input vector. On the other side, the left factor $\boldsymbol{B}$ is a matrix that concurs to the definition of the input-output transfer matrix $\boldsymbol{G}$ and as such is part of the structure itself (and in fact the size of $\boldsymbol{B}$ is $n \times n_i$ where n is a defining parameter of the system itself).

4. Singular value decomposition (SVD) and matrix norms

SVD is a powerful algebraic tool that applies to any (generally) rectangular complex matrix with no further requirement. It consists in a three-factor decomposition the details of which are discussed shortly after. From an engineering standpoint, the physical meaning of the three matrix factors depends on the specific problem at hand. When the SVD is applied to a transfer matrix that governs the input-output mapping between applied loads and structural response, the three factors assume a clear physical interpretation that is highlighted next. One should furthermore observe that the singular values of a matrix define uniquely three matrix norms that are shown to be suitable goal functions to be minimized in topology optimization. Also, computing the sensitivity of such norms is, at least as long as a static behavior is considered, a relatively straightforward and computationally cheap task.

4.1. SVD definition

Let n_i and n_o respectively denote the number of inputs and outputs of a Multi-Input Multi-Output (MIMO) system of given transfer matrix $\boldsymbol{G}$ (in the static regime we may assume $\boldsymbol{G}$ as a real matrix). The singular value decomposition of the $n_o \times n_i$ matrix $\boldsymbol{G}$ then writes:

$$\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{7}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary $n_o \times n_o$ and $n_i \times n_i$ matrices, respectively, i.e. such that $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$ and $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$, $\boldsymbol{\Sigma}$ is an $n_o \times n_i$ matrix that reads

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 \\ \boldsymbol{0} \end{bmatrix}, \text{ if } n_o \geq n_i \text{ or}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{0} \end{bmatrix}, \text{ if } n_o \leq n_i,$$

in which $\boldsymbol{\Sigma}_1 = \text{diag}\left\{\sigma_1, \sigma_2, \ldots \sigma_r\right\}$ is the diagonal matrix of the (all positive) singular values, $r = \min(n_o, n_i)$ is the rank of $\boldsymbol{G}$ and the singular values are sorted in descending order, i.e. $\overline{\sigma} \equiv \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \equiv \underline{\sigma} > 0$.

4.2. Singular values, singular vectors, matrix norms and plant directions

The proposed approach naturally calls for suitable definitions of matrix norms that allow to define the amplitude of the response to any given disturbance in terms of the "size" of the input-output matrix $\boldsymbol{G}$. Leaving the interested reader to [14] and [17] for the details, the two-norm (2), the Nuclear norm (N) and the Frobenius norm (F) are chosen for their engineering significance as well as for the fact that they are uniquely defined by the singular values of $\boldsymbol{G}$. The latter should be considered a crucial property in view of the physical meaning of the singular values as discussed below. As for the definitions of the norms themselves, one may write:

$$\begin{aligned} \text{2-norm}: \quad & ||\boldsymbol{G}||_2 = \sigma_1 \\ \text{N-norm}: \quad & ||\boldsymbol{G}||_N = \sigma_1 + \sigma_2 + \ldots + \sigma_r \\ \text{F-norm}: \quad & ||\boldsymbol{G}||_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots \sigma_r^2} \end{aligned} . \tag{8}$$

The N-norm and the F-norm depend on all the singular values unlike the 2-norm that coincides with the largest singular value σ_1. If one recalls, see [14], that the matrix $\boldsymbol{G}$ may be written as the sum of independent rank-one matrices, i.e.:

$$\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \ldots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T, \tag{9}$$

the 2-norm is seen to take care of (i.e. it allows to minimize) that part of $\boldsymbol{G}$ that is associated with the first triple $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ only, whereas both the Nuclear and the Frobenius norms address and allow to minimize a suitable average of all the input-output channels in which the matrix $\boldsymbol{G}$ is decomposed using Equation (9). To gain an even deeper insight into the physical meaning behind the singular value decomposition, Equation (7) is right-multiplied by $\boldsymbol{V}$ and, thanks to the orthogonality of the $\boldsymbol{v}$-s and the $\boldsymbol{u}$-s, one arrives at the componentwise mapping:

$$\boldsymbol{G}\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k, \quad k = 1, 2, \ldots, r. \tag{10}$$

Since $||\boldsymbol{v}_k||_2 = ||\boldsymbol{u}_k||_2 = 1$, Equation (10) allows to get the full meaning of the singular values. In fact one may write:

$$\sigma_k(\boldsymbol{G}) = ||\boldsymbol{G}\boldsymbol{v}_k||_2 = \frac{||\boldsymbol{G}\boldsymbol{v}_k||_2}{||\boldsymbol{v}_k||_2}, \tag{11}$$

that allows one to conclude that the k-th singular value σ_k is the blow-up factor (gain) along the k-th plant channel governed by the pair of

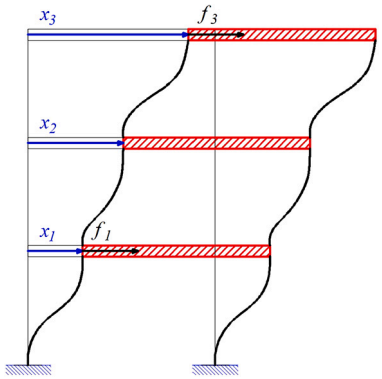

Fig. 4. System, loads and displacements.

singular input and output vectors $\boldsymbol{v}_k$ and $\boldsymbol{u}_k$. Also, being the singular values sorted in descending order, one has:

$$\sigma_1 = \max_{\boldsymbol{v}} \frac{||\boldsymbol{G}\boldsymbol{v}||_2}{||\boldsymbol{v}||_2}, \tag{12}$$

that is the main reason why focusing on σ_1 only, as the two-norm does, makes full sense.

5. Generalized definition(s) of compliance

For the sake of simplicity reference is made to the three-storey shear-type structure sketched in Fig. 4. What matters is that the load vector $\boldsymbol{f} = \begin{bmatrix} f_1 & f_3 \end{bmatrix}^T$ is the dual (in the virtual work sense) of the loaded displacement vector $\boldsymbol{s} = \begin{bmatrix} x_1 & x_3 \end{bmatrix}^T \subset \boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$. The condition $\boldsymbol{s} \subset \boldsymbol{x}$ makes the problem general enough to replicate any case of practical interest but is at the same time simple enough to allow a synthetic presentation of all formulations of practical interest in the framework of compliance minimization. The four possible cases, i.e. Single-Input Single-Output (SISO), Multi-Input Single-Output (MISO), Single-Input Multi-Output (SIMO) and Multi-Input Multi-Output (MIMO) are presented next.

5.1. The SISO case

In the SISO case $n_o = 1$ and $r = 1$, i.e. we are aiming to minimize the classic scalar compliance in the presence of a single load case to which all the external forces belong. Upon writing

$$\boldsymbol{B} = \begin{bmatrix} f_1 \\ 0 \\ f_3 \end{bmatrix}, \ \boldsymbol{f} = 1, \ \boldsymbol{C} = \begin{bmatrix} f_1 & 0 & f_3 \end{bmatrix} = \boldsymbol{B}^T, \tag{13}$$

the output vector $\boldsymbol{z}$ reduces to the scalar compliance $\mathscr{C}$ that reads:

$$\boldsymbol{z} = \boldsymbol{C}\boldsymbol{x} = \begin{bmatrix} f_1 & 0 & f_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = f_1 x_1 + f_3 x_3 \equiv \mathscr{C}. \tag{14}$$

One should notice that in a SISO setting $\boldsymbol{\Sigma} = \sigma_1$ reduces to a scalar as well and therefore the three norms (2, Nuclear and Frobenius) coincide.

5.2. The MISO case

In this case $n_o = 1$ and $n_i = 2$, i.e. the two loads f_1 and f_3 belong to two different load cases. One therefore writes:

$$\boldsymbol{B} = \begin{bmatrix} f_1 & 0 \\ 0 & 0 \\ 0 & f_3 \end{bmatrix}, \ \boldsymbol{f} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} f_1 & 0 & f_3 \end{bmatrix}. \tag{15}$$

The resulting compliance $\mathscr{C}$ then reads:

$$\boldsymbol{z} = \begin{bmatrix} f_1 & 0 & f_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = f_1 x_1 + f_3 x_3 = \mathscr{C}_1 + \mathscr{C}_2 = \mathscr{C}. \tag{16}$$

In this case the size of $\boldsymbol{G}$ is $n_o \times n_i = 1 \times 2$, $\boldsymbol{U}$ is a 1×1 (scalar) whereas attention should be paid to $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ that respectively write:

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{bmatrix} = \left[\begin{array}{c|c} v_{11} & v_{21} \\ v_{12} & v_{22} \end{array}\right], \ \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \end{bmatrix}. \tag{17}$$

The vector $\boldsymbol{v}_1$ spans the row space of $\boldsymbol{G}$ and is associated to the singular value σ_1 whereas $\boldsymbol{v}_2$ spans the null space $\mathscr{N}(\boldsymbol{G})$. From an engineering point of view this is of the utmost importance:

- $\boldsymbol{v}_1$ spanning the row space of $\boldsymbol{G}$ amounts to saying that the SVD, automatically, determines the linear combination of the two loads

  $$v_{11} f_1 + v_{12} f_3,$$

  to which the maximum gain is associated and such gain is actually σ_1;
- $\boldsymbol{v}_2$ spanning the null space of $\boldsymbol{G}$ means that to the load combination

  $$v_{21} f_1 + v_{22} f_3,$$

  there is a null associated compliance. Such a combination is therefore automatically ruled out from the computations by the SVD.

5.3. The SIMO case

In this case $n_o = 2$ and $n_i = 1$, i.e. the two loads f_1 and f_3 belong to the same (and only) load case whereas two outputs exist that are $\mathscr{C}_1$ and $\mathscr{C}_2$, i.e. the compliance associated to f_1 and f_3, respectively. In formulas one has:

$$\boldsymbol{B} = \begin{bmatrix} f_1 \\ 0 \\ f_3 \end{bmatrix}, \ \boldsymbol{f} = 1, \ \boldsymbol{C} = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & 0 & f_3 \end{bmatrix}, \tag{18}$$

such that the output vector reads

$$\boldsymbol{z} = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & 0 & f_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} f_1 x_1 \\ f_3 x_3 \end{bmatrix} \equiv \begin{bmatrix} \mathscr{C}_1 \\ \mathscr{C}_2 \end{bmatrix}. \tag{19}$$

In this case the size of $\boldsymbol{G}$ is $n_o \times n_i = 2 \times 1$, $\boldsymbol{V}$ is a 1×1 scalar, whereas attention should be paid to $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$ that respectively write:

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 \end{bmatrix} = \left[\begin{array}{c|c} u_{11} & u_{12} \\ u_{21} & u_{22} \end{array}\right], \ \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 \\ 0 \end{bmatrix}. \tag{20}$$

The vector $\boldsymbol{u}_1$ spans the column space of $\boldsymbol{G}$ and is associated to the singular value σ_1 whereas $\boldsymbol{u}_2$ spans the so-called left null space $\mathscr{N}(\boldsymbol{G}^T)$. From an engineering point of view the following considerations are of interest:

- $\boldsymbol{u}_1$ spanning the column space of $\boldsymbol{G}$ amounts to saying that the SVD, automatically, determines the linear combination of the compliance $\mathscr{C}_1$ and $\mathscr{C}_2$

  $$u_{11} \mathscr{C}_1 + u_{21} \mathscr{C}_2,$$

  to which the maximum gain is associated and such gain is actually σ_1;
- $\boldsymbol{u}_2$ spanning the left null space of $\boldsymbol{G}$ means that to the compliance combination

  $$u_{12} \mathscr{C}_1 + u_{22} \mathscr{C}_2,$$

  there is a null associated input. Such a combination is therefore automatically ruled out from the computations by the SVD.

5.4. The MIMO case

In this case $n_o = 2$ and $n_i = 2$, i.e. the two loads f_1 and f_3 belong to two different load cases and the compliance $\mathscr{C}_1$ and $\mathscr{C}_2$ are independent entries of the output vector $\boldsymbol{z}$. Vectors and matrices are now as follows:

$$\boldsymbol{B} = \begin{bmatrix} f_1 & 0 \\ 0 & 0 \\ 0 & f_3 \end{bmatrix}, \ \boldsymbol{f} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & 0 & f_3 \end{bmatrix} = \boldsymbol{B}^T. \tag{21}$$

In general, the $n_o \times n_i = 2 \times 2$ transfer matrix $\boldsymbol{G}$ is full rank and the matrix $\boldsymbol{\Sigma}$ is given as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}, \tag{22}$$

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. The null space $\mathcal{N}(\boldsymbol{G})$ and the left null space $\mathcal{N}(\boldsymbol{G}^T)$ reduces to the zero vector $\boldsymbol{0}$ and the matrix $\boldsymbol{G}$ separates into two rank-1 contributions:

$$\boldsymbol{G} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T, \tag{23}$$

that means that both the triples $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ and $(\sigma_2, \boldsymbol{u}_2, \boldsymbol{v}_2)$ contribute to the overall response of the structure. Since $\sigma_1 > \sigma_2$, the contribution of the first triplet to the overall response is expected to be larger than the one of the second triplet. This justifies the frequent adoption of the 2-norm when it comes to assessing the size of a transfer matrix.

6. The topology optimization problem and its numerical solution

We are now in the position to state the general topology optimization problem based on the minimization of a norm of the transfer matrix $\boldsymbol{G}$:

$$\begin{cases} \min\limits_{\boldsymbol{p}} \mathscr{F}(\boldsymbol{p}) & = & ||\boldsymbol{G}(\boldsymbol{p})|| \\ \text{s.t.} & & \boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{C}\boldsymbol{K}^{-1}(\boldsymbol{p})\boldsymbol{B} \\ & & V(\boldsymbol{p}) \le V_{\max} \\ & & \boldsymbol{p}_{\min} \le \boldsymbol{p} \le \boldsymbol{p}_{\max} \end{cases}, \tag{24}$$

where:

- $(24)_1$ sets the norm $||\boldsymbol{G}(\boldsymbol{p})||$ as the goal function $\mathscr{F}(\boldsymbol{p})$ to be minimized in the space of the design parameters grouped in the vector $\boldsymbol{p}$. As for the choice of the norm, the ones in Section 4.2 (2, Nuclear and Frobenius) appear to be convenient ones in that they are uniquely defined in terms of the singular values of $\boldsymbol{G}$;
- the definition of the transfer matrix $\boldsymbol{G}$ itself enters the formulation as the constraint given in $(24)_2$;
- the standard constraint on the maximum allowable volume is given in $(24)_3$. In this respect, the goal function $(24)_1$ and the volume constraint $(24)_3$ may be switched so as to arrive at a minimum volume problem under the constraint of a maximum norm of the transfer matrix $\boldsymbol{G}$. Such possibility is not exploited herein and is left to future investigations;
- Side constraints on the densities at the element level are imposed in $(24)_4$ that should be read componentwise. As usual a SIMP interpolation of the densities is adopted following [18].

The solution of Problem (24) is herein pursued via the well-known method of moving asymptotes (MMA) [1]. As for all gradient-based approaches, the goal function and the constraints should be computed along with their sensitivities. Next paragraph gives the details on the computation of $||\boldsymbol{G}||$ (for the 2, Nuclear and Frobenius norms) and its sensitivity using Matlab (but the proposed developments apply to any programming language of current use).

6.1. Using the proposed formulation vs standard approaches

There arise a few main advantages when using the proposed formulation over more standard ones (though this is not a claim of superiority whatsoever):

- when the output is the compliance, as it happens in the vast majority of cases, the proposed formulation allows a systematic treatment of load cases and load combinations. The possibility offered by the MIMO idealization to have vectors as inputs and output offers in the author's opinion an unparalleled flexibility and generality as far as loading and optimal design are concerned.
- the goal function is however not restricted to the compliance but may include any quantity of engineering interest such as displacements, stresses and strains. When this is the case, no specific formulas should be derived for the semi-analytic computation of the sensitivities, since the formulation. In fact, whatever the physical meaning of the goal function, when computing the sensitivities it all comes to evaluating the sensitivities of the singular values for which the relevant formulas were derived once for all.

6.2. Computing $||\boldsymbol{G}||$ and its gradient

The main ingredient to compute the three norms of $\boldsymbol{G}$ and their gradients is the singular value decomposition of $\boldsymbol{G}$ itself. In the Matlab environment adopted herein the following line of code solves the problem:

```
[U,S,V] = svds(G),
```
(25)

but any function (in any programming language) performing the SVD does the job. In Equation (25), `[U,S,V]` are respectively the matrices of left singular vectors, singular values and right singular vectors (see also Equation (7) and notations thereafter). Once the singular values of $\boldsymbol{G}$ are available, the three norms may be computed by definition as shown in Equation (8).

6.2.1. Computing the gradient of three norms $||\boldsymbol{G}||_{2,N,F}$

For all the three norms of interest, the chain rule applies and allows to write the sensitivity of $\boldsymbol{G}$ with respect to any component of the design variable vector $\boldsymbol{p}$, say p_k, as

$$\frac{\partial ||\boldsymbol{G}||}{\partial p_k} = \sum_{h=1}^{r} \frac{\partial ||\boldsymbol{G}||}{\partial \sigma_h} \frac{\partial \sigma_h}{\partial p_k}. \tag{26}$$

The first factor in Equation (26) is straightforward and for each of the norms respectively reads:

$$\begin{aligned} \frac{\partial ||\boldsymbol{G}||_2}{\partial \sigma_h} &= \frac{\partial \sigma_1}{\partial \sigma_h} = \delta_{1h} \ \text{(Kronecker delta)} \\ \frac{\partial ||\boldsymbol{G}||_N}{\partial \sigma_h} &= \frac{\partial (\sigma_1 + \ldots + \sigma_r)}{\partial \sigma_h} = 1 \\ \frac{\partial ||\boldsymbol{G}||_F}{\partial \sigma_h} &= \frac{\sigma_h}{\sqrt{\sigma_1^2 + \ldots + \sigma_r^2}} \end{aligned}. \tag{27}$$

The second factor in Equation (26) is more challenging and represents an issue in that the existence of the formula of practical use adopted hereinafter depends on some very technical conditions for which the reader is referred to [20][21]. We shall assume such conditions to be met hereinafter. However, a critical peculiarity in this respect of the 2-norm only shall be highlighted for the sake of completeness in the paragraph presenting numerical results. The formula being developed and used hereafter is eventually the following one:

$$\frac{\partial \sigma_h}{\partial p_k} = \mathscr{R}\left(\boldsymbol{u}_h^T \frac{\partial \boldsymbol{G}(\boldsymbol{p})}{\partial p_k} \boldsymbol{v}_h \right), \tag{28}$$

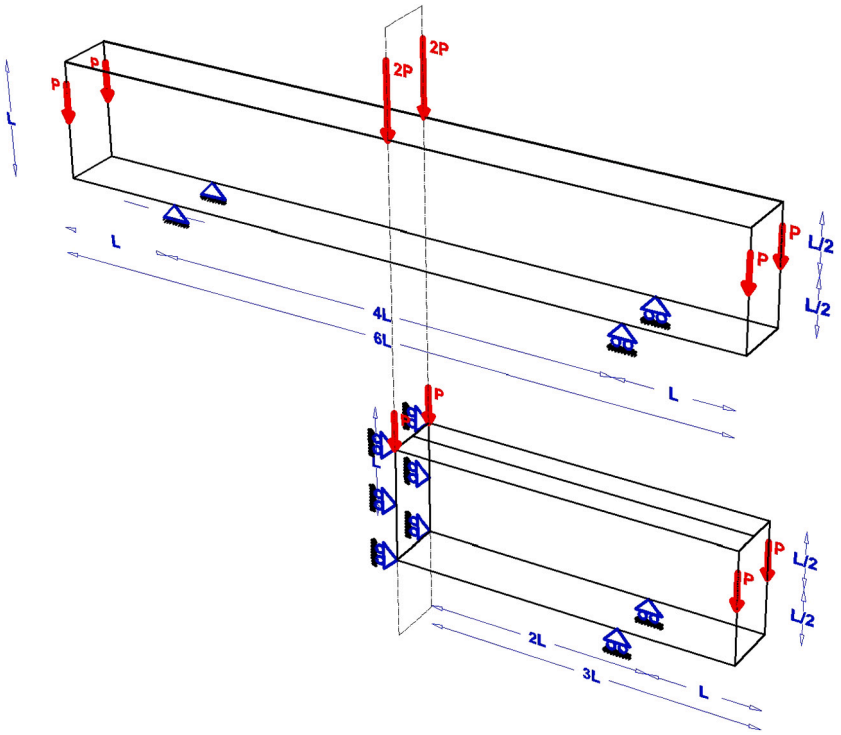

Fig. 5. Cantilevered structure - The engineering problem.

where $\mathscr{R}$ indicates the real part of a (possibly) complex number. What is then needed to proceed is a semi-analytic formula for the sensitivity of the $\boldsymbol{G}$ matrix. If one recalls that sensitivity of the inverse reads:

$$\frac{\partial \boldsymbol{K}^{-1}}{\partial p_k} = -\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial p_k}\boldsymbol{K}^{-1}, \tag{29}$$

upon direct differentiation of Equation (5) one gets:

$$\begin{aligned}\frac{\partial \boldsymbol{G}(\boldsymbol{p})}{\partial p_k} &= \boldsymbol{C}\frac{\partial \boldsymbol{K}^{-1}}{\partial p_k}\boldsymbol{B} = \\ &= -\boldsymbol{C}\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial p_k}\boldsymbol{K}^{-1}\boldsymbol{B}\end{aligned}. \tag{30}$$

By plugging (30) into (28) and the resulting expression into (26), the norm sensitivity is eventually computed.

7. Numerical results

A few representative numerical results are presented hereafter to highlight the main features of the proposed SVD-based approach. The four possible system idealizations (SISO, MISO, SIMO and MIMO) as well as the three proposed norms defined in terms of the transfer-matrix singular values (2, Nuclear and Frobenius) are all exploited and the main features of relevant designs highlighted.

It seems worth emphasizing at this point the capability of the proposed approach to deal with multiple objectives, as is it happens for the Multi-Output cases (both SIMO and MIMO) that are investigated in the cantilevered beam (Problem 1) and the tall building (Problem 2) studies. Other than resorting to Pareto-like strategies, using norm-based approaches appears to be nearly the only available method to pursue multiple-goal design optimization.

Toward a better understanding of the results to be presented below, a few features of the proposed approach that were introduced earlier on in the manuscript are recalled next:

- The engineering meaning of the transfer matrix $\boldsymbol{G}$ depends on the specific choice of the designer. More often than not the output of $\boldsymbol{G}$ happens to be the compliance $\mathscr{G}$. When this is the case, the convergence curves that show the evolution of $||\boldsymbol{G}||$ with the iterations are actually representing the evolution of the compliance $\mathscr{C}$ itself (but eventually for the number of outputs that may be larger than one as recalled next);
- In general $\boldsymbol{G}$ has as many rows as there are outputs, i.e. n_o that is the number of quantities of engineering interest for that specific design, and as many columns as there are inputs, i.e. independent loads acting upon the structure. This abstract setting encompasses a wide variety of possible cases, ranging from the simplest one ($n_o = n_i = 1$, i.e. the SISO case such that a single load case is considered, and a single overall compliance is being minimized, in which case the "matrix" $\boldsymbol{G}$ reduces to a scalar), all the way up to very complex idealizations where several independent load cases are considered (n_i large) and several outputs are present as well (n_o large), all of which concur to the definition of $||\boldsymbol{G}||$.

7.1. Problem 1 - cantilevered beam

To start with, the cantilevered structure shown in Fig. 5 is considered. Thanks to the symmetry of the problem, only the half-structure in Fig. 5 is modeled. Mid-span and free-edge loads act upon the structure within either a Single-Input idealization (that happens when the two loads belong to the same load case) or a Multi-Input one wherein two independent load cases are assumed. As for the amplitude of the loads, the following comments are appropriate at this stage:

- what matters toward the definition of the optimal topology at convergence is the amplitude of the different load sets relative to one another. With reference to Fig. 5, mid-span and free-edge load are assigned the same amplitude (after halving the overall structure). Of course, if different load rations are considered, different optimal topologies are found;
- the absolute amplitude of the load themselves has conversely no relevance for the sake of this paper wherein linear materials are considered, but it would have an importance of its own as soon as nonlinear materials and/or stress constraints enter the formulation.

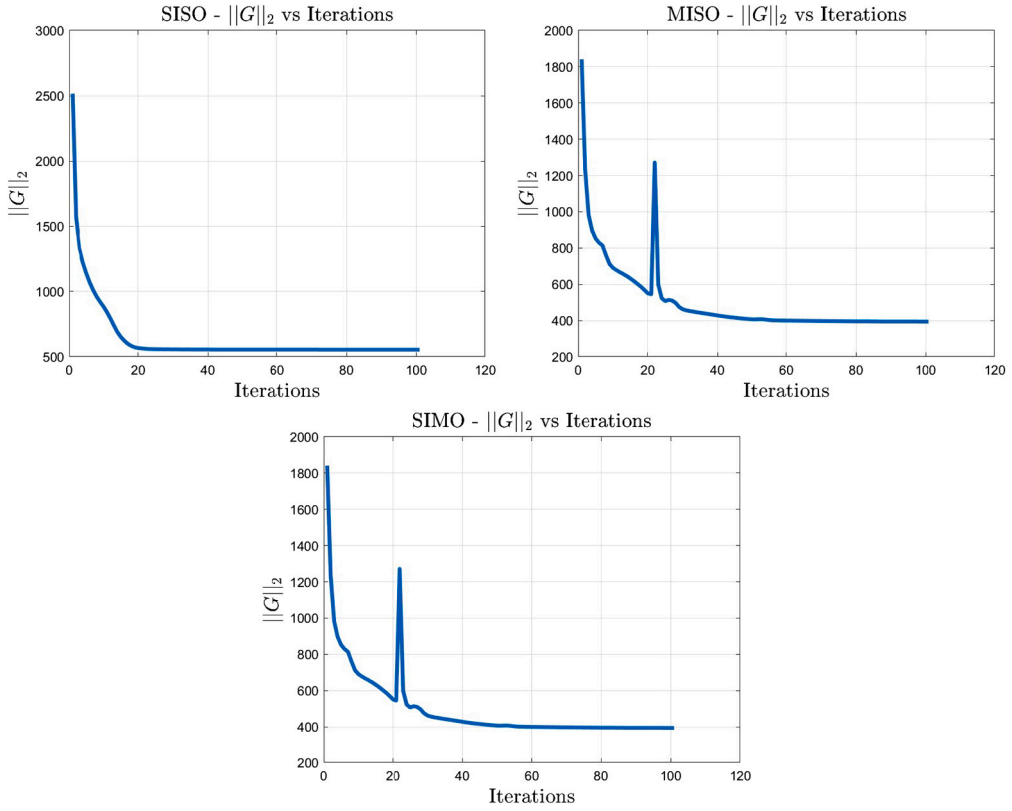

Fig. 6. Cantilevered structure - SISO (up), MISO (mid), SIMO (bottom) convergence curves.

Analogous considerations apply to all the numerical studies presented hereinafter.

The output is likewise allowed to be a single one, i.e. a single overall compliance is minimized, or a multiple one wherein each compliance is associated to a single load and represent an independent entry of the output vector $\boldsymbol{z}$. Geometry, mechanical and numerical parameters for the optimization of the problem are summarized next:

- `nely` = 60 (number of elements in the vertical direction);
- `nelx` = 3 × `nely` = 180 (number of elements in the longitudinal horizontal direction);
- `nelz` = `nely`/10 = 6 (number of elements in the transverse (depth) direction);
- `volfrac` = 0.5 (maximum volume fraction available for material);
- `penal` = 3.0 (SIMP penalization factor);
- `rmin` = 4.5 (filter distance);
- `E0` = 1 (Nominal Young modulus);
- `Emin` = 1e-9 (Minimum (void) Young modulus);
- `nu` = 0.3; (Constant Poisson ratio).

7.1.1. SISO, MISO and SIMO formulations

Though technically different, SISO, MISO and SIMO idealizations share the fact that the underlying transfer matrix $\boldsymbol{G}$ has rank one. A single non-trivial singular value σ_1 is therefore available and this implies that the three norms introduced above coincide. Relevant optimal designs are therefore quite similar as are the convergence curves of the method of moving asymptotes. Convergence curves and optimal topologies are respectively shown in Fig. 6 and Fig. 7.

A key point to fully understand the engineering nuances of the SVD-based approach is discussed next that is based on the general formulation of Section 5. Matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ that define the singular value decomposition of $\boldsymbol{G}$ at convergence are also presented for all the input-output cases.

- SISO case. The "transfer matrix" $\boldsymbol{G}$ is actually a scalar quantity and one has

$$\boldsymbol{U} = \boldsymbol{V} = 1,\ \boldsymbol{\Sigma} = \sigma_1 = 554.30. \tag{31}$$

The SISO case basically replicates the standard scalar compliance minimization of a structure acted upon loads belonging to a single load case.

- MISO case. $\boldsymbol{G}$ is in this case a 1×2 matrix. Matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ respectively read:

$$\boldsymbol{U} = 1,\ \boldsymbol{V} = \begin{bmatrix} 0.696 & -0.718 \\ 0.718 & 0.696 \end{bmatrix},\ \boldsymbol{\Sigma} = \begin{bmatrix} 394.11 & 0 \end{bmatrix}. \tag{32}$$

The compliance $\mathscr{C}_1$ and $\mathscr{C}_2$ are evaluated independently of one another and are respectively due to the midspan load, say f_1, and the free–edge load, say f_2. The components of the first singular vector $\boldsymbol{v}_1$ span the row space of $\boldsymbol{G}$, i.e. they provide the coefficients of the linear combination

$$\underbrace{v_{11} f_1}_{\mathscr{C}_1} + \underbrace{v_{12} f_2}_{\mathscr{C}_2} = 0.696 \cdot f_1 + 0.718 \cdot f_2 = \mathscr{C}, \tag{33}$$

to which the maximum plant gain (i.e. the largest singular value σ_1) is associated. On the other hand, upon combining the two input vectors using as coefficients the components of the second input singular vector $\boldsymbol{v}_2$, i.e.

$$\underbrace{v_{21} f_1}_{\mathscr{C}_1} + \underbrace{v_{22} f_2}_{\mathscr{C}_2} = -0.718 \cdot f_1 + 0.696 \cdot f_2, \tag{34}$$

a null compliance is obtained, since $\boldsymbol{v}_2$ spans the null space $\mathcal{N}(\boldsymbol{G})$. The bottom line here is therefore the fact that SVD-based topology optimization inherently focuses the design on the worst-case scenario as far as load combination is concerned whereas trivial (i.e. with null associated compliance) are automatically ruled out.

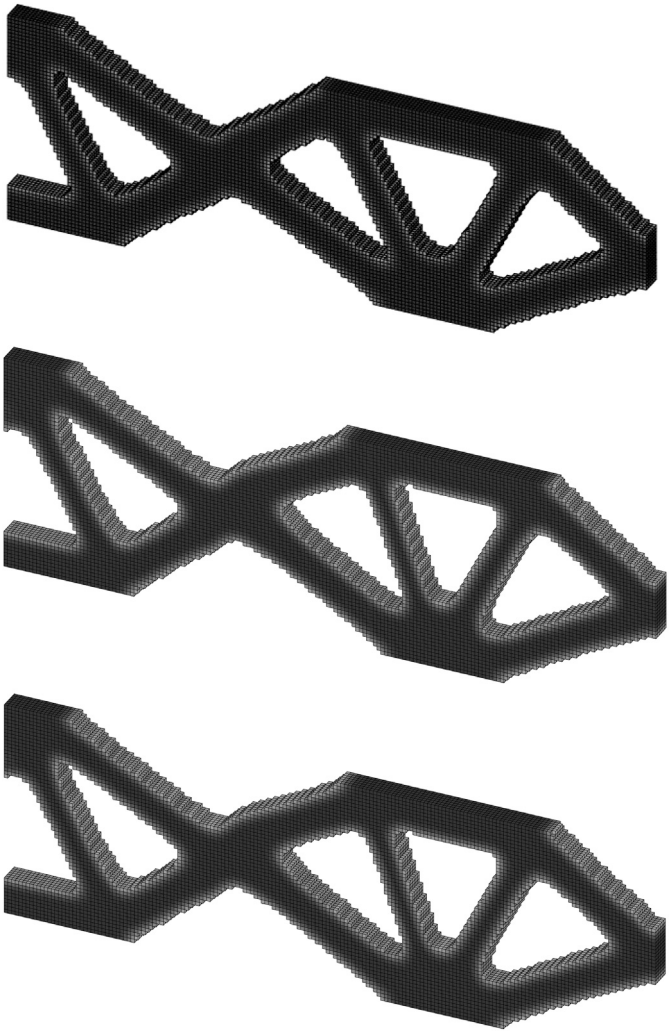

Fig. 7. Cantilevered structure - SISO (up), MISO (mid), SIMO (bottom) 3D views of the optimal topologies.

- SIMO case. $\boldsymbol{G}$ is in this case a 2×1 matrix. Matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ respectively read:

$$\boldsymbol{U}=\begin{bmatrix}0.696 & -0.718\\ 0.718 & 0.696\end{bmatrix},\ \boldsymbol{V}=1,\ \boldsymbol{\Sigma}=\begin{bmatrix}394.11\\ 0\end{bmatrix}. \tag{35}$$

In this case, $\mathscr{C}_1$ and $\mathscr{C}_2$ represent the independent entries of the 2×1 output vector $\boldsymbol{z}$. The vector $\boldsymbol{u}_1$ spans the column space of $\boldsymbol{G}$ that is to say the linear combination

$$u_{11}\mathscr{C}_1+u_{21}\mathscr{C}_2=0.696\cdot\mathscr{C}_1+0.718\cdot\mathscr{C}_2 \tag{36}$$

is the one to which the maximum blow up factor σ_1 is associated. On the other side, $\boldsymbol{u}_2$ spans the left null space $\mathscr{N}(\boldsymbol{G}^T)$ and is therefore ruled out from the optimization procedure.

7.1.2. MIMO formulation: 2, N and F norms and standard multiload compliance design

The MIMO case originates a much wider variety of design solutions as a result of the fact that the relevant transfer matrix $\boldsymbol{G}$ is full rank (having in fact rank 2). Convergence curves for the three norms investigated herein exhibit quite a similar pattern as shown in Fig. 8. In examining the optimal designs in Fig. 9, one should recall that the 2-norm aims to minimize the first (largest) singular value only, whereas nuclear and Frobenius norms depend on both the singular values of the transfer matrix $\boldsymbol{G}$. That is why the 2-norm-based optimal topology (Fig. 9-up) represents a class of its own whereas Nuclear and Frobenius (Fig. 9-mid and bottom) designs belong to the same family, so to speak, as does the one obtained using Liu and Tovar's code [2] under the multiload option, that is shown in Fig. 10. This consideration is quite an important one in view of the extension of the presented approach to the dynamic regime (that is not pursued herein but shall be the object of a forthcoming contribution). In dynamics, the 2-norm originates the celebrated H_∞-norm that is in fact the supremum of the largest singular value over the whole frequency axis. Such norm is widely used especially when robustness with respect to plant uncertainties is an issue. However, focusing on the largest singular value only may not be a sound strategy, especially when the amplitude of smaller singular values is not negligible (see the rank-one decomposition of $\boldsymbol{G}$ in Equation (9) but also the Eckart-Young approximation theorem [14]). Upon comparing the convergence curves in Fig. 8, that refer to the norm-based approach presented herein, with the one in Fig. 10, that refers to Liu and Tovar's approach, a similar pattern emerges, even though the latter exhibits an apparently smoother graph. For a general lesson on this topic more insight could be probably gained by considering CPU time as a further variable. This is part of a short-term extension currently under investigation. Toward the assessment of the efficiency of each of the available designs, matrices entering the SVD of all the transfer matrices $\boldsymbol{G}$ at convergence are reported next.

Table 1
Cantilevered structure - Norm competition between the four available designs.

	$\|\|\mathscr{C}\|\|_2$	$\|\|\mathscr{C}\|\|_N$	$\|\|\mathscr{C}\|\|_F$
2-norm design	**827.28**	1399.71	1006.02
Nuclear norm design	848.73	**1192.73**	915.79
Frobenius norm design	832.52	1198.74	**909.51**
Liu and Tovar's code	885.96	1243.80	955.50

- 2 Norm.

$$\boldsymbol{V}=\begin{bmatrix}-0.644 & 0.765\\ 0.765 & 0.644\end{bmatrix},\ \boldsymbol{U}=\begin{bmatrix}-0.644 & 0.765\\ 0.765 & 0.644\end{bmatrix}$$
$$\boldsymbol{\Sigma}=\begin{bmatrix}827.28 & 0\\ 0 & 572.43\end{bmatrix} \tag{37}$$

- Nuclear norm.

$$\boldsymbol{V}=\begin{bmatrix}-0.620 & 0.785\\ 0.785 & 0.620\end{bmatrix},\ \boldsymbol{U}=\begin{bmatrix}-0.620 & 0.785\\ 0.785 & 0.620\end{bmatrix}$$
$$\boldsymbol{\Sigma}=\begin{bmatrix}848.73 & 0\\ 0 & 344.00\end{bmatrix} \tag{38}$$

- Frobenius norm.

$$\boldsymbol{V}=\begin{bmatrix}-0.635 & 0.773\\ 0.773 & 0.635\end{bmatrix},\ \boldsymbol{U}=\begin{bmatrix}-0.635 & 0.773\\ 0.773 & 0.635\end{bmatrix}$$
$$\boldsymbol{\Sigma}=\begin{bmatrix}832.52 & 0\\ 0 & 366.22\end{bmatrix} \tag{39}$$

- Liu and Tovar's code.

$$\boldsymbol{V}=\begin{bmatrix}-0.637 & 0.771\\ 0.771 & 0.637\end{bmatrix},\ \boldsymbol{U}=\begin{bmatrix}-0.637 & 0.771\\ 0.771 & 0.637\end{bmatrix}$$
$$\boldsymbol{\Sigma}=\begin{bmatrix}885.96 & 0\\ 0 & 357.84\end{bmatrix} \tag{40}$$

Table 1 presents the evaluation of the three norms (2, Nuclear and Frobenius) for all available designs. The following conclusions may be drawn:

- as of course expected (results in bold-face on the diagonal of the table), the minimum of each norm is attained by the design that was explicitly aiming at its minimization;

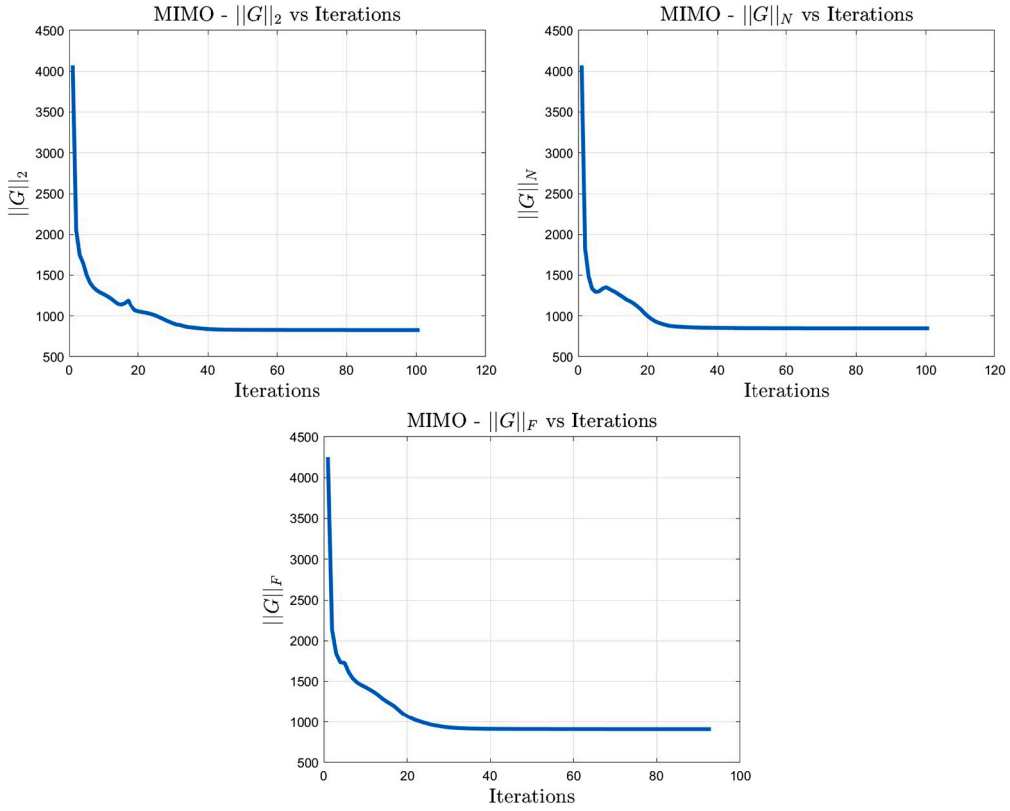

Fig. 8. Cantilevered structure - MIMO formulation - 2 (up), Nuclear (mid), Frobenius (bottom) convergence curves.

- it is fair to say that Liu and Tovar's code based design is not based on a norm minimization (as other designs do). Nevertheless, the performance of this design is shown to be quite a satisfactory one with respect to all the three norms considered in the paper;
- a consideration of its own should be made concerning the performance of the 2-norm. The 2-norm minimizes the largest singular value σ_1 that from an engineering point of view is the maximum gain σ_1 (the blow-factor using Strang's terminology [14]). It is therefore clear the motivation behind the choice of minimizing the 2-norm (or the H_∞-norm in dynamics). However, when the plant is a multi-modal one, i.e. when smaller singular values from σ_2 on do not have a negligible amplitude, focusing on σ_1 only may not be a proper design strategy. The severe performance drop of the 2-norm based design becomes clear when all the singular values are considered: Nuclear and Frobenius norms of the 2-norm based design happen to be way larger than those one gets using the two other design approaches. Therefore, the singular input-output channels from $(\boldsymbol{v}_2, \boldsymbol{u}_2)$ on may lead to unexpectedly poor results when not explicitly included as design goals to be taken care of.

An assessment of the overall behavior of the available designs may be gathered by looking at Table 2 wherein the following performance index PI (the smaller, the better) is evaluated for each of the designs:

$$\begin{aligned} PI \quad = \quad & \Big[\left(||\mathscr{C}||_2 - \min ||\mathscr{C}||_2 \right)^2 / \left(\min ||\mathscr{C}||_2 \right)^2 + \\ & \left(||\mathscr{C}||_N - \min ||\mathscr{C}||_N \right)^2 / \left(\min ||\mathscr{C}||_N \right)^2 + \\ & \left(||\mathscr{C}||_F - \min ||\mathscr{C}||_F \right)^2 / \left(\min ||\mathscr{C}||_F \right)^2 \Big]^{1/2} . \end{aligned} \tag{41}$$

One may notice once again that focusing on the largest singular value σ_1 only as the 2-norm does, allows the maximum reduction along the "most-important" plant channel $(\boldsymbol{v}_1, \boldsymbol{u}_1)$ but leads to a final design of questionable overall performance.

Table 2
Cantilevered structure - Overall performance of the designs (the smaller, the better).

	10^2 Performance Index PI
2-norm design	9.120
Nuclear norm design	5.097
Frobenius norm design	2.956
Liu and Tovar's code	6.802

Table 3
Cantilevered structure - Computational cost: CPU time in seconds.

Design type	CPU time (s)
2-norm design	1180
Nuclear norm design	1147
Frobenius norm design	1107
Liu and Tovar's code	1137

7.2. Computational cost

A few consideration on the computational cost needed to achieve the optimal solutions at convergence using the proposed approach and the one in [2] are given below. Reference is made to the optimal topologies in Section 7.1.2 that are concerned with Multi-Input, Multi-Output designs (Fig. 9 and Fig. 10, Table 3). Other than observing that all the four designs exhibit quite similar costs in terms of CPU time, the following observations seem to be appropriate:

1. the comparison between the proposed approach (using any of the three norms) and the one in [2] is not completely fair in that the proposed approach is based on a fully nonlinear programming approach that calls for the evaluation of the gradients of the objective

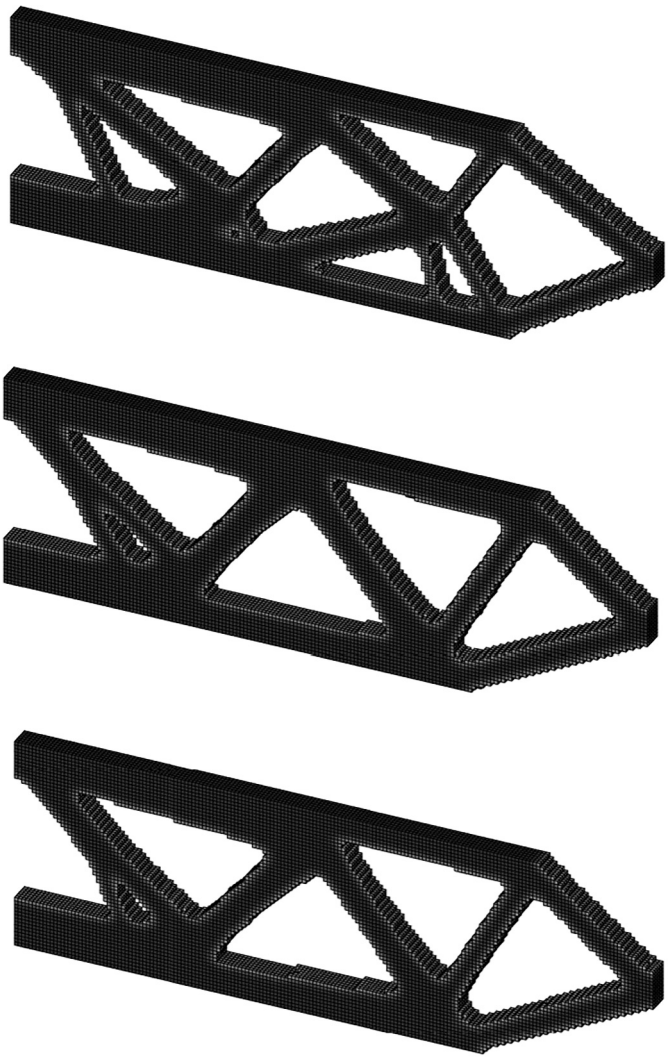

Fig. 9. Cantilevered structure - MIMO formulation - 2 (up), Nuclear (mid), Frobenius (bottom) 3D views.

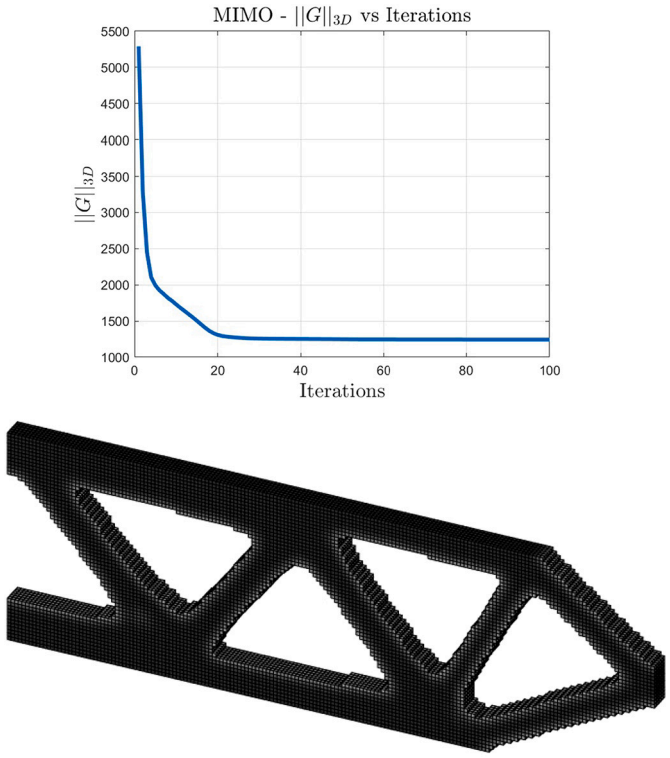

Fig. 10. Cantilevered structure - Convergence curve (up) and Optimal topology (bottom) using the Liu and Tovar's code with the multiload option.

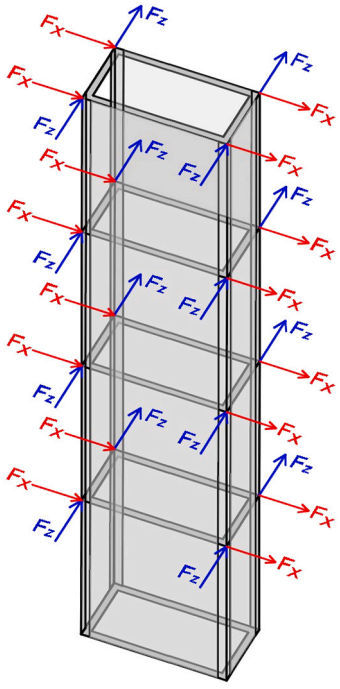

Fig. 11. Tall building structure - The engineering problem.

function, whereas [2] uses a heuristic formula to update the design variables that is certainly less demanding as far as CPU time is concerned;
2. the proposed approach uses most of the CPU time in the computation of the two factors $\boldsymbol{C}\boldsymbol{K}^{-1}$ (40% of the overall CPU time) and $\boldsymbol{K}^{-1}\boldsymbol{B}$ (40% of the overall CPU time) in Equation (30). Each of these two factors are computed once per iteration within the MMA solution scheme;
3. the method in [2] uses 60% of the overall CPU time in solving the equilibrium Equation (1), i.e. $\boldsymbol{K}\boldsymbol{U} = \boldsymbol{F}$.

Items 2 and 3 above show that inverting the stiffness matrix $\boldsymbol{K}$ (though not done explicitly in either method) represents the most demanding class of operations of the two approaches dealt with herein.

7.3. Problem 2 - tall building

7.3.1. Motivations

The proposed topology optimization approach is hereafter applied to the four-storey structure in Fig. 11 acted upon two sets of horizontal loads. The motivation of this investigation is twofold:

1. from a mechanical point of view, the design of braced elements to resist horizontal environmental loads such as strong winds and earthquakes is a matter of the utmost importance. Addressing such a crucial problem using structural and topology optimization techniques is a relatively recent strategy, see e.g. [22] and [23], mostly limited to 2D idealizations. A 3D approach that aims to the minimization of the (norm of the transfer matrix of) horizontal response seems therefore an interesting one for the engineering community;
2. there is however a further motivation of numerical nature. Though being a 3D problem wherein the overall resisting mechanism sees

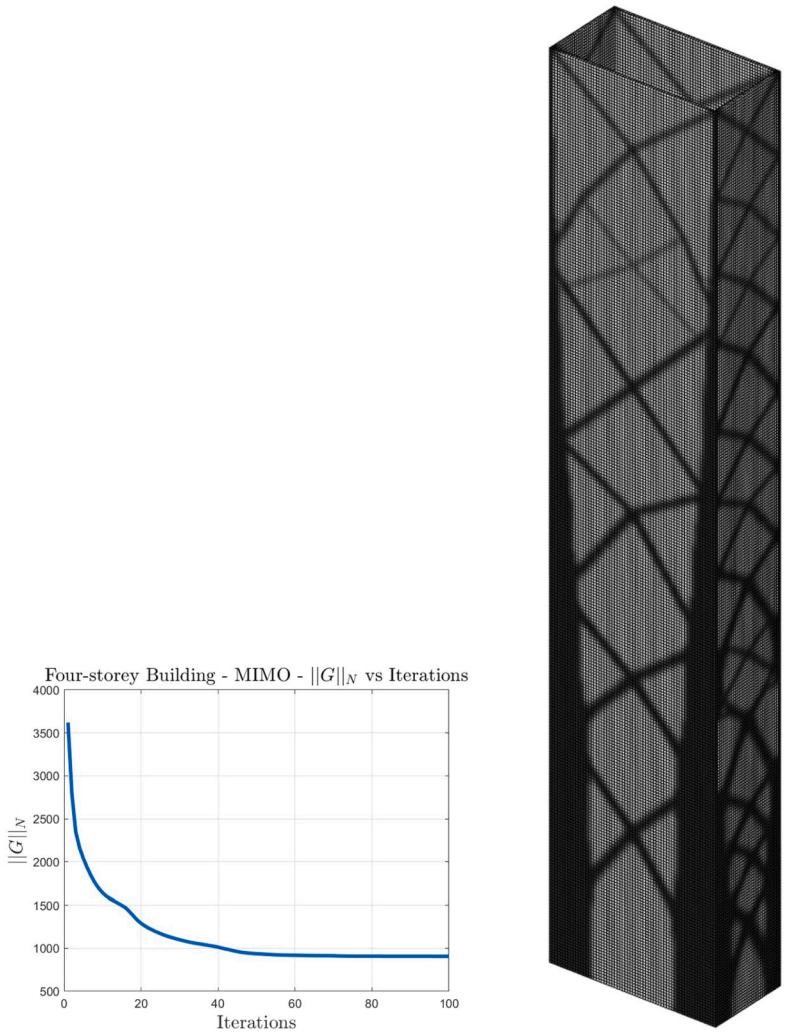

Fig. 12. Tall building - MIMO formulation - N-norm - Convergence curve and 3D view.

the cooperation of the bracing systems in the two orthogonal directions (especially as far as the torsional response is concerned), the problem is only weakly coupled in the two horizontal directions, each of which gives rise to a nearly independent input-output singular vector pair. For such type of structures, then, using a MIMO idealization coupled to a Nuclear and/or Frobenius norm is expected to be the most rewarding strategy. By so doing, in fact, both the singular values of the transfer matrix enter the goal function to be minimized so that the gains of both the loads in the two directions are addressed and resisted by the optimal structure. On the other side, using any strategy that depends on one singular value only, i.e. any SISO, MISO and SIMO approach or a MIMO model coupled to a 2-norm choice, may lead to numerical issues and questionable final designs as shown below. That is why in what follows Nuclear and Frobenius designs applied to a MIMO system are presented and represent the "true" ones, whereas those based on SISO, MISO and SIMO models as well as the MIMO case coupled to a 2-norm choice are reported for the sake of completeness, with the primary goal of enhancing relevant problems and indicating potential solving strategies object of ongoing investigation.

7.3.2. Tall building design against horizontal actions

The tall building structure displayed in Fig. 11 is analyzed next. The goal is to find a 3D-optimal structural topology capable to resist horizontal loads in the x and z directions (horizontal plane) that may either belong to a single load case or to two independent ones. Among the

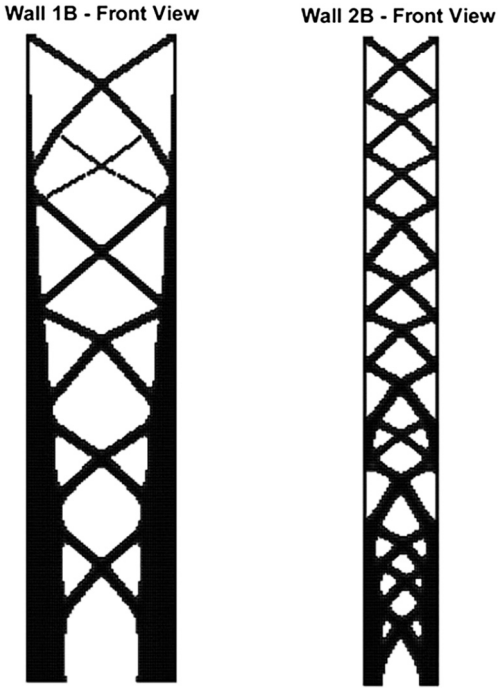

Fig. 13. Tall building - MIMO formulation - N-norm - Sides 1B and 2B.

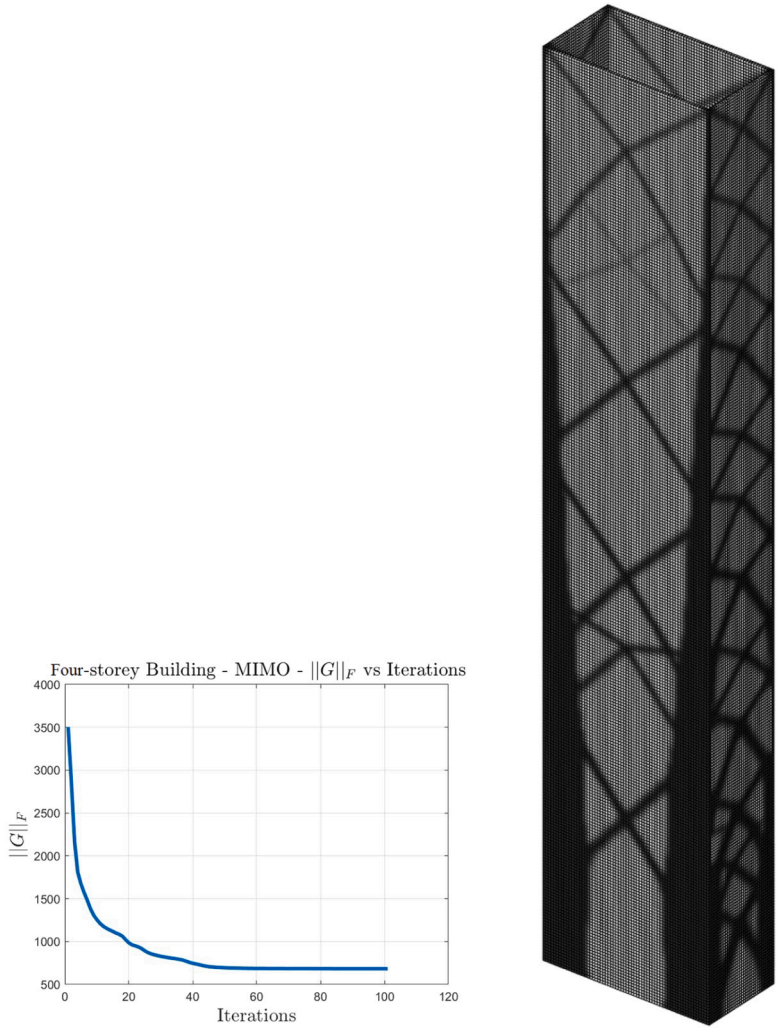

Fig. 14. Tall building - MIMO formulation - F-norm - Convergence curve and 3D view.

first contributions in this area, the papers of Glaucio Paulino's school [22] and [23] are worth mentioning. For the sake of completeness, it is recalled that vertical loads are not included or modeled within this analysis. However, an approach that couples existing steel frames that handle vertical loads with topologically optimal braces that increase the horizontal resistance is currently being developed and shall be presented in a forthcoming contribution. As for this latter topic, the exoskeleton design approach developed in [24] is among the most recent and successful ones. Referring to paragraph 3.2 for the meaning of a few symbols, geometry, mechanical and numerical parameters of the optimization problem are summarized next:

- `nelx1` = 2^6;
- `nstoreys` = 4;
- `nely`$_s$ = `ceil`(`nelx1`*1.25);
- `nely` = `nstoreys`*`nely`$_s$;
- `nely1` = `nely`;
- `nelz1` = 1;
- `nelx2` = `nelz1`;
- `nely2` = `nely`;
- `nelz2` = 2^5-2*`nelz1`;
- `volfrac` = 0.35 (maximum volume fraction available for material);
- `penal` = 3.0 (SIMP penalization factor);
- `rmin` = 2.25 (filter distance);
- `E0` = 100 (Nominal Young modulus);

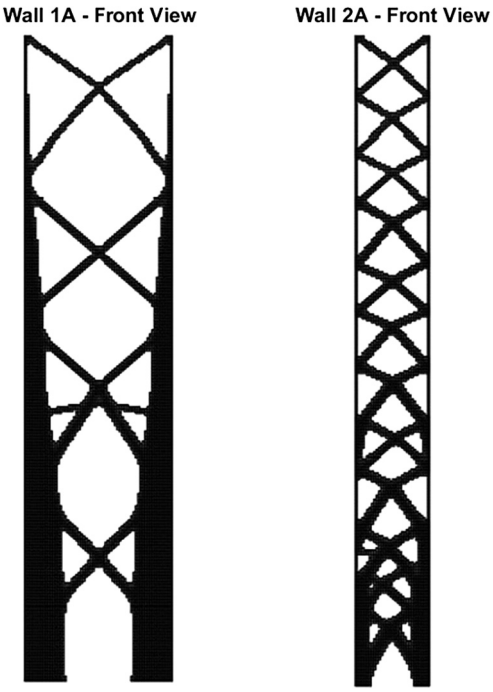

Fig. 15. Tall building - MIMO formulation - F-norm - Sides 1A and 2A.

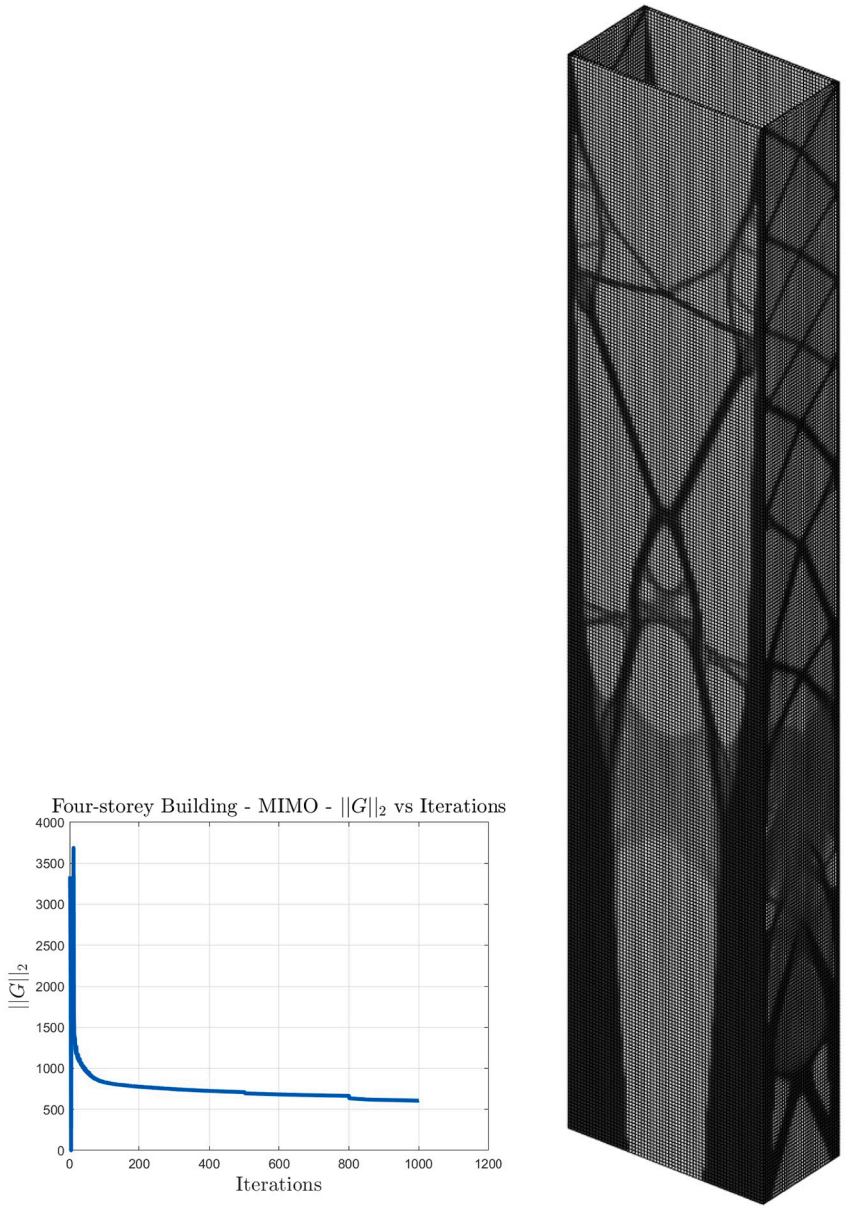

Fig. 16. Tall building - MIMO formulation - 2-norm - Convergence curve and 3D view.

- `Emin` = 1e-6 (Minimum (void) Young modulus);
- `nu` = 0.3; (Constant Poisson ratio).

7.3.3. MIMO formulation

To start with, the matrix factors of the singular value decomposition of $\boldsymbol{G}$ are presented for all the three available norms (Nuclear, Frobenius and 2).

- Nuclear norm.

$$\boldsymbol{V} = \begin{bmatrix} -0.000 & 1.000 \\ 1.000 & 0.000 \end{bmatrix}, \ \boldsymbol{U} = \begin{bmatrix} -0.000 & 1.000 \\ 1.000 & 0.000 \end{bmatrix}$$
$$\boldsymbol{\Sigma} = \begin{bmatrix} 633.70 & 0 \\ 0 & 273.47 \end{bmatrix} \tag{42}$$

- Frobenius norm.

$$\boldsymbol{V} = \begin{bmatrix} -0.000 & 1.000 \\ 1.000 & 0.000 \end{bmatrix}, \ \boldsymbol{U} = \begin{bmatrix} -0.000 & 1.000 \\ 1.000 & 0.000 \end{bmatrix}$$
$$\boldsymbol{\Sigma} = \begin{bmatrix} 604.10 & 0 \\ 0 & 318.29 \end{bmatrix} \tag{43}$$

- 2 Norm.

$$\boldsymbol{V} = \begin{bmatrix} -0.023 & 0.999 \\ 0.999 & 0.023 \end{bmatrix}, \ \boldsymbol{U} = \begin{bmatrix} -0.023 & 0.999 \\ 0.999 & 0.023 \end{bmatrix}$$
$$\boldsymbol{\Sigma} = \begin{bmatrix} 608.03 & 0 \\ 0 & 603.51 \end{bmatrix} \tag{44}$$

As for the left and right singular vectors in $\boldsymbol{U}$ and $\boldsymbol{V}$, the following points are worth being set:

- $\boldsymbol{v}_1 = [0\ 1]^T$ and $\boldsymbol{v}_2 = [1\ 0]^T$ indicate a weak (actually null) coupling between the two orthogonal directions x and z in the horizontal plane (see Fig. 11 for the definition of the x and z axis). Analogous considerations apply to $\boldsymbol{u}_1 = [0\ 1]^T$ and $\boldsymbol{u}_2 = [1\ 0]^T$;
- The first singular vectors $\boldsymbol{v}_1 = \boldsymbol{u}_1 = [0\ 1]^T$ are aligned with the z axis. From an engineering point view, this means that the maximum gain σ_1 is experienced along the z direction. This was of course expected since loads F_z are resisted by the moment of inertia I_x that is way lower than I_z that is the flexural moment of

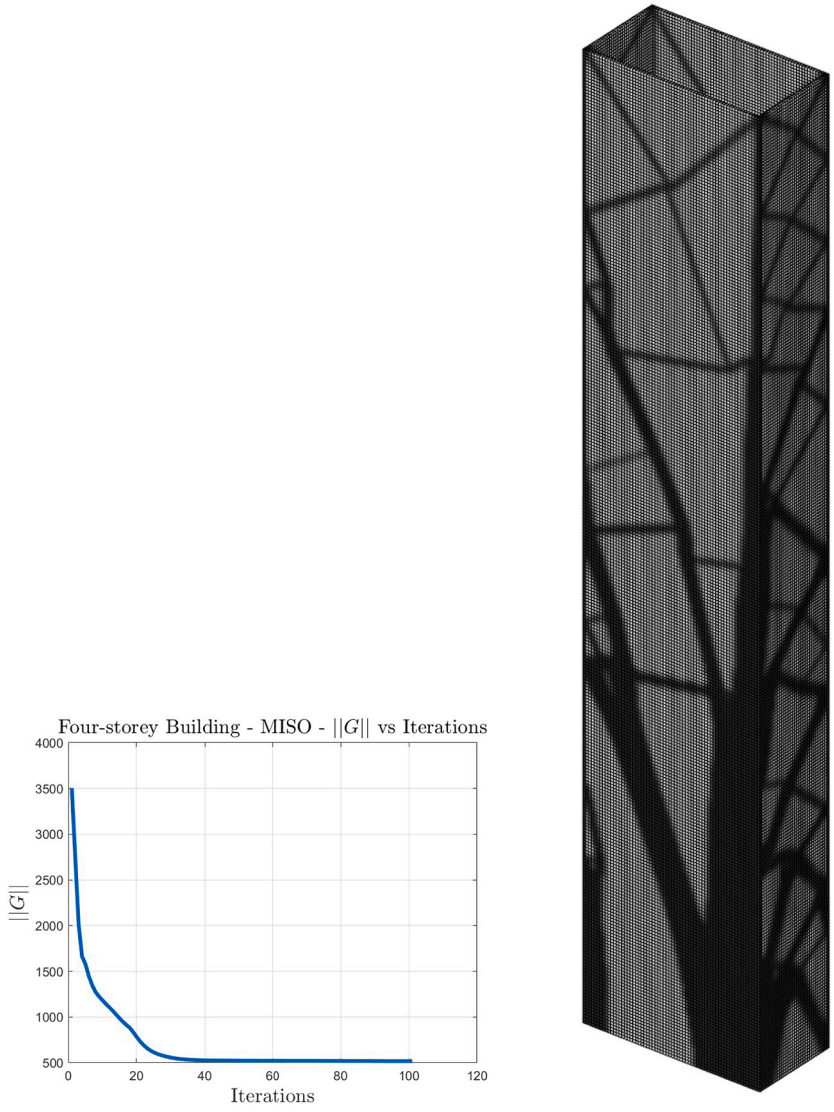

Fig. 17. Tall building - MISO and SIMO formulations - Convergence curve and 3D view.

inertia that enters the resisting mechanism to F_x, i.e. the load in the x direction.

Singular values that are stored in the diagonal matrix $\mathbf{\Sigma}$ deserve a few considerations of their own. The bottom line here is the fact that Nuclear and Frobenius norms depend on both the singular values σ_1 and σ_2 whereas the 2-norm depends and actually coincides with σ_1 only. Nuclear and Frobenius designs are capable to grade the singular values in such a way that, as expected, $\sigma_1 >> \sigma_2$. In both cases, a strong bracing system naturally arises in the weak (z) direction as shown in Figs. 12 and 13 for the Nuclear design and in Figs. 14 and 15 for the Frobenius one. The bracing system in the strong (x) direction happens to be different and is constituted by two tapered columns connected by many fewer braces than the ones used in the z direction. Both designs are convincing ones and aligned, so to speak, with the good engineering practice.

The design based on the minimization of the 2-norm shows a few severe limitations that were partially anticipated early on. By taking a look to Fig. 16, one may see that not only was the convergence path extremely slow, but also convergence was not actually reached after 1000 iterations (whereas 100 iterations were enough to achieve convergence using the Nuclear and Frobenius norms). This should be considered a general drawback of minimizing the 2-norm of an uncoupled Multi-Input system: focusing on σ_1 only means that only one of the inputs is being taken care of (things would be different and better in the case of coupled systems wherein the largest singular value σ_1 would be representative of all the input-output channels). The minimization of σ_1, with no care for any input-output pair but the one that gives rise to the 2-norm, causes σ_2 to grow until it reaches the value of σ_1. A competition between the two then starts that has the effect of dramatically slowing down convergence. A further undesired side effect of this issue, is the fact that σ_2 cannot decrease as much as it would (and it does) when using goal functions that depend on both σ_1 and σ_2 as is the case of Nuclear and Frobenius design.

That said, a synthesis of all the available designs may be gathered thanks to Tables 4 and 5. Table 4 shows that Nuclear and Frobenius norms lead to quite satisfactory designs wherein each of the two norms is minimized by the design that was actually aiming to the goal. Not so for the 2-norm that is not even capable to attain the overall minimum of the 2-norm itself (the Frobenius design is shown to be better in this respect). This is again due to the jeopardizing effect of designing an uncoupled MIMO system by minimizing a function that depends on a single plant direction. Table 5 grades the designs according to the same performance index introduced for the cantilevered beam problem. The inadequacy of the 2-norm to handle this kind of problems clearly emerges also from this point of view.

Table 4
Tall building - Norm competition between the three available designs.

	$\|\|\mathscr{C}\|\|_2$	$\|\|\mathscr{C}\|\|_N$	$\|\|\mathscr{C}\|\|_F$
2-norm design	608.03	1211.54	856.69
Nuclear norm design	633.70	**907.17**	690.19
Frobenius norm design	**604.10**	922.39	**682.82**

Table 5
Tall building - Overall performance of the designs (the smaller, the better).

	10^2 Performance Index PI
2-norm design	43.32
Nuclear norm design	4.93
Frobenius norm design	2.81

$$PI = \Big[\big(||\mathscr{C}||_2 - \min||\mathscr{C}||_2\big)^2 / \big(\min||\mathscr{C}||_2\big)^2 + \big(||\mathscr{C}||_N - \min||\mathscr{C}||_N\big)^2 / \big(\min||\mathscr{C}||_N\big)^2 + \big(||\mathscr{C}||_F - \min||\mathscr{C}||_F\big)^2 / \big(\min||\mathscr{C}||_F\big)^2\Big]^{1/2}.$$

7.3.4. MISO and SIMO formulations

Both the MISO and SIMO formulations are in a sense coarser models than the MIMO one in that the relevant transfer matrix $\boldsymbol{G}$ has rank 1 only, and is therefore governed by the single non-trivial singular value σ_1. All the three norms are therefore the same. Interestingly enough, however, assuming a single input or a single output leads to a coupling between the directions of the load sets x and z and in fact the matrix factors of the singular value decomposition at convergence read:

- MISO model

$$\boldsymbol{V} = \begin{bmatrix} -0.571 & 0.821 \\ -0.821 & -0.571 \end{bmatrix}, \ \boldsymbol{U} = 1$$
$$\boldsymbol{\Sigma} = \begin{bmatrix} 517.19 & 0 \end{bmatrix} \tag{45}$$

- SIMO model

$$\boldsymbol{V} = 1, \ \boldsymbol{U} = \begin{bmatrix} -0.571 & 0.821 \\ -0.821 & -0.571 \end{bmatrix}$$
$$\boldsymbol{\Sigma} = \begin{bmatrix} 517.19 \\ 0 \end{bmatrix} \tag{46}$$

Thanks to this coupling between the directions in the horizontal plane operated by the singular value decomposition the numerical optimization code does not get trapped into the same competition experienced in the MIMO case when using a 2 norm. One in fact ends up with the design shown in Fig. 17 that, though not as regular as the MIMO ones using Nuclear and Frobenius norms, has clearly captured the spatial nature of the problem and should be considered a satisfactory one, see also Fig. 18.

7.3.5. SISO formulation

The resisting mechanism of the SISO formulation is quite a peculiar one as shown in Figs. 19 and 20. Its main features are as follows:

- The bracing system in the weak plane is similar to the one we got using a MIMO formulation coupled to Nuclear or Frobenius norms and should therefore be considered quite efficient;
- being $\boldsymbol{G}$ a scalar in this SISO model, no competition arises between gains as witnessed by the convergence curve in Fig. 19 that is a smooth and nearly monotonic one;
- as for the resisting mechanism in the strong plane, it is basically not directly addressed by the computations that are only concerned with the plant directions $(\boldsymbol{v}_1, \boldsymbol{u}_1)$. As a result, no bracing system arises but the resisting mechanism consists of two strongly tapered perimeter columns connected by two horizontal struts positioned roughly at one thirds and two thirds of the building height;
- the overall mechanism is eventually a suitable one even though, as already said, the MIMO designs one gets using Nuclear or Frobeniu norms are the ones to be used.

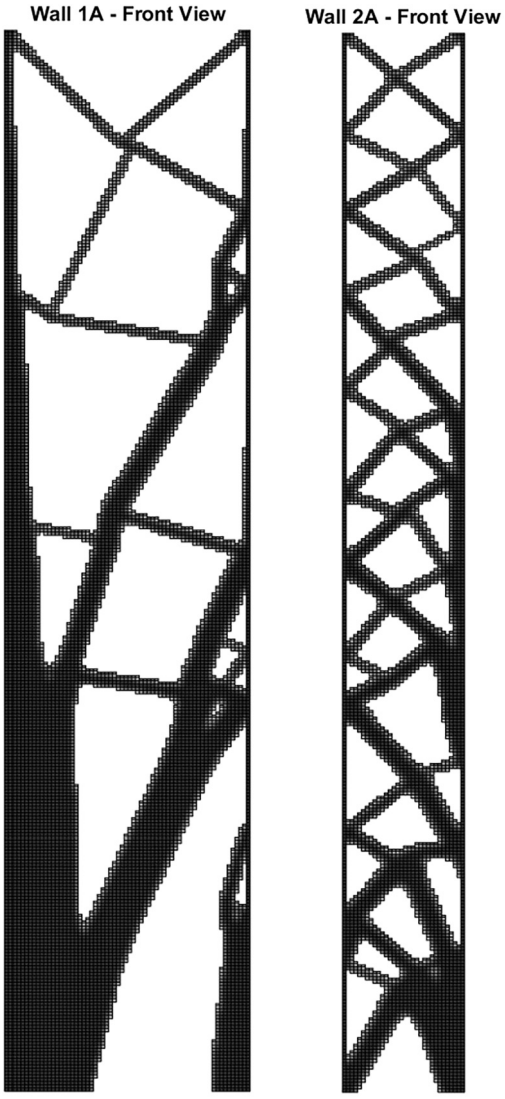

Fig. 18. Tall building - MISO and SIMO formulations - Sides 1A and 2A.

7.4. Problem 3 - L-shaped problem

7.4.1. Investigating the role of the adopted filter

The classical L-shaped problem is tackled next, see Fig. 21. The problem is of course interesting *per se*, but for the sake of this paper it is used to assess the efficiency of the proposed approach with respect to the choice of the filters used to avoid checkerboard patterns and the like. In what follows the dependence of the optimal solutions is investigated with respect to:

1. the choice of density vs sensitivity filters, [12];
2. the so-called filter radius r_{min}, [12].

Further possibilities such as using filters based on the solution of a (3D) Helmholtz type equation are left to future communications.

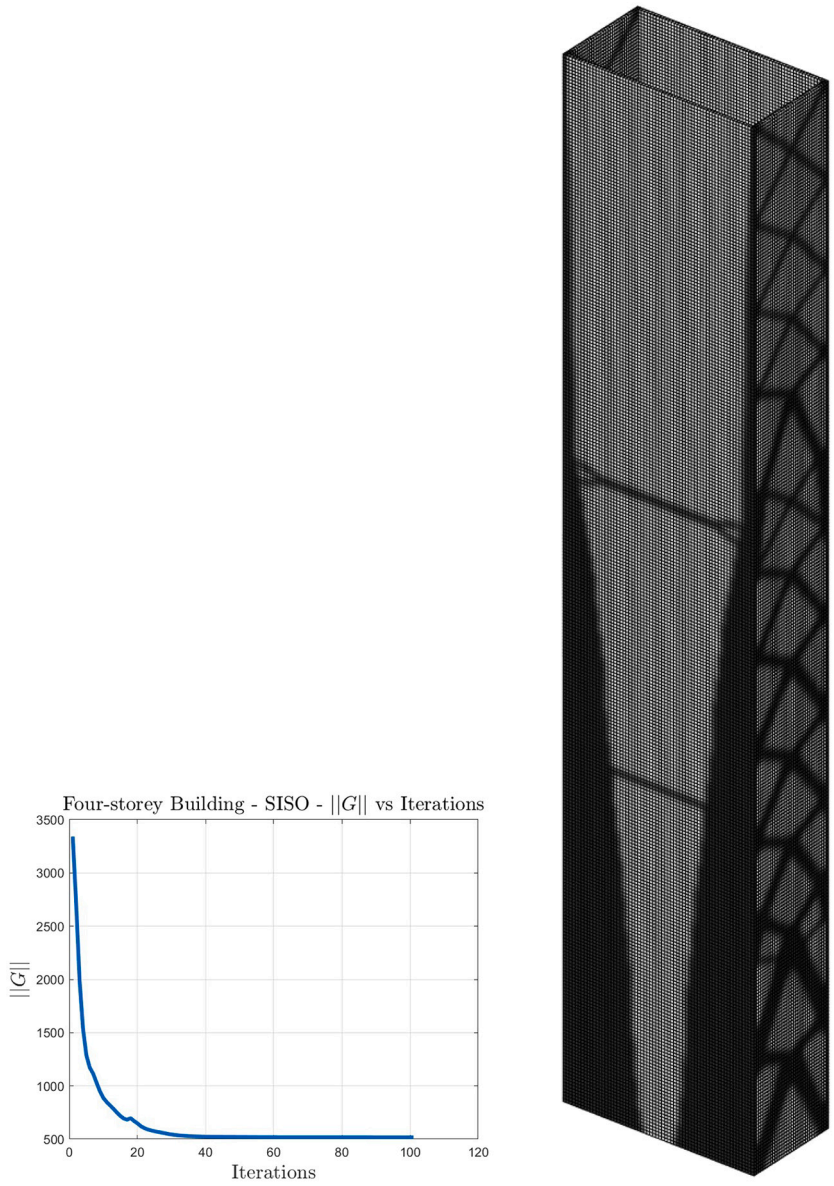

Fig. 19. Tall building - SISO formulation - Convergence curve and 3D view.

7.4.2. Mechanical parameters

All the simulations to follow share the following geometrical and mechanical data (see Fig. 21):

- `nx1` = 32, `nx2` = 64;
- `ny1` = 32, `ny2` = 64;
- `nz` = 5;
- `volfrac` = 0.45 (maximum volume fraction available for material);
- `E0` = 1 (Nominal Young modulus);
- `Emin` = 1e-9 (Minimum (void) Young modulus);
- `nu` = 0.3 (Constant Poisson ratio);
- `penal` = 3 (SIMP penalization factor).

The six cases in Table 6 are investigated next that differ from one another for the filter type and the filter radius.

7.4.3. Optimal topologies and convergence curves

Fig. 22 shows the optimal topologies for all the cases in Table 6. Sensitivity and density filters solutions are shown in the left and right column, respectively. The filter radius increases top-down assuming values 2.5, 4.0 and 5.5. The general lessons learned by these results are the following:

Table 6
L-shaped simulations: labeling with respect to filter and radius.

Simulation	Filter type	r_{min}
1l	Sensitivity	2.5
1r	Density	2.5
2l	Sensitivity	4.0
2r	Density	4.0
3l	Sensitivity	5.5
3r	Density	5.5

- the sensitivity filter is more stable in that the optimal topologies (left column) depend only weakly on the choice of the radius r_{min}. On the other side, solutions based on the density filter exhibit a remarkable dependence on the filter radius r_{min}.
- Solutions based on sensitivity filters are characterized by beam-like components only, whereas bulky structural components show up when using density filters.

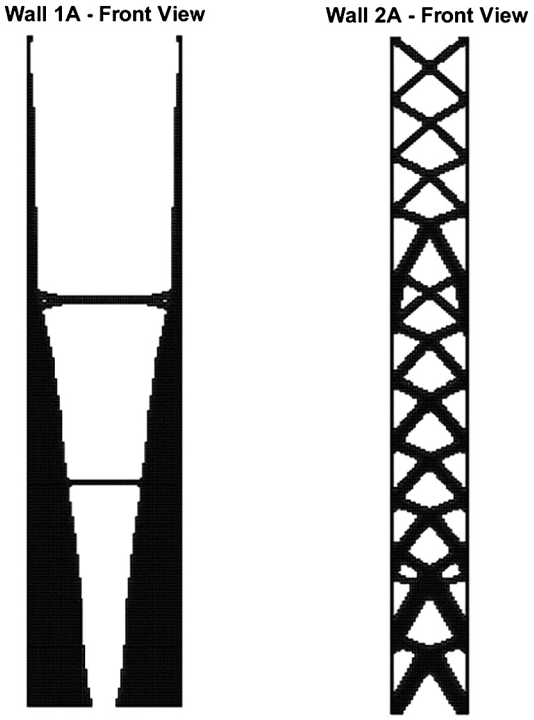

Fig. 20. Tall building - SISO formulation - Sides 1A and 2A.

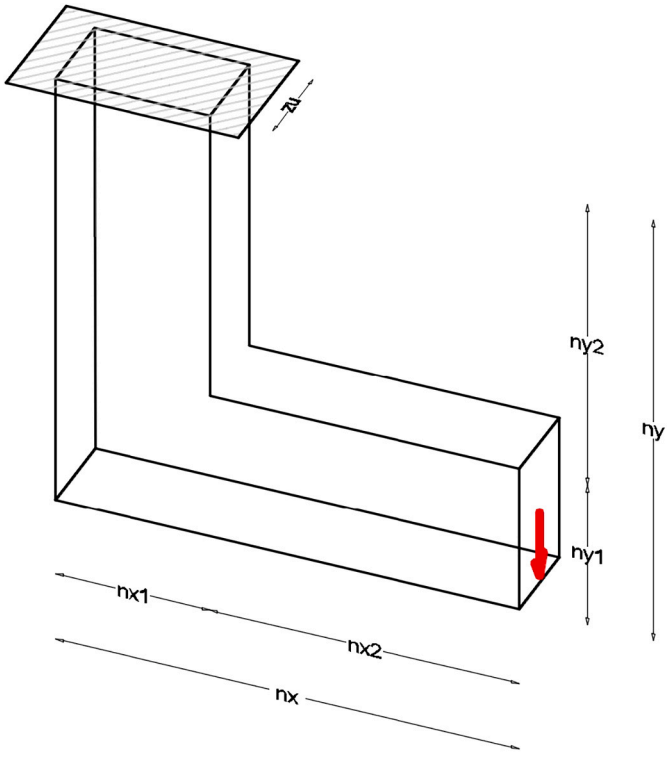

Fig. 21. L-shaped structure - The engineering problem.

- Overall, the proposed method is robust with respect to the filter in that the optimal topology is uniquely defined, even though a few differences may be seen.

As for convergence curves, a look to Fig. 23 allows to conclude that both filters lead to a monotonically decreasing path toward convergence, that is even smoother when using density filters.

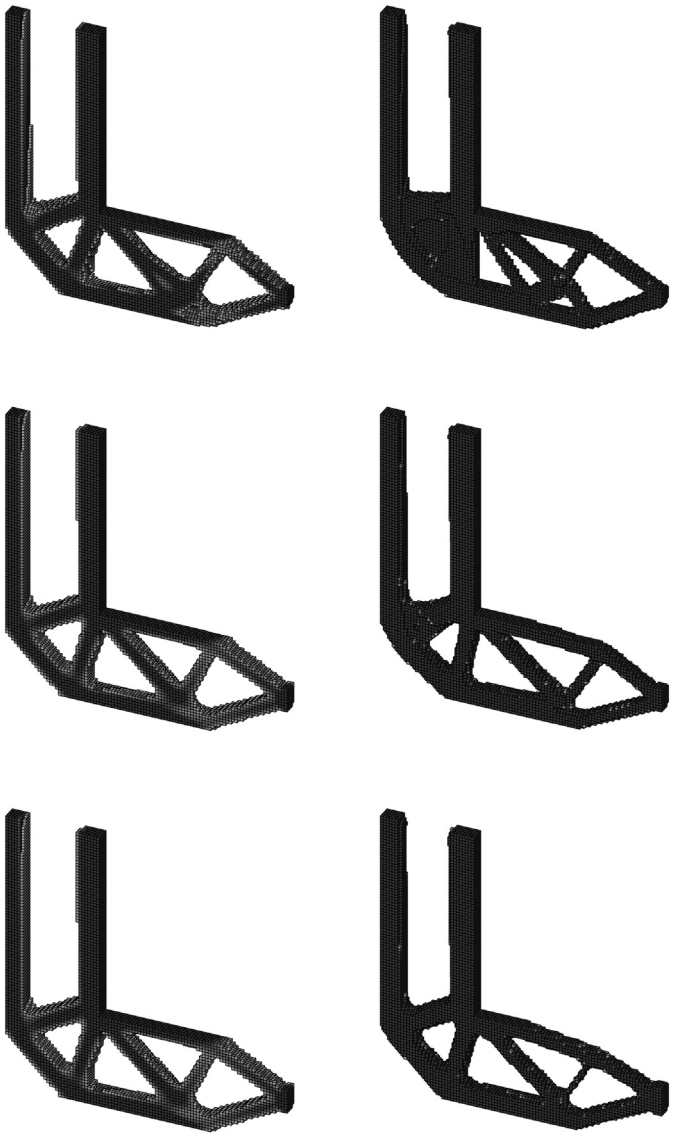

Fig. 22. L-shaped problem - Optimal Topologies. Left column: sensitivity filter - Right column: density filter. Top row: $r_{min} = 2.5$, Mid row: $r_{min} = 4.0$, Bottom row: $r_{min} = 5.5$.

8. Conclusions and need for further investigations

A new approach has been presented for the topology optimization of 3D systems and structures that is based on the singular value decomposition of the input-output transfer matrix, say $\boldsymbol{G}$. Far from having a purely algebraic meaning only, the three matrix factors that define the singular value decomposition of $\boldsymbol{G}$ are shown to carry a profound engineering significance. Right and left singular vectors were in fact shown to define the plant directions along which the blow up factors or gains are retained by the associated singular values. This clear physical interpretation is the milestone on which the proposed approach rests that is given numerical substance by the choice as goal functions of proper matrix uniquely defined by the singular values themselves. Extensive numerical results were presented to validate the theoretical and numerical frameworks. They are concerned with a cantilevered beam structure and a tall building. To model the latter, a Lagrange multiplier technique has been adopted to link orthogonal ones to one another. An in depth analysis of the results has been presented and a few guidelines were given as for the selection of the system idealization (SISO, MISO, SIMO and MIMO) and the matrix norm (2, Nuclear and Frobenius) to be adopted according to the specific goal of the design.

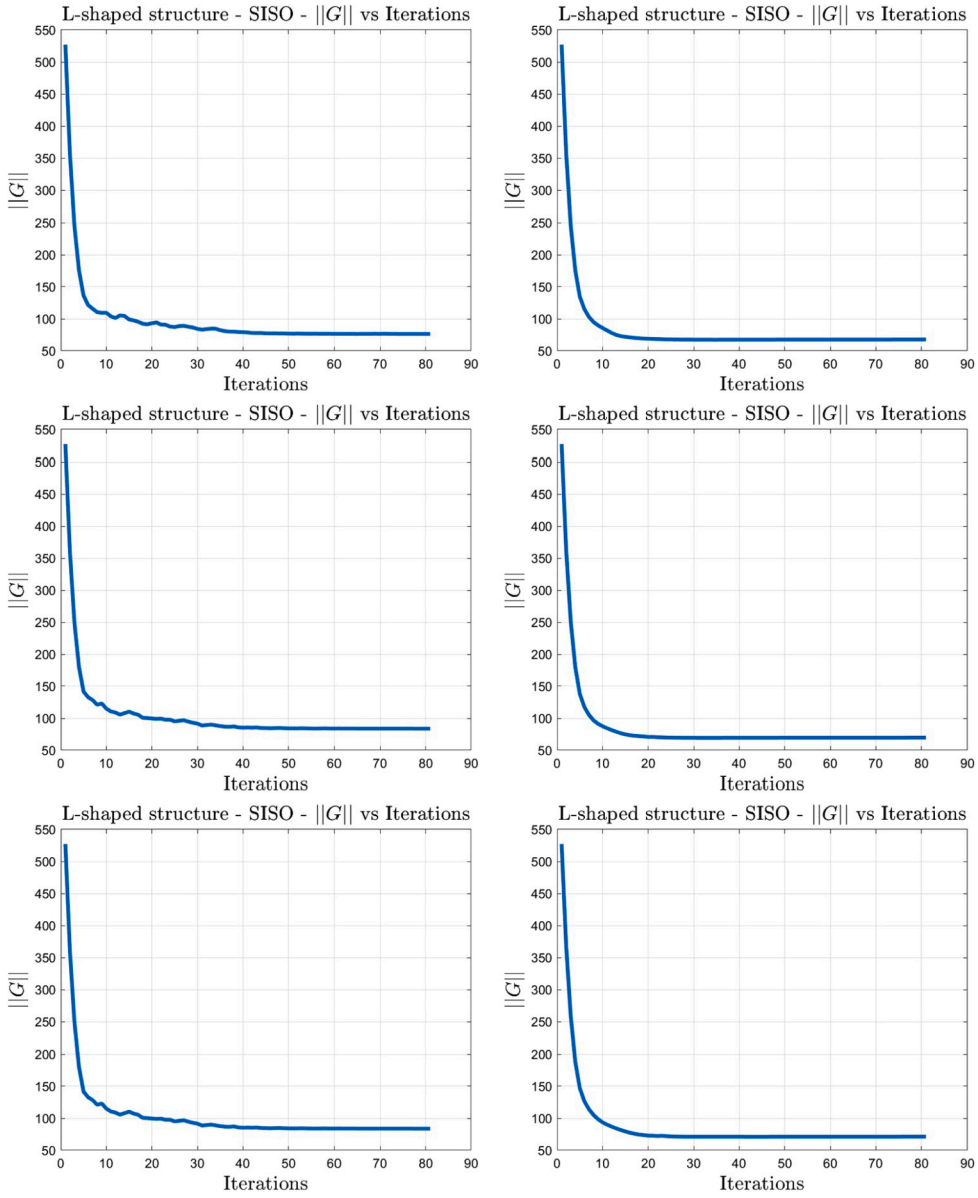

Fig. 23. L-shaped problem - Convergence curves. Left column: sensitivity filter - Right column: density filter. Top row: $r_{min} = 2.5$, Mid row: $r_{min} = 4.0$, Bottom row: $r_{min} = 5.5$.

Ongoing and near future planned extensions include the following ones:

1. the derivation and use of reduced order models to speed up the computations. This seems to be rather straightforward as far as the evaluation of the goal functions (matrix norms) is concerned. Not so for the sensitivities since reduced order models lead to a loss of the physical meaning of the reduced-order state variables;
2. the method seems prone to be applied to dynamic problems framed in the frequency domain. As a matter of fact, the concepts of transfer matrix and matrix norm (that are the two pillars on which the proposed approach is founded) can be (and have already been) defined in the dynamic regime as well [3];
3. though numerical solutions appear quite satisfactory, it is planned to extend the formulation to handle additive manufacturing constraints. This would allow a practical realization of the designed specimen that is the natural and final goal of any 3D design procedure;
4. reference made to the tall-building problem, the proposed brace-design methodology shall be used to design an optimal bracing reinforcement for existing framed structures within the framework of the exoskeleton reinforcement approach (see [24] among others).

The full Matlab code solving the tall-building problems is provided in Appendix A. The only missing part is the Matlab version of the MMA code [1] that should be asked to Krister Svanberg (that is herein gratefully acknowledged for providing it to this author). The cantilevered problem does not call for any Lagrange multiplier strategy and may therefore be addressed by properly simplifying the provided Matlab code.

Declarations

Replication of results

The full Matlab code solving the 3D static topology optimization problem is given below.

CRediT authorship contribution statement

P. Venini: Conceptualization, Data curation, Funding acquisition, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

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

Appendix A. Matlab code

```
%% CLEAN ALL
clear all
close all
fclose('all')
clc

global ITYPE INORM FX FZ UU VV SS Bstate Cstate passive active BT G A

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

tic

%% PROBLEM TYPE
%ITYPE = 1; s1 = 'SISO_'; %SISO
%ITYPE = 2; s1 = 'MISO_'; %MISO
%ITYPE = 3; s1 = 'SIMO_'; %SIMO
ITYPE = 4;  s1 = 'MIMO_'; %MIMO

%% NORM TYPE
%INORM = 1; s2 = '2_'; %2 NORM
INORM = 2;  s2 = 'NUC_'; %NUCLEAR NORM
%INORM = 3; s2 = 'FRO_'; %FROBENIUS NORM

s_intro = strcat(s1,s2);

%% MATERIAL AND OTHER PROPERTIES
length_x = 1; length_y = 1; length_z = 1;
volfrac = 0.35;
penal = 3.0;
ft = 2;
displayflag = 1;        % Display structure flag
%
nelx1 = 2^5;
rmin = 2.25;
nstoreys = 4; nely_s = ceil(nelx1*1.25);
nely = nstoreys*nely_s;
nely1 = nely; nelz1 = 1;
nelx2 = nelz1; nely2 = nely; nelz2 = nelx1/2-2*nelz1; % nelz2 = nelx1-2*nelz1; %value square building
%
nele1 = nelx1*nely1*nelz1;
nnode1 = (nelx1+1)*(nely1+1)*(nelz1+1);
ndof1 = 3*nnode1;
%
nele2 = nelx2*nely2*nelz2;
nnode2 = (nelx2+1)*(nely2+1)*(nelz2+1);
ndof2 = 3*nnode2;

% USER-DEFINED LOOP PARAMETERS
displayflag = 1;  % Display structure flag

% USER-DEFINED MATERIAL PROPERTIES
E0 = 10^2;           % Young's modulus of solid material
Emin = 1e-6;      % Young's modulus of void-like material
nu = 0.3;         % Poisson's ratio

%% WALL P1A
nodegrd = reshape(1:(nely1+1)*(nelx1+1),nely1+1,nelx1+1);
nodeids = reshape(nodegrd(1:end-1,1:end-1),nely1*nelx1,1);
nodeidz = 0:(nely1+1)*(nelx1+1):(nelz1-1)*(nely1+1)*(nelx1+1);
nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
```

```
edofVec = 3*nodeids(:)+1;
edofMat1a = repmat(edofVec,1,24)+ ...
    repmat([0 1 2 3*nely1 + [3 4 5 0 1 2] -3 -2 -1 ...
    3*(nely1+1)*(nelx1+1)+[0 1 2 3*nely1 + [3 4 5 0 1 2] -3 -2 -1]],nele1,1);
iK1a = reshape(kron(edofMat1a,ones(24,1))',24*24*nele1,1);
jK1a = reshape(kron(edofMat1a,ones(1,24))',24*24*nele1,1);
x1a = repmat(volfrac,[nely1,nelx1,nelz1]);
xPhys1a = x1a;
display_3D_shift2(xPhys1a,0,0,0);
% USER-DEFINED SUPPORT FIXED DOFs
[iif,jf,kf] = meshgrid(0:nelx1,0,0:nelz1);                  % Coordinates
fixednid1a = kf*(nelx1+1)*(nely1+1)+iif*(nely1+1)+(nely1+1-jf); % Node IDs
fixeddof1a = [3*fixednid1a(:); 3*fixednid1a(:)-1; 3*fixednid1a(:)-2]; % DOFs
% USER-DEFINED LOAD DOFsX x=0
loadnid1a = 1:nely_s:nely_s*nstoreys;
%loadnid1a=nely_s:nely_s:nely1; % Node IDs
loaddof1ax = 3*loadnid1a(:) - 2;
loaddof1az = 3*loadnid1a(:);   % DOFs
%%
% USER-DEFINED LOAD DOFsX x=nelx1
loadnid1aa=loadnid1a+(nelx1)*(nely1+1); % Node IDs
loaddof1aax = 3*loadnid1aa(:) - 2;      % DOFs
loaddof1aaz = 3*loadnid1aa(:);

% PASSIVE ELEMENTS (corner columns)
passive = [];
for ii=1:nelx2
    passive = [passive (ii-1)*nely1+1:ii*nely1];
end
passive_ = passive;
for ii=2:nelz1
    passive = [passive passive_+(ii-1)*nelx1*nely1];
end
passive = [passive passive+nely1*(nelx1-nelx2)];

%% WALL P1B
edofMat1b = edofMat1a + ndof1;
iK1b = reshape(kron(edofMat1b,ones(24,1))',24*24*nele1,1);
jK1b = reshape(kron(edofMat1b,ones(1,24))',24*24*nele1,1);
x1b = repmat(volfrac,[nely1,nelx1,nelz1]);
xPhys1b = x1b;
display_3D_shift2(xPhys1b,0,0,nelz1+nelz2);
% USER-DEFINED SUPPORT FIXED DOFs
fixednid1b = fixednid1a + nnode1; % Node IDs
fixeddof1b = [3*fixednid1b(:); 3*fixednid1b(:)-1; 3*fixednid1b(:)-2]; % DOFs
% USER-DEFINED LOAD DOFsX x=0
loadnid2a=loadnid1a+nnode1; % Node IDs
loaddof2ax = 3*loadnid2a(:) - 2;
loaddof2az = 3*loadnid2a(:);   % DOFs
% USER-DEFINED LOAD DOFsX x=nelx1
loadnid2aa=loadnid2a+(nelx1)*(nely1+1); % Node IDs
loaddof2aax = 3*loadnid2aa(:) - 2;
loaddof2aaz = 3*loadnid2aa(:);
% PASSIVE ELEMENTS (corner columns)
passive = [passive passive+nele1];

%% WALL P2A
nodegrd = reshape(1:(nely2+1)*(nelx2+1),nely2+1,nelx2+1);
nodeids = reshape(nodegrd(1:end-1,1:end-1),nely2*nelx2,1);
nodeidz = 0:(nely2+1)*(nelx2+1):(nelz2-1)*(nely2+1)*(nelx2+1);
nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
edofVec = 3*nodeids(:)+1;
edofMat2a = repmat(edofVec,1,24)+ ...
    repmat([0 1 2 3*nely2 + [3 4 5 0 1 2] -3 -2 -1 ...
    3*(nely2+1)*(nelx2+1)+[0 1 2 3*nely2 + [3 4 5 0 1 2] -3 -2 -1]],nele2,1)+max(max(edofMat1b));
iK2a = reshape(kron(edofMat2a,ones(24,1))',24*24*nele2,1);
jK2a = reshape(kron(edofMat2a,ones(1,24))',24*24*nele2,1);
x2a = repmat(volfrac,[nely2,nelx2,nelz2]);
xPhys2a = x2a;
display_3D_shift2(xPhys2a,0,0,nelz1);
% USER-DEFINED SUPPORT FIXED DOFs
[iif,jf,kf] = meshgrid(0:nelx2,0,0:nelz2);                  % Coordinates
fixednid2a = kf*(nelx2+1)*(nely2+1)+iif*(nely2+1)+(nely2+1-jf); % Node IDs
fixednid2a = fixednid2a + 2*nnode1;
fixeddof2a = [3*fixednid2a(:); 3*fixednid2a(:)-1; 3*fixednid2a(:)-2]; % DOFs
ndof3 = max(max(edofMat2a));

%% WALL P2B
edofMat2b = edofMat2a + ndof2;
iK2b = reshape(kron(edofMat2b,ones(24,1))',24*24*nele2,1);
```

```
jK2b = reshape(kron(edofMat2b,ones(1,24))',24*24*nele2,1);
x2b = repmat(volfrac,[nely2,nelx2,nelz2]);
xPhys2b = x2b;
display_3D_shift2(xPhys2b,nelx1-nelx2,0,nelz1);
% USER-DEFINED SUPPORT FIXED DOFs
fixednid2b = fixednid2a + nnode2; % Node IDs
fixeddof2b = [3*fixednid2b(:); 3*fixednid2b(:)-1; 3*fixednid2b(:)-2]; % DOFs

%SOME FURTHER GLOBAL PARAMETERS
edofMat = [edofMat1a; edofMat1b; edofMat2a; edofMat2b];
nele = size(edofMat,1);
ndof = max(max(edofMat));
fixeddofs = [fixeddof1a; fixeddof1b; fixeddof2a; fixeddof2b];
npassive = length(passive);
V0 = nele-npassive;
V01 = (nelx1-2*nelx2)*nely1*nelz1;
V02 = nelx2*nely2*nelz2;
active = setdiff(1:nele,passive);
nactive = length(active);

%STIFFNESS MATRIX INDEXES
iK = [iK1a; iK1b; iK2a; iK2b];
jK = [jK1a; jK1b; jK2a; jK2b];

%%
%KE = stiffnessMatrix_brick(1,nu,length_x,length_y,length_z);
KE = lk_H8(nu);

%% PREPARE FILTER (WALLS P1a and P1b)
iH = ones(nele1*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for k1 = 1:nelz1
    for i1 = 1:nelx1
        for j1 = 1:nely1
            e1 = (k1-1)*nelx1*nely1 + (i1-1)*nely1+j1;
            for k2 = max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-1),nelz1)
                for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx1)
                    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely1)
                        e2 = (k2-1)*nelx1*nely1 + (i2-1)*nely1+j2;
                        k = k+1;
                        iH(k) = e1;
                        jH(k) = e2;
                        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-k2)^2));
                    end
                end
            end
        end
    end
end
H1 = sparse(iH,jH,sH);
Hs1 = sum(H1,2);

%% PREPARE FILTER (WALLS P2a and P2b)
iH = ones(nele2*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for k1 = 1:nelz2
    for i1 = 1:nelx2
        for j1 = 1:nely2
            e1 = (k1-1)*nelx2*nely2 + (i1-1)*nely2+j1;
            for k2 = max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-1),nelz2)
                for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx2)
                    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely2)
                        e2 = (k2-1)*nelx2*nely2 + (i2-1)*nely2+j2;
                        k = k+1;
                        iH(k) = e1;
                        jH(k) = e2;
                        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-k2)^2));
                    end
                end
            end
        end
    end
end
H2 = sparse(iH,jH,sH);
Hs2 = sum(H2,2);

```

```
%% MATCHING NODES (LAGRANGE MULTIPLIERS)
% P1A-P2A
p1a = (nelx1+1)*(nely1+1)*nelz1;
node_p1a_p2a = p1a+1:p1a+(nely1+1)*(nelx2+1);
p2a = 2*nnode1;
node_p2a_p1a = p2a+1:p2a+(nely1+1)*(nelx2+1);
% P2A-P1B
p1b = nnode1;
node_p1b_p2a = p1b+1:p1b+(nely1+1)*(nelx2+1);
p2a = (nelx2+1)*(nely2+1)*nelz2+2*nnode1;
node_p2a_p1b = p2a+1:p2a+(nely1+1)*(nelx2+1);
% P1A-P2B
p1a = (nelx1+1)*(nely1+1)*(nelz1+1);
node_p1a_p2b = sort(p1a:-1:p1a-(nelx2+1)*(nely1+1)+1);
p2b = 2*nnode1+nnode2;
node_p2b_p1a = p2b+1:p2b+(nely1+1)*(nelx2+1);
% P2B-P1B
p1b = (nelx1+1)*(nely1+1)+nnode1;
node_p1b_p2b = sort(p1b:-1:p1b-(nelx2+1)*(nely1+1)+1);
p2b = (nelx2+1)*(nely2+1)*nelz2+2*nnode1+nnode2;
node_p2b_p1b = p2b+1:p2b+(nely1+1)*(nelx2+1);
%% ALL MULTIPLIERS
leftnid = [node_p1a_p2a node_p1b_p2a node_p1a_p2b node_p1b_p2b]';
rightnid = [node_p2a_p1a node_p2a_p1b node_p2b_p1a node_p2b_p1b]';
leftdofs = [3*leftnid(:); 3*leftnid(:)-1; 3*leftnid(:)-2];    % DOFs
rightdofs = [3*rightnid(:); 3*rightnid(:)-1; 3*rightnid(:)-2];   % DOFs
leftdofs = leftdofs(ismember(leftdofs,fixeddofs)==0);
rightdofs = rightdofs(ismember(rightdofs,fixeddofs)==0);
nLag = length(leftdofs);
nAll = ndof + nLag;

%% BT MATRIX
iB1 = [(1:nLag)'; (1:nLag)'];
jB1 = [leftdofs; rightdofs];
sB1 = [ones(length(leftdofs),1) -ones(length(leftdofs),1)];
BT = sparse(iB1,jB1,sB1,nLag,ndof);

%% DEFINE LOADS AND SUPPORTS (EXAMPLE 1)
FX = [loaddof1ax; loaddof1aax; loaddof2ax; loaddof2aax];
FZ = [loaddof1az; loaddof1aaz; loaddof2az; loaddof2aaz];
if ITYPE == 1
    F = sparse([FX; FZ],1,[zeros(size(FX)); ones(size(FZ))],nAll,1);
elseif ITYPE == 2
    F = sparse([FX; FZ],[ones(length(FX),1); 2*ones(length(FZ),1)],1,nAll,2);
elseif ITYPE == 3
    F = sparse([FX; FZ],1,1,nAll,1);
elseif ITYPE == 4
    F = sparse([FX; FZ],[ones(length(FX),1); 2*ones(length(FZ),1)],1,nAll,2);
else
    'System type unknown'
    exit
end

%% PREPARE FINITE ELEMENT ANALYSIS
freedofs = setdiff(1:ndof,fixeddofs)';
freedofs2 = setdiff(1:nAll,fixeddofs)';

%% INITIALIZE: UNIFORM INITIAL GUESS (but could be the last design from a previous run or else)
x1 = volfrac*ones(nele,1);
x1(passive) = 1;
xPhys1 = x1(active);
xPhysPrint1 = x1;

%% INITIALIZE 2: MMA PARAMETERS
initialize_mma_TALL;

%% START MMA AND GET THE OPTIMAL DESIGN
start_mma_TALL;

toc
```

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

global outeriter

m = 4;          %number of constraints
n = nele;       %number of design variables
epsimin = 1e-6;
```

```
eeen    = ones(n,1);
eeem    = ones(m,1);
zeron   = zeros(n,1);
zerom   = zeros(m,1);
xval    = [x1(:)];
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  = 100;
kkttol  = 5*10^(-4);
```

```
%% ITERATION "0"
if outeriter < 0.5
    [f0val,df0dx,fval,dfdx,xPhys] = compliance_stat_TALL(xval,H1,Hs1,H2,Hs2,E0,Emin,...
        penal,ft,V0,V01,V02,volfrac,KE,iK,jK,F,freedofs2,edofMat,nele1,nele2,nele,nAll,m);
    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;
    %xval(find(passive)) = 0;
    [f0val,df0dx,fval,dfdx,xPhys] = compliance_stat_TALL(xval,H1,Hs1,H2,Hs2,E0,Emin,...
        penal,ft,V0,V01,V02,volfrac,KE,iK,jK,F,freedofs2,edofMat,nele1,nele2,nele,nAll,m);
    xsave = [xsave xval(:)];
    nsave = [nsave; f0val];
    goalsave = [goalsave; f0val];
    %% 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(xPhys(active)));
    % SAVE HISTORY
    icheck = rem(outit,5);
    if icheck==0
        % PLOT DENSITIES
        if displayflag, clf;
            x1a = reshape(xPhys(1:nele1),[nely1,nelx1,nelz1]);
            display_3D_shift2(x1a,0,0,0);
            %
            x1b = reshape(xPhys(nele1+1:2*nele1),[nely1,nelx1,nelz1]);
            display_3D_shift2(x1b,0,0,nelz1+nelz2);
            %
            x2a = reshape(xPhys(2*nele1+1:2*nele1+nele2),[nely2,nelx2,nelz2]);
            display_3D_shift2(x2a,0,0,nelz1);
            %
            x2b = reshape(xPhys(2*nele1+nele2+1:2*nele1+2*nele2),[nely2,nelx2,nelz2]);
            display_3D_shift2(x2b,nelx1-nelx2,0,nelz1);
        end
        snum = num2str(outit);
        filename = strcat(s_intro,snum);
        filemat = strcat('x_TALL_ST4',s_intro);
        save(filename)
        save([filemat,num2str(outit,'%03d'),'.mat'], 'xPhys');
    end
end

```

```
PRINT_WALLS

nsave(4) = nsave(4)/10;
XITER = 1:length(nsave);
figure, plot(XITER,nsave,'LineWidth',3)
xlabel('Iterations','fontsize',16,'interpreter','latex')
ylabel('$||G||_2$','fontsize',16,'interpreter','latex')
title('Four-storey Building - MIMO - $||G||_2$ vs Iterations','fontsize',16,'interpreter','latex')
grid
print -djpeg CONVERGENCE
```

```
function [c,dcc,v,dvv,xPhys] = compliance_stat_TALL(x,H1,Hs1,H2,Hs2,E0,Emin,penal,ft,V0,V01,V02,volfrac,...
    KE,iK,jK,F,freedofs2,edofMat,nele1,nele2,nele,nAll,m)

global outeriter Bstate Cstate ITYPE INORM FX FZ UU VV SS passive BT G A

%% filtering densities
if ft == 1
    xPhys(:) = x(:);
elseif ft == 2
    x1a = x(1:nele1);
    x1b = x(nele1+1:2*nele1);
    x2a = x(2*nele1+1:2*nele1+nele2);
    x2b = x(2*nele1+nele2+1:2*nele1+2*nele2);
    xPhys(1:nele1) = (H1*x1a(:))./Hs1;
    xPhys(nele1+1:2*nele1) = (H1*x1b(:))./Hs1;
    xPhys(2*nele1+1:2*nele1+nele2) = (H2*x2a(:))./Hs2;
    xPhys(2*nele1+nele2+1:2*nele1+2*nele2) = (H2*x2b(:))./Hs2;
end
xPhys(passive) = 1;
xPhys = xPhys(:);

%% 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([FX; FZ],1,-1,nAll,1)]';
    elseif ITYPE == 3
        Cstate = [sparse([FX; FZ],[ones(length(FX),1); 2*ones(length(FZ),1)],-1,nAll,2)]';
    end
    Bstate = Bstate(freedofs2,:);
    Cstate = Cstate(:,freedofs2);
end

%% updated stiffness matrix
sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),24*24*nele,1);
K = sparse(iK,jK,sK); K = (K+K')/2;

A = [K BT'
    BT zeros(size(BT,1),size(BT,1))];

%% factors for sensitivities 1
FSX = Cstate/A(freedofs2,freedofs2);
FDX = A(freedofs2,freedofs2)\Bstate;

%% transfer matrix
G = Cstate*FDX;

%% svd
[UU,SS,VV] = svds(G,size(G,2));
[S_SS,S_II] = sort(diag(SS),'descend');
UU = UU(:,S_II); VV = VV(:,S_II);
SS = diag(S_SS);

%% factors for sensitivities
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(nele,1);

parfor jj = 1:nele
    %% stiffness matrix derivative
    kDer = sparse(size(A,1),size(A,2));
    kDer(edofMat(jj,:),edofMat(jj,:)) = penal*xPhys(jj)^(penal-1)*(E0-Emin)*KE;
    kDer = kDer(freedofs2,freedofs2);
    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);
dc = NormGrad;

% constraints
v1A = 1:nele1; v1B = nele1+1:2*nele1;
v2A = 2*nele1+1:2*nele1+nele2; v2B = 2*nele1+nele2+1:2*nele1+2*nele2;
xP1A = v1A(ismember(1:nele1,passive)==0);
xP1B = v1B(ismember(nele1+1:2*nele1,passive)==0);
xP2A = v2A(ismember(2*nele1+1:2*nele1+nele2,passive)==0);
xP2B = v2B(ismember(2*nele1+nele2+1:2*nele1+2*nele2,passive)==0);
v1 = sum(xPhys(xP1A))/V01-volfrac;
v2 = sum(xPhys(xP1B))/V01-volfrac;
v3 = sum(xPhys(xP2A))/V02-volfrac;
v4 = sum(xPhys(xP2B))/V02-volfrac;
v = [v1; v2; v3; v4];
%
dv1 = zeros(nele1,1);
dv2 = zeros(nele1,1);
dv3 = zeros(nele2,1);
dv4 = zeros(nele2,1);
dv1(ismember(1:nele1,passive)==0) = ones(length(xP1A),1)/V01;
dv2(ismember(nele1+1:2*nele1,passive)==0) = ones(length(xP1B),1)/V01;
dv3(ismember(2*nele1+1:2*nele1+nele2,passive)==0) = ones(length(xP2A),1)/V02;
dv4(ismember(2*nele1+nele2+1:2*nele1+2*nele2,passive)==0) = ones(length(xP2B),1)/V02;

%% filtering/modification of sensitivities
if ft == 1
    dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
elseif ft == 2
    dc1a = dc(1:nele1);
    dc1b = dc(nele1+1:2*nele1);
    dc2a = dc(2*nele1+1:2*nele1+nele2);
    dc2b = dc(2*nele1+nele2+1:2*nele1+2*nele2);
    dc(1:nele1) = (H1*dc1a(:))./Hs1;
    dc(nele1+1:2*nele1) = (H1*dc1b(:))./Hs1;
    dc(2*nele1+1:2*nele1+nele2) = (H2*dc2a(:))./Hs2;
    dc(2*nele1+nele2+1:2*nele1+2*nele2) = (H2*dc2b(:))./Hs2;
%
    dv1(:) = H1*(dv1(:)./Hs1);
    dv2(:) = H1*(dv2(:)./Hs1);
    dv3(:) = H2*(dv3(:)./Hs2);
    dv4(:) = H2*(dv4(:)./Hs2);
end
dcc = dc(:);
dvv = zeros(m,nele);
dvv(1,1:nele1) = dv1(:)';
dvv(2,nele1+1:2*nele1) = dv2(:)';
dvv(3,2*nele1+1:2*nele1+nele2) = dv3(:)';
dvv(4,2*nele1+nele2+1:2*nele1+2*nele2) = dv4(:)';
```

```
% === GENERATE ELEMENT STIFFNESS MATRIX ===
function [KE] = lk_H8(nu)
A = [32 6 -8 6 -6 4 3 -6 -10 3 -3 -3 -4 -8;
```

```
     -48 0 0 -24 24 0 0 0 12 -12 0 12 12 12];
k = 1/144*A'*[1; nu];

K1 = [k(1) k(2) k(2) k(3) k(5) k(5);
     k(2) k(1) k(2) k(4) k(6) k(7);
     k(2) k(2) k(1) k(4) k(7) k(6);
     k(3) k(4) k(4) k(1) k(8) k(8);
     k(5) k(6) k(7) k(8) k(1) k(2);
     k(5) k(7) k(6) k(8) k(2) k(1)];
K2 = [k(9)  k(8)  k(12) k(6)  k(4)  k(7);
     k(8)  k(9)  k(12) k(5)  k(3)  k(5);
     k(10) k(10) k(13) k(7)  k(4)  k(6);
     k(6)  k(5)  k(11) k(9)  k(2)  k(10);
     k(4)  k(3)  k(5)  k(2)  k(9)  k(12)
     k(11) k(4)  k(6)  k(12) k(10) k(13)];
K3 = [k(6)  k(7)  k(4)  k(9)  k(12) k(8);
     k(7)  k(6)  k(4)  k(10) k(13) k(10);
     k(5)  k(5)  k(3)  k(8)  k(12) k(9);
     k(9)  k(10) k(2)  k(6)  k(11) k(5);
     k(12) k(13) k(10) k(11) k(6)  k(4);
     k(2)  k(12) k(9)  k(4)  k(5)  k(3)];
K4 = [k(14) k(11) k(11) k(13) k(10) k(10);
     k(11) k(14) k(11) k(12) k(9)  k(8);
     k(11) k(11) k(14) k(12) k(8)  k(9);
     k(13) k(12) k(12) k(14) k(7)  k(7);
     k(10) k(9)  k(8)  k(7)  k(14) k(11);
     k(10) k(8)  k(9)  k(7)  k(11) k(14)];
K5 = [k(1) k(2)  k(8)  k(3) k(5)  k(4);
     k(2) k(1)  k(8)  k(4) k(6)  k(11);
     k(8) k(8)  k(1)  k(5) k(11) k(6);
     k(3) k(4)  k(5)  k(1) k(8)  k(2);
     k(5) k(6)  k(11) k(8) k(1)  k(8);
     k(4) k(11) k(6)  k(2) k(8)  k(1)];
K6 = [k(14) k(11) k(7)  k(13) k(10) k(12);
     k(11) k(14) k(7)  k(12) k(9)  k(2);
     k(7)  k(7)  k(14) k(10) k(2)  k(9);
     k(13) k(12) k(10) k(14) k(7)  k(11);
     k(10) k(9)  k(2)  k(7)  k(14) k(7);
     k(12) k(2)  k(9)  k(11) k(7)  k(14)];
KE = 1/((nu+1)*(1-2*nu))*...
     [ K1  K2  K3  K4;
     K2'  K5  K6  K3';
     K3' K6  K5' K2';
     K4  K3  K2  K1'];
KE = 0.5*(KE+KE');
end
```

```
figure
x1a = reshape(xPhys(1:nele1),[nely1,nelx1,nelz1]);
display_3D_shift2_cam(x1a,0,0,0,0,30)
title('Wall 1A - Front View')
file1 = strcat(s_intro,'Wall 1A');
print -djpeg file1
%
figure
x1b = reshape(xPhys(nele1+1:2*nele1),[nely1,nelx1,nelz1]);
display_3D_shift2_cam(x1b,0,0,0,180,30)
title('Wall 1B - Front View')
file1 =  strcat(s_intro,'Wall 1B');
print -djpeg file1
%
% estraggo pilastri
xpil1 = ones(nely,nelx2,nelz1);
figure
x2a = reshape(xPhys(2*nele1+1:2*nele1+nele2),[nely2,nelx2,nelz2]);
display_3D_shift2_cam(x2a,0,0,nelz1,90,30)
display_3D_shift2_cam(xpil1,0,0,0,90,30)
display_3D_shift2_cam(xpil1,0,0,nelz1+nelz2,90,30)
title('Wall 2A - Front View')
file1 =  strcat(s_intro,'Wall 2A');
print -djpeg file1
%
figure
x2b = reshape(xPhys(2*nele1+nele2+1:2*nele1+2*nele2),[nely2,nelx2,nelz2]);
display_3D_shift2_cam(x2b,0,0,nelz1,90,30)
display_3D_shift2_cam(xpil1,0,0,0,270,30)
display_3D_shift2_cam(xpil1,0,0,nelz1+nelz2,90,30)
title('Wall 2B - Front View')
```

```
file1 =  strcat(s_intro,'Wall 2B');
print -djpeg file1
```

```
% === DISPLAY 3D TOPOLOGY (ISO-VIEW) ===
function display_3D_shift2(rho,shiftx,shifty,shiftz)
[nely,nelx,nelz] = size(rho);
hx = 1; hy = 1; hz = 1;            % User-defined unit element size
face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
set(gcf,'Name','ISO display','NumberTitle','off');
for k = 1:nelz
    z = (k-1)*hz+shiftz;
    for i = 1:nelx
        x = (i-1)*hx+shiftx;
        for j = 1:nely
            y = nely*hy - (j-1)*hy+shifty;
            if (rho(j,i,k) ≥ 0.)  % User-defined display density threshold
                vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; x y z+hx;x y-hx z+hx; x+hx y-hx z+hx;x+hx y z+hx];
                vert(:,[2 3]) = vert(:,[3 2]); vert(:,2,:) = -vert(:,2,:);
                patch('Faces',face,'Vertices',vert,'FaceColor',...\\
                [0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k))]);
                hold on;
            end
        end
    end
end
%axis equal; axis tight; axis off; box on; view([25,30]); pause(1e-6);
axis equal; axis tight; axis off; box on; view([38,30]); pause(1e-6);
end
```

```
% === DISPLAY 3D TOPOLOGY (ISO-VIEW) ===
function display_3D_shift2_cam(rho,shiftx,shifty,shiftz,az,el)
[nely,nelx,nelz] = size(rho);
hx = 1; hy = 1; hz = 1;            % User-defined unit element size
face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
set(gcf,'Name','ISO display','NumberTitle','off');
for k = 1:nelz
    z = (k-1)*hz+shiftz;
    for i = 1:nelx
        x = (i-1)*hx+shiftx;
        for j = 1:nely
            y = nely*hy - (j-1)*hy+shifty;
            if (rho(j,i,k) > 0.35)  % User-defined display density threshold
                vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; x y z+hx;x y-hx z+hx; x+hx y-hx z+hx;x+hx y z+hx];
                vert(:,[2 3]) = vert(:,[3 2]); vert(:,2,:) = -vert(:,2,:);
                patch('Faces',face,'Vertices',vert,'FaceColor',...\\
                [0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k))]);
                hold on;
            end
        end
    end
end
axis equal; axis tight; axis off; box on; view([az,el]); pause(1e-6);
end
```

References

[1] Svanberg K. The method of moving asymptotes - a new method for structural optimization. Int J Numer Methods Eng 1987;24:359–73.

[2] Liu K, Tovar A. An efficient 3D topology optimization code written in Matlab. Struct Multidiscip Optim 2014;50:1175–96.

[3] Venini P, Pingaro M. Static and dynamic topology optimization: an innovative unifying approach. Struct Multidiscip Optim 2023;66:66–85. https://doi.org/10.1007/s00158-023-03528-6.

[4] Costa G, Montemurro M, Pailhés J. Nurbs hyper-surfaces for 3d topology optimization problems. Mech Adv Mat Struct 2021;28(7):665–84. https://doi.org/10.1080/15376494.2019.1582826.

[5] Zhuang C, Xiong Z, Ding H. An efficient 2d/3d nurbs-based topology optimization implementation using page-wise matrix operation in Matlab. Struct Multidiscip Optim 2023;66:254. https://doi.org/10.1007/s00158-023-03701-x.

[6] Gao J, Xue H, Gao L, Luo Z. Topology optimization for auxetic metamaterials based on isogeometric analysis. Comput Methods Appl Mech Eng 2019;352:211–36. https://doi.org/10.1016/j.cma.2019.04.021.

[7] Dong G, Tang Y, Zha Y. A 149 line homogenization code for three-dimensional cellular materials written in Matlab. J Eng Mater Technol Trans ASME 2021;141:011005. https://doi.org/10.1080/15376494.2019.1582826.

[8] Montemurro M. On the structural stiffness maximisation of anisotropic continua under inhomogeneous Neumann-Dirichlet boundary conditions. Compos Struct 2022;287:115289. https://doi.org/10.1016/j.compstruct.2022.115289.

[9] Roiné T, Montemurro M, Pailhès J. Stress-based topology optimization through non-uniform rational basis spline hypersurfaces. Mech Adv Mat Struct 2021;29(23):3387–407. https://doi.org/10.1080/15376494.2021.1896822.

[10] Xu J, Gao L, Xiao M, Gao J, Hao L. Design of materials using topology optimization and energy-based homogenization approach in Matlab. Int J Mech Sci 2020;166:105103. https://doi.org/10.1016/j.ijmecsci.2019.105103.

[11] Chao M, Na Q, Xiang X. A fully automatic computational framework for beam structure design from continuum structural topology optimization. Struct Multidiscip Optim 2023;66:250. https://doi.org/10.1007/s00158-023-03704-8.

[12] Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O. Efficient topology optimization in Matlab using 88 lines of code. Struct Multidiscip Optim 2011;43:1–16.

[13] Skogestad S, Postlethwaite I. Multivariable feedback control - analysis and design. Baffins Lane, Chichester: John Wiley and Sons; 1996.

[14] Strang G. Linear algebra and learning from data. Wellesley: Cambridge Press; 2019.
[15] Gerzen N, Barthold F-J. Enhanced analysis of design sensitivities in topology optimization. Struct Multidiscip Optim 2012;46:585–95. https://doi.org/10.1007/s00158-012-0778-4.
[16] Sigmund O. On benchmarking and good scientific practise in topology optimization. Struct Multidiscip Optim 2022;65(315):1–10. https://doi.org/10.1007/s00158-022-03427-2.
[17] Brunton S, Kutz J. Data-driven science and engineering – machine learning, dynamical systems and control. Cambridge: Cambridge University Press; 2019.
[18] Bendsøe M, Sigmund O. Material interpolation schemes in topology optimization. Arch Appl Mech 1999;69(9–10):635–54.
[19] Cook RD, Malkus DS, Plesha ME, Witt RJ. Concepts and applications of finite element analysis. 4th edition. Hoboken, NJ: John Wiley and Sons; 2002.
[20] Chen XS, li W. Sensitivity analysis for the generalized singular value decomposition. Numer Linear Algebra Appl 2013;20:138–49. https://doi.org/10.1002/nla.1829.
[21] Freudenberg JS, Looze DP, Cruz JB. Robustness analysis using singular value sensitivities. Int J Control 1982;35:95–116. https://doi.org/10.1080/00207178208922604.
[22] Stromberg L, Beghini A, Baker W, Paulino G. Application of layout and topology optimization using pattern gradation for the conceptual design of buildings. Struct Multidiscip Optim 2011;43:165–80. https://doi.org/10.1007/s00158-010-0563-1.
[23] Stromberg L, Beghini A, Baker W, Paulino G. Topology optimization for braced frames: combining continuum and beam/column elements. Eng Struct 2012;37:106–24. https://doi.org/10.1016/j.engstruct.2011.12.034.
[24] Di Lorenzo G, Tartaglia R, Prota A, Landolfo R. Design procedure for orthogonal steel exoskeleton structures for seismic strengthening. Eng Struct 2023;275. https://doi.org/10.1016/j.engstruct.2022.115252.