



HAL
open science

2D topology optimization MATLAB codes for piezoelectric actuators and energy harvesters

Abbas Homyouni, Thomas Schlinquer, Abdenbi Mohand Ousaid, Micky Rakotondrabe

► **To cite this version:**

Abbas Homyouni, Thomas Schlinquer, Abdenbi Mohand Ousaid, Micky Rakotondrabe. 2D topology optimization MATLAB codes for piezoelectric actuators and energy harvesters. Structural and Multidisciplinary Optimization, 2020, pp.32. hal-03006390

HAL Id: hal-03006390

<https://hal.science/hal-03006390v1>

Submitted on 15 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



2D topology optimization MATLAB codes for piezoelectric actuators and energy harvesters

Abbas Homayouni-Amlashi^{1,2} · Thomas Schlienger² · Abdenbi Mohand-Ousaid² · Micky Rakotondrabe¹

Received: 19 May 2020 / Revised: 8 August 2020 / Accepted: 14 August 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

In this paper, two separate topology optimization MATLAB codes are proposed for a piezoelectric plate in actuation and energy harvesting. The codes are written for one-layer piezoelectric plate based on 2D finite element modeling. As such, all forces and displacements are confined in the plane of the piezoelectric plate. For the material interpolation scheme, the extension of solid isotropic material with penalization approach known as PEMP-P (piezoelectric material with penalization and polarization) which considers the density and polarization direction as optimization variables is employed. The optimality criteria and method of moving asymptotes (MMA) are utilized as optimization algorithms to update the optimization variables in each iteration. To reduce the numerical instabilities during optimization iterations, finite element equations are normalized. The efficiencies of the codes are illustrated numerically by illustrating some basic examples of actuation and energy harvesting. It is straightforward to extend the codes for various problem formulations in actuation, energy harvesting and sensing. The finite element modeling, problem formulation and MATLAB codes are explained in detail to make them appropriate for newcomers and researchers in the field of topology optimization of piezoelectric material.

Keywords Topology optimization · MATLAB code · Piezoelectric actuator · Piezoelectric energy harvester

1 Introduction

Topology optimization (TO) is a methodology to distribute the material within a design domain in an optimal way while there is no prior knowledge about the final layout of the material (Bendsoe 2013). This main specification of TO provides a great degree of freedom in terms of designing innovative structures to satisfy predefined

engineering goals. Historically, minimization of mechanical deformation of a structure under application of different loading conditions was a classical engineering goal (Schmit 1960). Aiming for this goal, the work of Bendsoe and Kikuchi (1988) paves the way for a methodology known today as topology optimization. The general idea of this methodology is the combination of finite element method and optimization to maximize or minimize an objective function. In this regard, the design domain is discretized by a finite number of elements and design variables for the optimization problem are the variables attributed to each of these elements. Different approaches are introduced in the literature to implement the TO method (Sigmund and Maute 2013). Among these approaches, homogenization approach was proposed to optimize the porous elements as unit cells or microstructures within the design domain to obtain the final layout (Suzuki and Kikuchi 1991; Bendsoe and Sigmund 1995). However, the number of optimization variables is high in homogenization method which can make the optimization cumbersome. The other famous and popular approach is the SIMP approach which stands for solid isotropic material with penalization. In this approach, the elements in the design domain can have intermediate densities (Bendsoe 1989; Bendsoe 2013). This will let the

Responsible Editor: Palaniappan Ramu

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s00158-020-02726-w>) contains supplementary material, which is available to authorized users.

✉ Abbas Homayouni-Amlashi
abbas.homayouni@femto-st.fr

¹ Laboratoire Génie de Production, Nationale School of Engineering in Tarbes (ENIT), Toulouse INP, University of Toulouse, 47, Avenue d'Azereix, Tarbes, France

² CNRS, FEMTO-ST Institute, Université Bourgogne Franche-Comté, Besançon, 25000, France

elements to be gray in addition to black (material) and white (void). However, due to practical constraints, it is desired that the optimization finally converges to a black and white layout. To do so, a penalization factor is defined for intermediate densities. One of the reasons for the popularity of this approach is its simplicity of implementation in comparison with other approaches. Similar to SIMP approach, there is evolutionary structural optimization (ESO) (Xie and Steven 1993) or the more general form bi-directional evolutionary structural optimization (BESO) (Xia et al. 2018) which is about removing or adding the elements inside the design domain during optimization iterations. In addition to the aforementioned approaches, there are other approaches including level set method (van Dijk et al. 2013; Andreassen et al. 2020) and method of moving morphable components (MMC) (Guo et al. 2014; Zhang et al. 2017) which is a geometrical approach. For a detailed review and comparison between these approaches, one can refer to the following review papers: Sigmund and Maute (2013) and Deaton and Grandhi (2014).

Due to the success of TO methodology, several implementation codes in different software are published in the literature. Using the SIMP approach, Sigmund (2001) published the 99 lines of MATLAB code for 2D topology optimization of compliance problems. Andreassen et al. (2011) published the 88 lines of MATLAB code which was an improvement of Sigmund's 99 lines of code while having much faster speed in each iteration thanks to introducing the connectivity matrix that facilitates the assembly procedure of elemental matrices. Liu and Tovar (2014) published the TOP3D MATLAB code by extension of the 88 lines of code for topology optimization of 3D structures. Chen et al. (2019) published 213 lines of MATLAB code for 2D topology optimization of geometrically nonlinear structures. There are other published codes using other TO approaches like level set method (Challis 2010; Wei et al. 2018; Yaghmaei et al. 2020), BESO (Xia et al. 2018), and projection method (Smith and Norato 2020). These published codes facilitate the implementation of TO methodology for various applications. For this reason, the application of TO can be seen in solving different problems including the compliance problems (Bendsoe 2013), compliant mechanism problems (Zhu et al. 2020), heat conduction (Gersborg-Hansen et al. 2006), and smart materials in particular the piezoelectric materials (Sigmund and Torquato 1999).

Due to their electromechanical coupling effect, piezoelectric materials have applications in actuation, sensing and energy harvesting. Plenty of methods can be found in the literature to analyze and improve the performance of the piezoelectric structures, whether actuators, energy harvesters or sensors such as geometrical and size opti-

mization (Schlienger et al. 2017; Bafumba Liseli and Agnus 2019; Homayouni-Amlashi et al. 2020a), shape optimization (Muthalif and Nordin 2015), layers number optimization (Rabenorosa and et al 2015), or parameters sub-optimization (Rakotondrabe and Khadraoui 2013; Khadraoui et al. 2014) with interval techniques (Rakotondrabe 2011). After development of TO methodology, it is extended to different physics (Alexandersen and Andreassen 2020; Deaton and Grandhi 2014) including the piezoelectricity. Primarily, the homogenization approach is used (Silva et al. 1997; Sigmund et al. 1998). Afterwards, other approaches including SIMP (Kögl and Silva 2005), BESO (de Almeida 2019) or level set method (Chen et al. 2010) are also explored. By defining proper objective functions, TO methodology is applied to piezoelectric actuators (Moretti and Silva 2019; Gonçalves et al. 2018), sensors (Menuzzi et al. 2018) and energy harvesters (Homayouni-Amlashi et al. 2020b; Homayouni-Amlashi 2019; Townsend et al. 2019). The publications considered different types of system modeling including the static (Zheng et al. 2009), dynamic (Noh and Yoon 2012; Wein et al. 2009), modal (Wang et al. 2017) and electrical circuit coupling (Salas et al. 2018; Rupp et al. 2009). Different types of problem formulation can be found as well such as optimization with stress constraints (Wein et al. 2013). Although the application of TO methodology to piezoelectric materials is well established in the literature, no implementation code is published yet.

In this paper, 2D topology optimization codes are proposed for actuation and energy harvester by using the extension of SIMP approach known as PEMAP-P (piezoelectric material with penalization and polarization). The codes are written based on the 88 lines of MATLAB code written by Andreassen et al. (2011), except the code is extended and modified considerably to consider the electromechanical coupling effect of piezoelectric material and problem formulation. In Section 2, the finite element modeling of one-layer piezoelectric plate is presented by using the plane stress assumption. The finite element equation is derived for both actuation and energy harvesting. A normalization is applied to the finite element equation which significantly reduces the numerical instabilities in optimization iterations. Hence, the proposed codes in this paper work smoothly in both actuation and energy harvesting. In Section 3, first, the material interpolation scheme for piezoelectric material is explained. Then, the optimization problem is formulated for both actuation and energy harvesters. The problem formulations are basic for educational purposes. The sensitivity analysis is performed, and finally, the optimization algorithms are explained. In Section 4, the MATLAB codes are explained part by part in detail. The explanations in this part help the readers to

extend and implement the codes for their own purposes. In Section 5, different numerical examples are illustrated and the modification to the original codes to implement those numerical examples are expressed as well.

2 Finite element modeling

2.1 Constitutive equation

The general linearly coupled mechanical and electrical constitutive equation of piezoelectric materials by neglecting the thermal coupling can be written as (Lerch 1990)

$$\begin{aligned} \bar{T} &= c^E \bar{S} - e \bar{E} \\ \bar{D} &= e^T \bar{S} + \varepsilon^S \bar{E} \end{aligned} \tag{1}$$

In (1), \bar{T} and \bar{S} are the vectors of mechanical stress and strain while c^E is the stiffness tensor in constant electrical field. \bar{D} and \bar{E} are the vectors of electrical displacement and electrical field. e is the piezoelectric matrix, ε^S is the matrix of permittivity in constant mechanical strain and T shows the matrix transpose.

The 4-mm tetragonal crystal class piezoelectric material (Piefort 2001) which has orthotropic anisotropy is considered to derive the corresponding model. This class includes most of the piezoelectric material in particular the well-known PZT (lead zirconate titanate) materials. By this consideration, the mechanical stiffness tensor, piezoelectric matrix and permittivity for full 3D modeling are

$$\begin{aligned} c^E &= \begin{bmatrix} c_{11}^E & c_{12}^E & c_{13}^E & 0 & 0 & 0 \\ c_{12}^E & c_{11}^E & c_{13}^E & 0 & 0 & 0 \\ c_{13}^E & c_{13}^E & c_{33}^E & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44}^E & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44}^E & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66}^E \end{bmatrix} \\ e^T &= \begin{bmatrix} 0 & 0 & 0 & 0 & e_{15} & 0 \\ 0 & 0 & 0 & e_{15} & 0 & 0 \\ e_{31} & e_{31} & e_{33} & 0 & 0 & 0 \end{bmatrix} \\ \varepsilon^S &= \begin{bmatrix} \varepsilon_{11}^S & 0 & 0 \\ 0 & \varepsilon_{11}^S & 0 \\ 0 & 0 & \varepsilon_{33}^S \end{bmatrix} \end{aligned} \tag{2}$$

Now, a piezoelectric plate sandwiched between two electrodes as shown in Fig. 1 is considered. Without loss of generality, several assumptions are considered for this configuration.

- the thickness to length ratio of the piezoelectric plate is less than 1/10,

- the piezoelectric plate is confined to have planar movement and it is subjected to loading only in the xy plane,
- the thickness of the electrodes is negligible in comparison with thickness of piezoelectric plate,
- the electromechanical system is assumed to be linear,
- the electrodes are perfectly conductive,
- the polarization direction is perpendicular to the plate in parallel to \vec{z} axis,
- the electrical field is uniform in the direction of thickness aligned with the polling direction,
- the variation of the potential in the direction of the thickness is linear,

The first two assumptions let us use the plane stress assumption modeling for the piezoelectric plate (Hutton and Wu 2004). In this case, any stress in the direction of z will be zero. By considering transversely isotropic piezoelectric material and considering plane stress assumption, the piezoelectric plate has in-plane isotropic behavior. Furthermore, by poling the piezoelectric material in the z direction, the only non-zero electric field will be in the z direction. In this case, the reduced (2D) form of piezoelectric constitutive equation can be derived in the following form (Junior et al. 2009)

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ D_3 \end{bmatrix} = \begin{bmatrix} c_{11}^* & c_{12}^* & 0 & -e_{31}^* \\ c_{12}^* & c_{11}^* & 0 & -e_{31}^* \\ 0 & 0 & c_{33}^* & 0 \\ e_{31}^* & e_{31}^* & 0 & \varepsilon_{33}^* \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ E_3 \end{bmatrix} \tag{3}$$

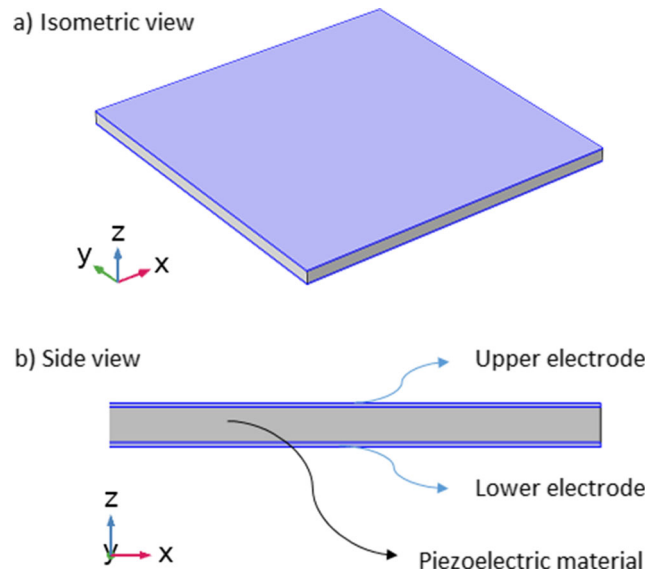


Fig. 1 Piezoelectric plate sandwiched between two electrodes. a Isometric view. b Side view

The components of the reduced constitutive equation can be written as (Junior et al. 2009)

$$c_{11}^{*E} = c_{11}^E - \frac{(c_{13}^E)^2}{c_{33}^E}, \quad c_{12}^* = c_{12}^E - \frac{(c_{13}^E)^2}{c_{33}^E}, \quad c_{33}^* = c_{66}$$

$$e_{31}^* = e_{31} - e_{33} \frac{c_{13}^E}{c_{33}^E}, \quad \varepsilon_{33}^* = \varepsilon_{33}^S + \frac{e_{33}^2}{c_{33}^E} \quad (4)$$

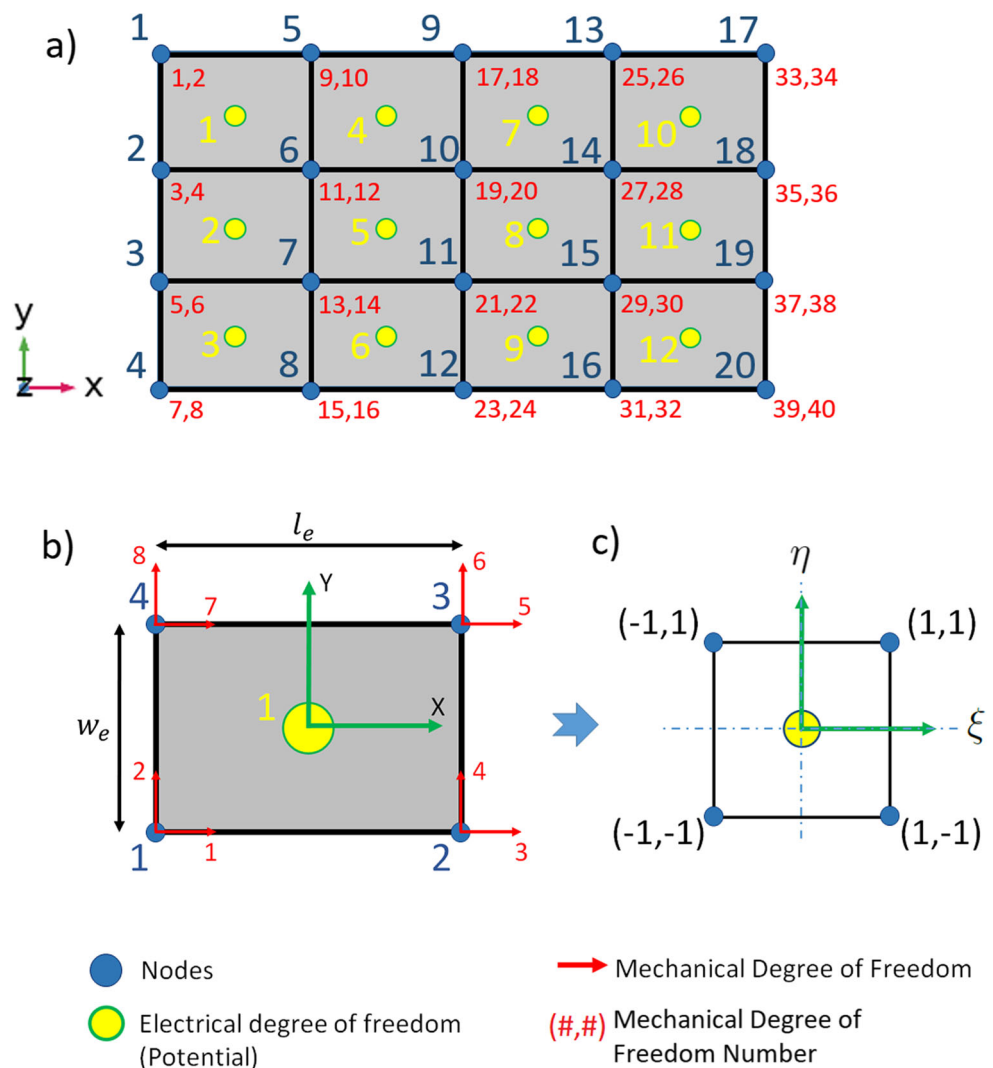
The obtained constitutive equation in (3) will be used in the finite element modeling of the piezoelectric plate which will be discussed in the next section.

2.2 Piezoelectric finite element model

In this section, to derive the finite element (FE) formulation, the piezoelectric plate is discretized by rectangular elements which are particular form of the more general 2D elements called “bilinear quadrilateral element” (Hutton and Wu 2004; Kattan 2010). It should be noted that since the

thickness of electrodes is negligible in comparison with the thickness of the piezoelectric plate, its structural effects are neglected in the modeling. Hence, only the piezoelectric plate is discretized by the finite number of elements. The schematic form of this discretization is illustrated in Fig. 2a. As can be seen in this figure, The piezoelectric plate is discretized as 3 by 4 elements. It is clear that a finer discretization will be used for the numerical optimization. It can be seen in the figure that each rectangular element has 4 nodes and each node has 2 in-plane mechanical degrees of freedom regarding the displacement in x and y directions. The rectangular element shown in Fig. 2b can have arbitrary length l_e and width w_e . In fact, with the method of finite element modeling which is used to write the optimization code, the dimensions of plate and number of elements can be defined separately. This freedom will have two advantages: first, for a predefined geometry of a piezoelectric plate, higher number of elements can be defined to have better results in terms of having small detail.

Fig. 2 Finite element discretization of design domain. Panel (a) is the numbering format inside the design domain. Panel (b) is the numbering format inside each element. Panel (c) is the parent element in the natural coordinates



Second, it is possible to define lower number of elements when reducing the computation time is necessary. As it is illustrated in Fig. 2c, the rectangular element is mapped to a parent element which is a square element with natural coordinates ξ and η . The displacement of every point within the element will be expressed by the displacement of the nodes through the interpolation functions in the following format (Hutton and Wu 2004).

$$\begin{aligned} x &= x_1n_1 + x_2n_2 + x_3n_3 + x_4n_4 \\ y &= y_1n_1 + y_2n_2 + y_3n_3 + y_4n_4 \end{aligned} \tag{5}$$

and the interpolation functions can be written based on the natural coordinates

$$\begin{aligned} n_1 &= \frac{1}{4}(1 - \xi)(1 - \eta), & n_2 &= \frac{1}{4}(1 + \xi)(1 - \eta) \\ n_3 &= \frac{1}{4}(1 + \xi)(1 + \eta), & n_4 &= \frac{1}{4}(1 - \xi)(1 + \eta) \end{aligned} \tag{6}$$

where the matrix of interpolation function can be written as,

$$N = \begin{bmatrix} n_1 & 0 & n_2 & 0 & n_3 & 0 & n_4 & 0 \\ 0 & n_1 & 0 & n_2 & 0 & n_3 & 0 & n_4 \end{bmatrix} \tag{7}$$

So far, for each element, just mechanical degrees of freedom are considered. However, in piezoelectric material, the mechanical and electrical fields are coupled. Therefore, the electrical degree of freedom should be modeled as well. The general approach for this case is to consider one electrical degree of freedom for each node in addition to mechanical degrees of freedom as explained in Lerch (1990). However, here by assuming that conductive electrodes are placed on top and bottom of the piezoelectric plates as shown in Fig. 1, the electrical potential over each electrode is constant. This condition is known as equipotential condition. Furthermore, by considering the bottom electrode as ground electrode, the whole piezoelectric plate will have one electrical degree of freedom. On the other hand, for the purpose of elemental sensitivity analysis which will be explained in Section 3, for each element, one electrical degree of freedom is considered as it is shown in yellow in Fig. 2. The global equipotential condition will be imposed after assembling the global matrices.

Now, the strain and electrical field of each element can be expressed with the help of mechanical and electrical degrees of freedom

$$\bar{S} = B_u u, \quad \bar{E} = B_\phi \phi \tag{8}$$

In (8), u and ϕ are the vectors of mechanical displacement and scalar value of electric potential respectively. B_u is

the strain displacement matrix which is written as follows (Kattan 2010),

$$\begin{aligned} B_u &= \frac{1}{|J|} [B_1 \ B_2 \ B_3 \ B_4] \\ B_i &= \begin{bmatrix} a \frac{\partial n_i}{\partial \xi} - b \frac{\partial n_i}{\partial \eta} & 0 \\ 0 & c \frac{\partial n_i}{\partial \eta} - d \frac{\partial n_i}{\partial \xi} \\ c \frac{\partial n_i}{\partial \eta} - d \frac{\partial n_i}{\partial \xi} & a \frac{\partial n_i}{\partial \xi} - b \frac{\partial n_i}{\partial \eta} \end{bmatrix} \end{aligned} \tag{9}$$

and the parameters a, b, c and d are given by Kattan (2010),

$$\begin{aligned} a &= \frac{1}{4}[y_1(\xi - 1) + y_2(-1 - \xi) + y_3(1 + \xi) + y_4(1 - \xi)] \\ b &= \frac{1}{4}[y_1(\eta - 1) + y_2(1 - \eta) + y_3(1 + \eta) + y_4(-1 - \eta)] \\ c &= \frac{1}{4}[x_1(\eta - 1) + x_2(1 - \eta) + x_3(1 + \eta) + x_4(-1 - \eta)] \\ d &= \frac{1}{4}[x_1(\xi - 1) + x_2(-1 - \xi) + x_3(1 + \xi) + x_4(1 - \xi)] \end{aligned} \tag{10}$$

where x_i and y_i are the coordinates of the nodes in the rectangular element before mapping.

The determinant of Jacobian matrix J , which transfers the natural coordinates to the generalized coordinates, is

$$\begin{aligned} |J| &= \frac{1}{8} [x_1 \ x_2 \ x_3 \ x_4] \\ &\times \begin{bmatrix} 0 & 1 - \eta & \eta - \xi & \xi - 1 \\ \eta - 1 & 0 & \xi + 1 & -\xi - \eta \\ \xi - \eta & -\xi - 1 & 0 & \eta + 1 \\ 1 - \xi & \xi + \eta & -\eta - 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \end{aligned} \tag{11}$$

By considering the last two assumptions, B_ϕ is (Junior et al. 2009)

$$B_\phi = 1/h \tag{12}$$

where h is the thickness of the piezoelectric plate. Now by using the Hamilton’s variational principle and neglecting the damping effect, the linear differential equation for one single element can be written in the following form (Lerch 1990)

$$\begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{u} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} k_{uu} & k_{u\phi} \\ k_{\phi u} & -k_{\phi\phi} \end{bmatrix} \begin{bmatrix} u \\ \phi \end{bmatrix} = \begin{bmatrix} f \\ q \end{bmatrix} \tag{13}$$

in which m is the mass matrix, k_{uu} is the mechanical stiffness matrix, $k_{u\phi}$ is the piezoelectric coupling matrix, $k_{\phi\phi}$ is the dielectric stiffness matrix, f is the external

mechanical force and q is the charge. These components of linear differential equation are derived in the following form

$$\begin{aligned}
 k_{uu} &= h \int_A B_u^T c^E B_u |J| d\xi d\eta \\
 k_{u\phi} &= h \int_A B_u^T e^T B_\phi |J| d\xi d\eta \\
 k_{\phi\phi} &= h \int_A B_\phi^T \varepsilon^S B_\phi |J| d\xi d\eta \\
 m &= \rho h \int_A N^T N |J| d\xi d\eta
 \end{aligned} \tag{14}$$

where A is the top surface area of the element and ρ is the density of the material. In fact, (14) illustrates the analytical calculations of elemental matrices. However, for numerical implementation in MATLAB, two-point Gauss quadrature method (Hutton and Wu 2004) is utilized for calculation of the elemental matrices numerically, which gives the exact values. The implementation procedure is explained later in Section 4.

To have the global FEM equation for a whole piezoelectric plate, the elemental matrices in (13) should be assembled, which is a general procedure in the FEM methodology and which will also be explained in Section 4. After assembling the elemental matrices, the global finite element equation for the whole design domain can be written as

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{U} \\ \ddot{\Phi} \end{bmatrix} + \begin{bmatrix} K_{uu} & K_{u\phi} \\ K_{\phi u} & -K_{\phi\phi} \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} = \begin{bmatrix} F \\ Q \end{bmatrix} \tag{15}$$

Now, for two cases of actuation and energy harvesting, the global FEM (15) can be interpreted in different ways. Here, we focus on static actuation so that the dynamics will not be considered. Therefore, the global FEM equation for the actuation can be written as

$$K_{uu}U + K_{u\phi}\Phi = F \tag{16}$$

This equation will be used to calculate the mechanical displacement due to applied potential.

For the energy harvesting case, the external charge (Q) is considered to be zero. In addition, the external force is considered to be a harmonic excitation of frequency Ω . In this case, by considering a linear electromechanical system, the force and the response of the system can be stated as

$$\begin{aligned}
 F &= f_0 e^{i\Omega t} \\
 U &= u_0 e^{i\Omega t}, \Phi = \phi_0 e^{i\Omega t}
 \end{aligned} \tag{17}$$

where f_0 , u_0 and ϕ_0 are the amplitude of harmonic force, displacement and potential. By substituting the (17) in (15), the global FEM equation for the energy harvesting case can be written as

$$-\Omega^2 \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} + \begin{bmatrix} K_{uu} & K_{u\phi} \\ K_{\phi u} & -K_{\phi\phi} \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \tag{18}$$

Equation (18) can also be written in following form

$$\begin{bmatrix} K_{uu} - M\Omega^2 & K_{u\phi} \\ K_{\phi u} & -K_{\phi\phi} \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \tag{19}$$

To solve the FEM (19), the mechanical boundary condition and equipotential condition should be applied. This will be explained in detail in Section 4.

2.3 Normalization

Here, the critical point is that the scale difference between the piezoelectric matrices including the mechanical stiffness matrices (k_{uu}) and ($k_{u\phi}$) and the dielectric stiffness matrix ($k_{\phi\phi}$) is huge. This huge scale difference can bring numerical instabilities in form of singularities in solving the final global FEM equation during the optimization loops. To eliminate the scale difference, a normalization is suggested (Homayouni-Amlashi 2019; Homayouni-Amlashi et al. 2020b) by factorizing the highest value of each elemental matrix which can be expressed in the following format

$$\begin{aligned}
 \tilde{k}_{uu} &= k_{uu}/k_0, & \tilde{k}_{u\phi} &= k_{u\phi}/\alpha_0 \\
 \tilde{k}_{\phi\phi} &= k_{\phi\phi}/\beta_0, & \tilde{m} &= m/m_0
 \end{aligned} \tag{20}$$

Starting by this normalization of elemental matrices, the actuation FEM (16) can be rewritten as

$$\tilde{K}_{uu}\tilde{U} + \tilde{K}_{u\phi}\tilde{\Phi} = \tilde{F} \tag{21}$$

in which

$$\begin{aligned}
 \tilde{F} &= F/f_0, & \tilde{U} &= U/u_0, & \tilde{\Phi} &= \Phi/\phi_0 \\
 u_0 &= f_0/k_0, & \phi_0 &= f_0/\alpha_0
 \end{aligned} \tag{22}$$

The same normalization can be performed on the energy harvesting FEM (19)

$$\begin{bmatrix} \tilde{K}_{uu} - \tilde{M}\tilde{\Omega}^2 & \tilde{K}_{u\phi} \\ \tilde{K}_{\phi u} & -\gamma\tilde{K}_{\phi\phi} \end{bmatrix} \begin{bmatrix} \tilde{U} \\ \tilde{\Phi} \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ 0 \end{bmatrix} \tag{23}$$

where

$$\tilde{\Omega}^2 = \Omega^2 m_0/k_0, \quad \gamma = k_0\beta_0/\alpha_0^2 \tag{24}$$

Here γ is a normalization factor which keeps the solution of the system equal before and after applying the normalization. This normalization factor is having the scale of 10^1 and in this way the scale difference between the piezoelectric matrices is eliminated. The proof of normalization is provided in the Appendix.

Now, by having the FEM equations (21) and (23), it is possible to enter the optimization phase. In the upcoming sections, optimization of actuator and energy harvester is separated.

3 Topology optimization

As explained before, topology optimization is about distribution of the material within a design domain while there is no prior knowledge of the final optimized layout of the structures (Bendsoe 2013). There are several approaches for topology optimization method (Maute and Sigmund 2013). However, the density-based approach is chosen for this paper since its efficiency is already established in many researches specially in the area of piezoelectric actuators (Kögl and Silva 2005; Ruiz et al. 2017; Moretti and Silva 2019) or energy harvesters (Homayouni-Amlashi 2019; Zheng et al. 2009; Noh and Yoon 2012).

3.1 Piezoelectric material interpolation scheme

One of the famous material interpolation schemes is the density-based approach which has been introduced to relax the optimization from the binary (void-material) problem. In this approach, for a discretized design domain by finite number of elements, the material properties of each element are related to element’s density through a power law interpolation function. For passive isotropic material, this interpolation function is referred to as solid isotropic material with penalization (SIMP) which relates the element’s Young modulus of elasticity to its density. For active non-isotropic piezoelectric material, the interpolation function is the extension of SIMP scheme which can be written as follows (Kögl and Silva 2005)

$$\begin{aligned}
 \tilde{k}_{uu}(x) &= (E_{\min} + x^{p_{uu}}(E_0 - E_{\min})) \tilde{k}_{uu} \\
 \tilde{k}_{u\phi}(x, P) &= (e_{\min} + x^{p_{u\phi}}(e_0 - e_{\min}))(2P - 1)^{p_P} \tilde{k}_{u\phi} \\
 \tilde{k}_{\phi\phi}(x) &= (\varepsilon_{\min} + x^{p_{\phi\phi}}(\varepsilon_0 - \varepsilon_{\min})) \tilde{k}_{\phi\phi} \\
 \tilde{m}(x) &= x \tilde{m}
 \end{aligned}
 \tag{25}$$

where E_{\min} , e_{\min} and ε_{\min} are small numbers to define the minimum values for stiffness, coupling and dielectric matrices while E_0 , e_0 and ε_0 are equal to 1 to define the maximum values of the respected matrices. The definition of minimum values is provided to avoid the singularities during the optimization iterations. x is the density ratio of each element which has a value between 0 and 1. P is the polarization variable which also has the value between 0 and 1 and determines the direction of polarization. p_{uu} , $p_{u\phi}$, $p_{\phi\phi}$ and p_P are penalization coefficients for the stiffness, coupling, dielectric matrices and polarization value respectively. It is obvious that in (25), the normalized form of piezoelectric matrices is used. However, the interpolation function is true for non-normalized matrices as well.

The introduced material interpolation scheme is known as PEMP-P (piezoelectric material with penalization and

polarization) (Kögl and Silva 2005), which is the extension of the SIMP approach. Although some projections are defined for SIMP to have a robust topology optimization (Wang et al. 2011), there are alternative interpolation functions like RAMP (rational approximation of material properties) (Stolpe and Svanberg 2001), or newer interpolation function introduced in Clausen et al. (2015). This latter one is also used in topology optimization of piezoelectric transducers (Donoso and Sigmund 2016). In fact, the study on preference of these interpolation functions is not in the scope of this paper. But, with the proposed MATLAB code, it will be easy for implementation of other interpolation functions.

After establishing the material interpolation scheme, the rest of this section will be divided into two parts: actuation and energy harvesting.

3.2 Actuation

3.2.1 Problem formulation

Following the classic approach for compliant mechanisms reported in Bendsoe (2013), the optimization of a planar piezoelectric actuator can be defined simply as displacement optimization or minimization of the following objective function,

$$\begin{aligned}
 \text{minimize } J_{\text{act}} &= -L^T \tilde{U} \\
 \text{Subject to } V(x) &= \sum_{i=1}^{NE} x_i v_i \leq V \\
 &0 < x_i \leq 1 \\
 &0 \leq P_i \leq 1
 \end{aligned}
 \tag{26}$$

where L is a vector with a value of 1 that corresponds to the output displacement node and 0 otherwise. In addition, a constraint is defined on the final volume of the optimized design. V is the target volume which is a fraction of the overall volume of the design domain while v_i is the volume of each element and NE is the total number of elements while i is the number of each element.

3.2.2 Sensitivity analysis

For applying a gradient-based optimization, the sensitivity of objective function with respect to the optimization variables should be calculated. As such, the sensitivity of objective function with respect to the x_i can be derived as

$$\begin{aligned}
 \frac{\partial J}{\partial x_i} &= \frac{\partial}{\partial x_i} \left(-L^T \tilde{U} + \Lambda^T \left(\tilde{K}_{uu} \tilde{U} + \tilde{K}_{u\phi} \tilde{\Phi} - \tilde{F} \right) \right) \\
 &= \frac{\partial}{\partial x_i} \left(\left(-L^T + \Lambda^T \tilde{K}_{uu} \right) \tilde{U} + \Lambda^T \tilde{K}_{u\phi} \tilde{\Phi} - \Lambda^T \tilde{F} \right)
 \end{aligned}
 \tag{27}$$

Through using the procedure known as adjoint method, Λ is introduced to avoid taking the derivative of displacement with respect to design variable, i.e., $\frac{\partial \tilde{u}_i}{\partial x}$. Therefore, the following adjoint equation should be solved

$$-L^T + \Lambda^T \tilde{K}_{uu} = 0 \tag{28}$$

where Λ is the global adjoint vector. By solving the adjoint (28), the sensitivity values can be obtained as

$$\frac{\partial J}{\partial x_i} = \lambda_i^T \frac{\partial \tilde{k}_{uu}}{\partial x_i} \tilde{u}_i + \lambda_i^T \frac{\partial \tilde{k}_{u\phi}}{\partial x_i} \tilde{\phi}_i \tag{29}$$

where λ_i is the elemental format of global adjoint vector Λ .

By using the same procedure, sensitivity analysis with respect to polarization P is derived in the following form as well

$$\frac{\partial J}{\partial P_i} = \lambda_i^T \frac{\partial \tilde{k}_{u\phi}}{\partial P_i} \tilde{\phi}_i \tag{30}$$

where λ_i is the same adjoint vector which is already calculated in (28).

Based on (29) and (30), the derivative of piezoelectric stiffness and coupling matrices with respect to design variables is required which can be derived with the help of (25) as

$$\begin{aligned} \frac{\partial \tilde{k}_{uu}}{\partial x_i} &= p_{uu}(E_0 - E_{min})x_i^{p_{uu}-1} \tilde{k}_{uu} \\ \frac{\partial \tilde{k}_{u\phi}}{\partial x_i} &= p_{u\phi}(e_0 - e_{min})x_i^{p_{u\phi}-1} (2P_i - 1)^{p_P} \tilde{k}_{u\phi} \end{aligned} \tag{31}$$

$$\frac{\partial \tilde{k}_{u\phi}}{\partial P_i} = 2p_P(e_0 - e_{min})(2P_i - 1)^{p_P-1} x_i^{p_{u\phi}} \tilde{k}_{u\phi} \tag{32}$$

After performing the sensitivity analysis, and defining the constraint, the optimization of variables should be done by the optimization algorithms which will be discussed later in this section.

3.3 Energy harvesting

3.3.1 Problem formulation

In the case of energy harvesting optimization, electrical boundary condition should be applied in addition to mechanical boundary condition. As mentioned previously, by considering perfectly conductive electrodes, the equipotential boundary condition can be applied by using the Boolean matrix in the following form (Cook and et al 2007)

$$\Phi = BV_p \tag{33}$$

in which V_p is the voltage of the top electrode while the bottom electrode considered as ground. For a general case of multi-layer piezoelectric plates, Boolean matrix B is having the dimension of $N_e \times N_p$ where N_e is the number of nodes and N_p is the number of electrodes. However, in the case of

the one piezoelectric plate of this paper, B will be a vector of ones.

It is worth to note that the equipotential condition given in (33) will be applied only to the energy harvesting FEM (19). For the case of actuation, the equipotential condition will be applied automatically by defining equal applied voltage for all of the elements.

By applying the equipotential condition in (33), the energy harvesting FEM (19) can be rewritten as

$$\begin{bmatrix} \overline{K_{uu}} & \overline{K_{u\phi}} \\ \overline{K_{\phi u}} & -\overline{K_{\phi\phi}} \end{bmatrix} \begin{bmatrix} \tilde{U} \\ V_p \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ 0 \end{bmatrix} \tag{34}$$

where

$$\begin{aligned} \overline{K_{uu}} &= [\tilde{K}_{uu} - \tilde{M}\tilde{\Omega}^2]_{bc} \\ \overline{K_{u\phi}} &= [\tilde{K}_{u\phi}B]_{bc} \\ \overline{K_{\phi\phi}} &= \gamma B^T \tilde{K}_{\phi\phi} B \end{aligned} \tag{35}$$

in which $([\]_{bc})$ shows the application of mechanical boundary condition.

Now, the objective function for the energy harvesting application should be defined. Generally, the energy conversion ratio (Zheng et al. 2009; Noh and Yoon 2012) or electromechanical coupling coefficient (de Almeida 2019) which is equivalent mathematically is chosen as the objective function. However, this format of objective function suffers from numerical instabilities during optimization iterations where it is suggested to penalize the mechanical energy as suggested in de Almeida (2019) and Salas et al. (2018). Therefore, to avoid the numerical instabilities here, a classical format of objective function is defined. The optimization is defined as minimization of the weighted sum of the mechanical and electrical energy of the system,

$$\begin{aligned} \text{minimize } J_{EH} &= w_j \Pi^S - (1 - w_j) \Pi^E \\ \text{Subject to } V(x) &= \sum_{i=1}^{NE} x_i v_i \leq V \\ &0 < x_i \leq 1 \\ &0 \leq P_i \leq 1 \end{aligned} \tag{36}$$

Π^E and Π^S are electrical and mechanical energies respectively which are defined in the following form (Noh and Yoon 2012; Zheng et al. 2009)

$$\Pi^S = \left(\frac{1}{2}\right) \tilde{U}^T \overline{K_{uu}} \tilde{U}, \Pi^E = \left(\frac{1}{2}\right) V_p^T \overline{K_{\phi\phi}} V_p \tag{37}$$

In optimization of (36), w_j is the weighing factor which has the value between 0 and 1. Choosing the value of 1 for w_j will make the optimization problem, a minimum compliance problem in which the goal is to minimize the mechanical deflection of the system under the applied force. By decreasing the value of w_j , more weight will be given to maximize the output electrical energy. However, choosing

very small values for w_j will result in mechanically unstable layouts. Therefore, the value for w_j will be found by using trial and error approach. The basis for choosing this value can be the maximum energy conversion factor of the plate under the same force. For example, in Homayouni-Amlashi et al. (2020b), the maximum energy conversion factor for an optimized piezoelectric plate under planar excitation is found to be 0.03 while in Noh and Yoon (2012) for a two-layer optimized piezoelectric plate under the bending force this ratio is 0.1. Therefore, the initial value of w_j for the trial error approach can be considered between 0.01 and 0.1. The final chosen value of w_j depends on the maximum stress and strain induced by the defined mechanical input to the structure which can be revealed by the post processing analysis. On the other hand, stress and strain constraints can also be considered in the optimization problem as it is investigated by Wein et al. (2013).

In fact, the advantage of the objective function defined here is that the optimization algorithm converges very smoothly to the final result. However, the drawback of this objective function is that the obtained result can be sub-optimal depending on the chosen value for the w_j . On the other hand, the problem of sub-optimal results exists in other formats of the objective function. For example, Noh and Yoon (2012) showed that by considering the energy conversion factor (Π^E / Π^S) as objective function, different values of penalization factors can produce different results.

Eventually, we believe that the chosen objective function suits the educational purpose of this paper. Indeed, with the help of provided MATLAB code, the readers can easily change the code to implement other objective functions.

3.3.2 Sensitivity analysis

Similar to the actuation case, the next step after defining the objective function is sensitivity analysis. Since the objective function in (36) consists of mechanical and electrical energies, the sensitivity of each energy with respect to density ratio x can be found as (Zheng et al. 2009; Homayouni-Amlashi 2019; Homayouni-Amlashi et al. 2020b)

$$\frac{\partial \Pi^S}{\partial x_i} = \left(\frac{1}{2} \tilde{u}_i^T + \lambda_{1,i}^T \right) \frac{\partial (\tilde{k}_{uu} - \tilde{m} \Omega^2)}{\partial x_i} \tilde{u}_i + \lambda_{1,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial x_i} \tilde{\phi}_i + \mu_{1,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial x_i} \tilde{u}_i - \mu_{1,i}^T \frac{\gamma \partial \tilde{k}_{\phi\phi}}{\partial x_i} \tilde{\phi}_i \quad (38)$$

$$\frac{\partial \Pi^E}{\partial x_i} = \frac{1}{2} \tilde{\phi}_i^T \frac{\gamma \partial \tilde{k}_{\phi\phi}}{\partial x_i} \tilde{\phi}_i - \mu_{2,i}^T \frac{\gamma \partial \tilde{k}_{\phi\phi}}{\partial x_i} \tilde{\phi}_i + \lambda_{2,i}^T \frac{\partial (\tilde{k}_{uu} - \tilde{m} \Omega^2)}{\partial x_i} u_i + \lambda_{2,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial x_i} \tilde{\phi}_i + \mu_{2,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial x_i} \tilde{u}_i \quad (39)$$

in which μ and λ are the elemental adjoint vectors which are calculated by the following global coupled system

$$\begin{bmatrix} \overline{K_{uu}} & \overline{K_{u\phi}} \\ \overline{K_{\phi u}} & -\overline{K_{\phi\phi}} \end{bmatrix} \begin{bmatrix} \Lambda_1 \\ \Upsilon_1 \end{bmatrix} = \begin{bmatrix} -\overline{K_{uu}} \tilde{U} \\ 0 \end{bmatrix} \\ \begin{bmatrix} \overline{K_{uu}} & \overline{K_{u\phi}} \\ \overline{K_{\phi u}} & -\overline{K_{\phi\phi}} \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Upsilon_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\overline{K_{\phi\phi}} V_p \end{bmatrix} \quad (40)$$

where Λ and Υ are the global adjoint vectors which need to be disassembled to form the elemental adjoint vectors

$$[\lambda_1]_{bc} = \Lambda_1, [\lambda_2]_{bc} = \Lambda_2, [\mu_1] = B \Upsilon_1, [\mu_2] = B \Upsilon_2 \quad (41)$$

Now, the sensitivities with respect to polarization (P) are calculated as well (Homayouni-Amlashi et al. 2020b; Homayouni-Amlashi 2019)

$$\frac{\partial \Pi^S}{\partial P_i} = \lambda_{1,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial P_i} \tilde{\phi}_i + \mu_{1,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial P_i} \tilde{u}_i \\ \frac{\partial \Pi^E}{\partial P_i} = \lambda_{2,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial P_i} \tilde{\phi}_i + \mu_{2,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial P_i} \tilde{u}_i \quad (42)$$

Based on sensitivity equations in (39) and (42), the derivative of all piezoelectric matrices with respect to the design variables is required. The derivative of stiffness and coupling matrices is found in (31) and (32). Here, the derivative of dielectric matrix and mass matrix is also required which are

$$\frac{\partial \tilde{k}_{\phi\phi}}{\partial x_i} = p_{\phi\phi} (\epsilon_0 - \epsilon_{min}) x_i^{p_{\phi\phi}-1} \tilde{k}_{\phi\phi} \\ \frac{\partial \tilde{m}}{\partial x_i} = \tilde{m}_i \quad (43)$$

In addition to derivative of piezoelectric matrices with respect to density, derivation of the piezoelectric coupling matrix with respect to polarization variable is also required

$$\frac{\partial \tilde{k}_{u\phi}}{\partial P_i} = 2 p_P (2 P_i - 1)^{p_P-1} x_i^{p_{u\phi}} \tilde{k}_{u\phi} \quad (44)$$

After calculation of sensitivities, the optimization variables can be updated in each iteration of optimization with the help of optimization algorithm which is the subject of the next section.

3.4 Optimization algorithms

For solving the optimization problem, there are several optimization algorithm like sequential linear programming (SLP), sequential quadratic programming (SQP), method of moving asymptotes (MMA) or the optimality criteria (OC) method. This latter one is more historical than the other methods and its application is more simple. However, MMA is more powerful in terms of solving multi-variable and multi-constraint optimization problems. In addition, the

convergence in MMA is more assured due to consideration of two past successive iterations during optimization.

Therefore, in this paper to solve the actuation problem, the OC method is implemented so the proposed code is self-working. However, the OC method has convergence problem in energy harvesting code due to coupling effect of piezoelectric material. In the upcoming two sections, the OC method and MMA and their implementation codes will be explained.

3.4.1 Optimality criteria method

Optimality criteria is a heuristic method to update the design variables in each element of design domain during each iteration of optimization. Here, there are two design variables for each element including the density and polarization. It is common for structural optimization with optimality criteria that the mutual influence of the design variables on each other and from element to element can be ignored (Hassani and Hinton 1998). Therefore, they can be updated separately from each other in each iteration of optimization. In fact, this is similar to the case of topology optimization by homogenization method where the length, width and rotation angle of the hole in a microstructure are optimized by OC separately in each iteration of optimization (Suzuki and Kikuchi 1991).

Here, the densities should be chosen such a way to respect the volume constraints. In this case, the following Karush–Kuhn–Tucker (KKT) condition for intermediate densities ($0 < x_i < 1$) should be satisfied to guarantee the convergence

$$\frac{\partial J}{\partial x_i} + \bar{\lambda} \frac{\partial V}{\partial x_i} = 0 \tag{45}$$

in which $\bar{\lambda}$ is the Lagrange multiplier to augment the volume constraint in the optimization. To solve the optimization problem mentioned in KKT (45), the OC algorithm given by Bendsoe and Sigmund (1995) and Bendsoe (2013) is used here which can be written as

$$x_i^{\text{new}} = \begin{cases} \max(0, x_i - move) & \text{if } x_i \beta_i^\eta \leq \max(0, x_i - move) \\ \min(1, x_i + move) & \text{if } x_i \beta_i^\eta \geq \min(1, x_i - move) \\ x_i \beta_i^\eta & \text{otherwise} \end{cases} \tag{46}$$

where *move* parameter is the maximum amount of density change in each iteration of optimization, η is a numerical damping coefficient and

$$\beta_i = -\frac{\partial J}{\partial x_i} \left(\bar{\lambda} \frac{\partial V}{\partial x_i} \right)^{-1} \tag{47}$$

Similar to the classical compliant problems of passive material (Bendsoe 2013), the values of *move* and η are considered to be 0.2 and 0.3 respectively.

For optimization of polarization there is no volume constraint. In fact, since the polarization of each element will be optimized separately, it is not necessary to force the optimization algorithm to push the polarization value to zero for the elements with low density values. Therefore, the optimization problem can be simply defined as follows

$$\frac{\partial J}{\partial P_i} = 0 \tag{48}$$

The algorithm for this simple optimization can be obtained by modifying the OC algorithm given in (46) in the following form

$$P_i^{\text{new}} = \begin{cases} \max(0, P_i - move) & \text{if } \frac{\partial J}{\partial P_i} \geq 0 \\ \min(1, P_i + move) & \text{if } \frac{\partial J}{\partial P_i} < 0 \end{cases} \tag{49}$$

By defining the optimization algorithm for polarization based on (49), the polarity value (P_i) for all of the elements will be steered to -1 or $+1$ even for the elements with very low density. However, this will not affect the optimization results since based on (25) and with maximum value of polarization (P), low density (x) will push the coupling matrix to its minimum value.

3.4.2 The method of moving asymptotes

The method of moving asymptotes (MMA) is a structural optimization method proposed by Svanberg (1987). The problem formulation of this methodology is as follows (Svanberg 2007)

$$\begin{aligned} &\text{minimize } f_0(\chi) + a_0 \underline{z} + \sum_{i=1}^{m_c} \left(c_i \underline{y}_i + \frac{1}{2} d_i \underline{y}_i^2 \right) \\ &\text{subject to } f_i(\chi) - a_i \underline{z} - \underline{y}_i \leq 0 \quad i = 1, \dots, m_c \\ &\quad \chi \in X, \quad \underline{y}_i \geq 0, \quad \underline{z} \geq 0 \end{aligned} \tag{50}$$

in which $X = \{ \chi \in \mathfrak{R}^{n_{\text{var}}} | \chi_i^{\min} \leq 0 \leq \chi_i^{\max} \}$, while n_{var} is the number of design variables, χ is the vector of all design variables, \underline{y}_i and \underline{z} are the artificial optimization variables, $f_0(\chi)$ is the cost function to be minimized, m_c is the number of constraints and a_0, a_i, c_i, d_i are the coefficients which have to be determined to match the optimization problem mentioned in (50) to different types of optimization problems.

For the optimization problem of this paper, there is only one constraint defined on the maximum volume of the optimized design. Therefore, based on the description given in Svanberg (2007), by considering $a_0 = 1$ and $a_i = 0$ for all i , then $\underline{z} = 0$ in any optimal solution, and by considering $d_i = 0$ and $c_i =$ “a large number,” then the variables $y_i = 0$

for all i and the optimization problem mentioned in (50) will be matched to the optimization problem of this paper.

Finally, the OC and MMA optimization algorithms can update the optimization variables in each iteration of optimization. To do so, the implementation code for both of OC and MMA will be given in Section 4.

3.5 Filtering

Like other problems in structural optimization, topology optimization of piezoelectric materials also suffers from numerical instabilities such as checker board problem or mesh dependency. To remedy, the solutions need to be filtered in each iteration of optimization. So far, many filtering methods are suggested in the literature (Sigmund 2007). Among the proposed filtering methods, sensitivity filter and density filter proved their success in overcoming the aforementioned numerical instabilities. The sensitivity filter is used in the 99 lines of topology optimization code written by Sigmund (2001) and density filter is used as an alternative option in the 88 lines of topology optimization code written by Andreassen et al. (2011). As such, in this paper, the density and sensitivity filters are used as two available options. To implement these filters, the same lines of codes written in 88 lines of code by Andreassen et al. (2011) are employed. Therefore, the concepts behind these filters will not be explained here since the related explanations can be found in the mentioned references.

The important point here is that the density filter shows more promising performance in comparison with sensitivity filter specially in the case of energy harvesting problem with using the MMA as the optimization algorithm.

Finally, the filters will be applied only to the densities and filtering of the polarization variable is not necessary.

3.6 Choosing proper penalization factors

After preparing the ingredients of topology optimization algorithm, the important question would be how to tune the parameters of optimization? in addition to parameters which are the same for optimization of passive and active materials like radius of filtering and volume fraction, there are other parameters which belong to multiphysics nature of piezoelectric topology optimization like penalization coefficients of the piezoelectric matrices, i.e., $p_{u\phi}$, $p_{\phi\phi}$ and p_P . In fact by choosing the penalization coefficients, we are pursuing two goals: (1) guaranteeing the convergence to perfect void-material in the final obtained layout, (2) avoiding local optima. For choosing p_{uu} , it is already proven that the value of 3 is the best choice (Bendsoe 2013) to reach a perfect void/material in the final layout.

There are several studies which focused on defining the criteria for choosing other penalization coefficients, i.e., $p_{u\phi}$, $p_{\phi\phi}$ and p_P (Kim and Shin 2013; Noh and Yoon 2012; Kim et al. 2010). In particular, Noh and Yoon (2012) chose the penalization factors randomly and obtained different density layouts. The final conclusion was that the penalization factors have extreme effect on the final topology. But, no criteria or rule is presented in this study about the method of choosing the penalization factors. On the other hand, a detailed study is presented by Kim et al. (2010) for choosing the penalization factors. Based on this study, necessary condition for choosing the penalization factors is that the electromechanical coupling coefficient (EMCC) should be increased when the density (x) increases and vice versa. Based on this condition, the following intrinsic condition which is independent from objective function is proposed for choosing the penalization factors (Kim et al. 2010)

$$2p_{u\phi} - (p_{uu} + p_{\phi\phi}) > 0 \quad (51)$$

In addition to this condition, Kim et al. (2010) proposed other objective dependent criteria for the actuation and energy harvesting application. In particular, for the actuation objective function in (26), the following condition is proposed (Kim et al. 2010),

$$p_{u\phi} - p_{uu} > 0 \quad (52)$$

and for energy harvesting approach, when the objective function is energy conversion factor then the following conditions should be satisfied (Kim et al. 2010),

$$\begin{aligned} p_{u\phi} - p_{uu} &> 0 \\ p_{\phi\phi} - p_{uu} &> 0 \end{aligned} \quad (53)$$

It should be noted that these conditions are proposed to guarantee the convergence of the final topology and yet there is no study on the methods to define these penalization factors to avoid the local optima.

For polarization penalization, the value of 1 seems the best choice as it is suggested by Kögl and Silva (2005). Indeed, it is not necessary to penalize the polarization.

4 MATLAB implementation codes

In this section, the goal is to establish the MATLAB code for the piezoelectric optimization methodology explained in the previous section. Two MATLAB codes are mentioned in the Appendix of the paper. The first code is for actuation and the second code is for energy harvesting. Each MATLAB code is partitioned so the readers can have a perception of the goal of each part. The important lines of codes are also labeled

to provide a connection between the line and the analytical calculation. Again, the rest of this section will be divided for actuation and energy harvesting codes and in each section different parts of the codes will be explained more in detail.

4.1 Actuation

4.1.1 General definition

The first part of the code is GENERAL DEFINITION. This part consists of geometrical dimensions of the piezoelectric plate, the resolution of the mesh, i.e., the number of elements in the \vec{x} and \vec{y} axes, the penalization factors, the type of filter, the filter radius and maximum number of iterations and the stiffness of the attached spring are defined. It should be noted that two stopping criteria are defined by the codes. The classical criteria is the density change between two last successive iterations. The other criteria is the maximum number of iterations. Satisfying either of these will stop the optimization. The reason behind adding the second criteria is that in contrast to pure mechanical problems, here we have different types of material and different types of objective functions. As such, the density change will not stop the optimization generally or it needs very high number of iterations while there are no significant oscillations in the objective function's value.

4.1.2 Material properties

The second part of the code is MATERIAL PROPERTIES. In this part of the code, the properties of the chosen piezoelectric material are given. These properties are the density, e_{31}^* coupling coefficient, ϵ_{33}^* permittivity coefficient and the elements of the stiffness tensor all after applying the plane stress assumption as mentioned in (4). The chosen material for the code is the PZT 4 which is popular in the literature. There are PZT materials with lower and higher coupling coefficients. If one wants to investigate the results of other piezoelectric materials, the values of this part can be changed.

It should be noted that choosing different PZT materials in this section will not affect the final layout obtained by optimization algorithm. In fact the resulted layout of the optimization is independent from Piezoelectric coefficients.

4.1.3 Finite element model

In the section called PREPARE FINITE ELEMENT ANALYSIS, the proposed code for the finite element model of the piezoelectric plate is the extension of the MATLAB code provided by Kattan (2010) for the bilinear quadrilateral elements of the passive materials. However, instead of using

analytical calculations, elemental matrices are calculated by using the two-point Gauss quadrature method. To do so, first the geometrical sizes of each element are calculated based on the defined geometry of the design domain and desired resolution of mesh. Then, Gauss quadrature points are defined in matrix GP. Afterwards, the elemental matrices are found inside a loop with the help of Gauss points as follows,

```
47 kuu = kuu+h*J*transpose(Bu)*C*Bu; %
    Mechanical stiffness matrix:
48 kup = kup+ h*J*transpose(Bu)*e'*Bphi; %
    Piezoelectric coupling matrix:
```

In this method of elemental matrix calculation, there is no analytical integration, the calculation time is very fast and at the same time the code is flexible in terms of different geometrical dimensions and mesh resolution and aspect ratios.

After calculating the elemental matrices, normalization is applied based on (20) and normalization factors are obtained with the following lines of code

```
50 k0 = max(abs(kuu(:))); alpha = max(kup(:));
    % Normalization Factors
51 kuu = kuu/k0; kup = kup/alpha; %
    Normalization
```

It should be noted that for simplicity, the non-normalized matrices are replaced by normalized matrices. Since the normalization factors are saved, it is easy to find the real values after the calculation of the final results.

So far, the elemental matrices are calculated, normalized and the normalization factors are derived. For assembling the elemental matrices, the element connectivity matrix known as edofMat (Andreassen et al. 2011) is built with the help of the numbering format shown in Fig. 2. In this figure, the coarse discretization is shown with the numbering of elements, nodes and mechanical and electrical degrees of freedom. This numbering format is similar to the one presented in 99 (Sigmund 2001) and 88 (Andreassen et al. 2011) lines of MATLAB codes for structural problems. The numberings of the nodes and mechanical degrees of freedom are from top to bottom and from left to right. It should be noted that inside each element, the numbering order is different as it is shown in Fig. 2b. The numbering inside each element is counterclockwise and it starts from bottom left and the sequence of numbers in each row of edofMat matrix is following this numbering order. Furthermore, since the system here is having additional degree of freedom as potential, a vector of potential connectivity is defined as edofMatPZT. In fact, by considering one potential degree of freedom for each node, potential connectivity would be a

matrix with rows containing the node IDs. However, due to equipotential condition, one potential degree of freedom is considered for each element and the potential connectivity is a vector. At last, the connectivity matrix `edofMat` and the potential connectivity vector `edofMatPZT` is defined as

$$\begin{array}{ccc}
 \text{edofMatPZT} & \text{edofMat} & \text{Element} \\
 & & \text{Number} \\
 \left[\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \end{array} \right] & \left[\begin{array}{cccccccc} 3 & 4 & 11 & 12 & 9 & 10 & 1 & 2 \\ 5 & 6 & 13 & 14 & 11 & 12 & 3 & 4 \\ 7 & 8 & 15 & 16 & 13 & 14 & 5 & 6 \\ 11 & 12 & 19 & 20 & 17 & 18 & 9 & 10 \\ 13 & 14 & 21 & 22 & 19 & 20 & 11 & 12 \\ 15 & 16 & 23 & 24 & 21 & 22 & 13 & 14 \\ 19 & 20 & 27 & 28 & 25 & 26 & 17 & 18 \\ 21 & 22 & 29 & 30 & 27 & 28 & 19 & 20 \\ 23 & 24 & 31 & 32 & 29 & 30 & 21 & 22 \\ 27 & 28 & 35 & 36 & 33 & 34 & 25 & 26 \\ 29 & 30 & 37 & 38 & 35 & 36 & 27 & 28 \\ 31 & 32 & 39 & 40 & 37 & 38 & 29 & 30 \end{array} \right] & \left[\begin{array}{l} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 5 \\ \leftarrow 6 \\ \leftarrow 7 \\ \leftarrow 8 \\ \leftarrow 9 \\ \leftarrow 10 \\ \leftarrow 11 \\ \leftarrow 12 \end{array} \right.
 \end{array} \quad (54)$$

The implementation lines to create the `edofMat` matrix and `edofMatPZT` vector with the help of `nodeIDs` are

```

54 nodenrs = reshape(1:(1+nelx)*(1+nely), 1+
    nely, 1+nelx);
55 edofVec = reshape(2*nodenrs(1:end-1, 1:end
    -1)+1, nele, 1);
56 edofMat = repmat(edofVec, 1, 8)+repmat([0 1
    2*nely+[2 3 0 1] -2 -1], nele, 1);
57 edofMatPZT = 1:nele;

```

where `nele` is the number of elements, `nodenrs` is the matrix of node numbers and `edofVec` is the vector of `nodeIDs` containing the first `nodeID` of each element. Now, with the `edofMat` and `edofMatPZT`, it is possible to assemble the elemental matrices and build the global matrices with following lines of code

```

58 iK = kron(edofMat, ones(8, 1))';
59 jK = kron(edofMat, ones(1, 8))';
60 iKup = edofMat';
61 jKup = kron(edofMatPZT, ones(1, 8))';

109 sKuu = kuu(:)*(Emin+xPhys(:)'.^penalKuu*(
    E0-Emin));
110 sKup = kup(:)*(eMin+xPhys(:)'.^penalKup*(
    e0-eMin).*(2*pol(:)-1)'.^penalPol));
111 Kuu = sparse(iK, jK, sKuu); % Global
    stiffness matrix
112 Kup = sparse(iKup(:), jKup(:), sKup(:)); %
    Global piezoelectric coupling matrix

```

The matrices `iK` and `jK` corresponds to (i, j) entry of stiffness matrix for each element. With the same strategy, `iKup` and `jKup` are written for the i th and j th entry of $K_{u\phi}$

matrix. To avoid redundancy, there is no `iKpp` and `jKpp`. In fact, these latter are just equal to the `edofMatPZT`.

The material interpolation scheme mentioned in (25), is applied in `sKuu` and `sKup` matrices. `xPhys(:)` and `pol(:)` are the vectorized physical density and polarization matrices respectively which will be updated in each iteration of optimization.

4.1.4 Boundary condition

The mechanical boundary conditions are defined in the part `DEFINITION OF BOUNDARY CONDITION`. In this part, `fixeddofs` and `freedofs` are defined to contain the fixed degrees of freedom and the free degrees of freedom respectively. They are defined by using the method reported in 88 lines (Andreassen et al. 2011) and 99 lines (Sigmund 2001) of code. Therefore, for applying the clamped boundary condition on the left side of the design domain, the following lines of code can be used,

```

70 fixeddofs1 = 1:2*(nely+1); % Main supports

```

By considering that for the coarse mesh in Fig. 2, `nelx` = 4 and `nely` = 3, `fixeddofs` in the aforementioned implementation line produce the numbers from 1 to 8 which are the mechanical degrees of freedom in the left side of the design domain.

4.1.5 Output displacement definition

In this part of the code for actuation, the goal is to define the particular point of the design domain where the maximization of displacement in a particular direction is desired. To do so, the desired mechanical degree of freedom should be defined. The variable `DMDOF` which is the abbreviation of desired mechanical degree of freedom is defined for this purpose. Thereafter, the vector `L` is created. This vector will be used later in the objective function and sensitivity analysis for optimization.

A spring is attached at `DMDOF` which simulates the reaction force exerted by an imaginary object. The modeled spring will modify the piezoelectric stiffness matrix with this line of code

```

113 Kuu(DMDOF, DMDOF) = Kuu(DMDOF, DMDOF)+Ks; %
    Assembling the stiffness of the modeled
    spring

```

where `Ks` is the stiffness of the modeled spring. Actually, by changing the stiffness of the modeled spring, it is possible to determine whether more force is desired or more displacement. Indeed, since the stiffness matrix is normalized, the stiffness of the modeled spring can be determined with respect to the stiffness of the piezoelectric plate. For example, by putting the stiffness of the spring

equal to 1, the stiffness will be equal to the highest value of the piezoelectric stiffness matrix. In this case, the piezoelectric layout will be optimized to produce maximum possible of force. However, by defining very low values of stiffness for the spring (i.e., 0.01), then the piezoelectric layout will be optimized for maximum possible of deflection.

4.1.6 Objective function

Calculation of objective function for actuator is a routine procedure to calculate the desired displacement in each iteration. To do so, first the mechanical displacement due to applied voltage should be calculated. Therefore, based on (21), the mechanical displacement vector is calculated by the following line of code

```
114 Uu(freedofs, :) = Kuu(freedofs, freedofs) \ (-
      Kup(freedofs, :) * Up); % Mechanical
      displacement
```

then based on (36), the objective function can be calculated by

```
116 CE = -(sum(L(edofMat) .* Uu(edofMat), 2));
117 c = full(sum(CE)); % Objective Function
```

4.1.7 Sensitivity analysis

For sensitivity analysis, the first step is the calculation of adjoint vectors. It is calculated based on (28),

```
118 Adjoint(freedofs, :) = Kuu(freedofs,
      freedofs) \ L(freedofs, :); % Adjoint
      vector
```

then the calculation of sensitivities based on (29) and (30) starts afterwards

```
119 DCKuuE = sum((Adjoint(edofMat) * kuu) .* Uu(
      edofMat), 2);
120 DCKupE = (Adjoint(edofMat) * kup) .* Up(
      edofMatPZT);
121 DCKuu = reshape(DCKuuE, [nely, nelx]);
122 DCKup = reshape(DCKupE, [nely, nelx]);
123 dc = penalKuu * (E0 - Emin) * xPhys .^ (penalKuu
      - 1) .* DCKuu + ...
124 penalKup * (E0 - Emin) * ((2 * pol - 1) .^ (penalPol))
      .* xPhys .^ (penalKup - 1) .* DCKup; %
      sensitivity with respect to x
125 dp = 2 * penalPol * ((2 * pol - 1) .^ (penalPol - 1))
      .* xPhys .^ (penalKup) .* DCKup; %
      sensitivity with respect to p
126 dv = ones(nely, nelx); % Volume
      sensitivity
```

where dc and dp are the sensitivities of the objective function with respect to x and P and dv is the sensitivities of the volume constraint with respect to x . It is obvious from

(26) that the sensitivity of volume constraint with respect to P is zero.

As can be seen in the aforementioned lines of the code, the sensitivities containing the stiffness (kuu) and coupling matrix (kup) are calculated separately. This will help us to define different penalization coefficients for each of these matrices.

4.1.8 Optimization algorithm

The implementation lines of code for the OC update of densities are the same as what is mentioned in 88 lines of code (Andreassen et al. 2011). In addition to densities, for updating the polarization based on algorithm (49), there is no need for the bi-sectioning loop and just one line of code can optimize the polarization in each iteration of optimization as follows

```
145 pol = max(0, max(pol - move, min(1., min(pol +
      move, sign(-dp))))); % OC update of
      polarization
```

4.1.9 Plot densities and polarization

In part (PLOT DENSITIES & POLARIZATION), the densities and polarization will be plotted in two figures separately. As it is common, in the density figure, white area means no material while the black area means material.

Different color spectrum is chosen for polarization profile in which the red and blue color shows opposite direction of optimization while the green color shows the neutral material with no polarization. The important point here is that the OC will steer the polarization value to 0 or 1 even for the elements with lower density. Therefore, here to eliminate the confusion, the matrix of densities is multiplied to the polarization. In this way, for element with minimum density (no material), the polarization turns to green as well.

4.1.10 Filtering

The density filtering lines of code are similar to 88 lines of code (Andreassen et al. 2011). Two types of density and sensitivity filter can be chosen. In general, the density filter is more recommended. However, since the code is written for both of these filters, the best choice will be up to readers.

4.2 Energy harvesting

4.2.1 General definition

In general definition part of the energy harvesting code, in addition to what is mentioned for the actuation code,

the objective function's weighing factor (w_j) and excitation frequency (ω) are defined.

4.2.2 Material properties

This part of the energy harvesting code is similar to the actuation code. However, the results of actuation are free from the PZT coefficients while in the case of energy harvesting for different materials and different piezoelectric coefficients, the results will be changed. PZT materials with higher coupling coefficients produce more electrical energy in comparison with ones with lower coupling coefficients. At the same time, PZT materials with higher coupling coefficients have more coupling effect which affect the optimization during the iteration and can change the final layout. Similar to actuation code, PZT 4 is chosen for the code.

4.2.3 Finite element model

In comparison with FE model presented in the actuation code, the FEM part of the energy harvesting is the same with some additional calculations. Indeed, the elemental matrices for mass and dielectric matrices are calculated as well in the following lines of code,

```
52 kpp = kpp + h*J*transpose(Bphi)*ep33*Bphi;
    % Dielectric stiffness matrix
53 N = [n1, 0, n2, 0, n3, 0, n4, 0; 0, n1, 0, n2, 0, n3, 0,
    n4]; % Matrix of interpolation
    functions
54 m = m+J*ro*h*(N')*N; % Mass matrix
```

For assembling the global mass matrix M , the same matrices of iK and jK as described in the actuation code can be used. The assembling lines of mass matrix and dielectric matrix $\tilde{K}_{\phi\phi}$ can be found in energy harvesting code as

```
129 sM = m(:)*xPhys(:)';

132 sKpp = kpp(:)*(epsMin+xPhys(:).^penalKpp
    *(eps0-epsMin));

134 M = sparse(iK(:), jK(:), sM(:)); % Global
    masss matrix

137 Kpp = sparse(edofMatPZT(:), edofMatPZT(:),
    sKpp(:)); % Global dielectric stiffness
    matrix
```

where the related material interpolation scheme is applied in sM and $sKpp$.

Furthermore, the stiffness matrix is modified by the dynamic matrix $-\omega^2 M$ which represents $(-\tilde{M}\Omega^2)$ in

(23). This modification is done by the following line of code

```
135 Kuu = sparse(iK, jK, sKuu) - omega*M;
```

After building the global piezoelectric matrices, the equipotential condition is applied with the following line of code

```
138 KupEqui = Kup(freedofs,:) * B; KppEqui = B' *
    Kpp * B; % Equipotential Condition
```

where $KupEqui$ and $KppEqui$ are the piezoelectric coupling and dielectric matrices after applying the mechanical and equipotential condition which represent $\bar{K}_{u\phi}$ and $\bar{K}_{\phi\phi}$.

4.2.4 Objective function

To find the value of objective function, the mechanical displacement and resulted potential due to applied force should be calculated based on (23). But, first a total matrix is built based on the left-hand side of (23). The symmetry is guaranteed afterward, and finally, the mechanical displacements and potential are calculated

```
139 Ktot = [Kuu(freedofs, freedofs), KupEqui;
    KupEqui', -gamma*KppEqui]; % Creation
    of total matrix with equipotential
    hypothesis
140 Ktot = 1/2*(Ktot + Ktot'); % Numerical
    symmetry enforcement
141 U = Ktot\Ftot; % Response vector
```

After calculation of mechanical displacement and potential, for calculation of objective function, W_m and W_e are defined for mechanical and electrical energies respectively.

```
155 Wm = Wm+ reshape(sum((Uu_i(edofMat)*kuu).*
    Uu_i(edofMat), 2), nely, nelx);
156 We = We+ reshape(sum((Up_i*kpp).*Up_i, 2),
    nely, nelx);
```

```
168 Wm = sum(sum((Emin+xPhys.^penalKuu*(E0-
    Emin)).*Wm)); % Mechanical energy
169 We = sum(sum((epsMin+xPhys.^penalKpp*(eps0-
    epsMin)).*We)); % Electrical energy
170 c = wj*Wm - (1-wj)*We; % Objective function
```

It is important to note that in the written code, the mechanical and electrical energies are calculated by summing the energies in each element. That is why the mechanical displacement and electrical potentials are multiplied to elemental stiffness matrix kuu and elemental dielectric matrix kpp . Then, the objective function for the energy harvester is the weighted sum of the energies while considering the weighting factor (w_j).

4.2.5 Sensitivity analysis

Sensitivity analysis starts by calculating adjoint vectors with the help of (40)

```

143 ADJ1 = Ktot\[-Kuu(freedofs, freedofs)*Uu(
    freedofs, :); zeros(1, nf)]; % First
    adjoint vector
144 lambda1(freedofs, :) = ADJ1(1:lf, :); mu1 =
    B*ADJ1(1f+1:end, :);
145 ADJ2 = Ktot\[zeros(1f, nf); -KppEqui*Up]; %
    Second adjoint vector
146 lambda2(freedofs, :) = ADJ2(1:lf, :); mu2 =
    B*ADJ2(1f+1:end, :);

```

Here, B is the Boolean matrix as defined in (41). In fact, μ_1 and μ_2 are vectors of equal values for all of the elements in the design domain due to equipotential condition. For sensitivity analysis, the adjoint vectors related to mechanical and electrical states should be separated. For this reason, ADJ resolved to λ and μ .

After calculation of adjoint vectors, the sensitivity analysis is performed by the following lines of code

```

151 for i = 1:nf % nf is the total number of
    forces
152 Uu_i = Uu(:, i); Up_i = B*Up(:, i);
153 lambda1_i = lambda1(:, i); lambda2_i =
    lambda2(:, i);
154 mu1_i = mu1(:, i); mu2_i = mu2(:, i);
155 Wm = Wm+ reshape(sum((Uu_i(edofMat)*kku). *
    Uu_i(edofMat), 2), nely, nelx);
156 We = We+ reshape(sum((Up_i*kpp). *Up_i, 2),
    nely, nelx);
157 dcKuuE = wj*(((1/2)*Uu_i(edofMat) +
    lambda1_i(edofMat))*kku). *Uu_i(edofMat)
    ) - (1-wj)*((lambda2_i(edofMat)*kku). *
    Uu_i(edofMat));
158 dcKupE = wj*(lambda1_i(edofMat)*kup). *
    Up_i + ((Uu_i(edofMat))*kup). *mu1_i
    - (1-wj)*((lambda2_i(edofMat))*kup). *
    Up_i + ((Uu_i(edofMat))*kup). *mu2_i);
159 dcKppE = wj*((-mu1_i*kpp). *Up_i) - (1-wj)
    *((1/2)*(Up_i*kpp). *Up_i - (mu2_i*kpp)
    . *Up_i);
160 dcME = wj*(((1/2)*Uu_i(edofMat) +
    lambda1_i(edofMat))*(-m*omega)). *Uu_i(
    edofMat) - (1-wj)*((lambda2_i(edofMat)
    . *(-m*omega)). *Uu_i(edofMat));
161 dcKuu = reshape(sum(dcKuuE, 2), [nely, nelx])
    ;
162 dcKup = reshape(sum(dcKupE, 2), [nely, nelx])
    ;
163 dcKpp = gamma*reshape(sum(dcKppE, 2), [nely,
    nelx]);
164 dcM = reshape(sum(dcME, 2), [nely, nelx]);
165 dc = dc + penalKuu*(E0-Emin)*xPhys.^(
    penalKuu-1). *dcKuu+penalKup*(e0-eMin)*
    xPhys.^(penalKup-1). *dcKup.*(2*pol-1)
    .^(penalPol))+penalKpp*(eps0-epsMin)*
    xPhys.^(penalKpp-1). *dcKpp+dcM;
166 dp = dp + (e0-eMin)*2*penalPol*((2*pol-1)
    .^(penalPol-1)). *xPhys. ^penalKup.*
    dcKup;% Polarization sensitivity
167 end

```

The code is written in a general form to consider nf number of load cases. Therefore, for each number of load cases the mechanical displacement, potential and adjoint vectors are separated by defining the Uu_i , Up_i , λ_{1_i} , λ_{2_i} , μ_{1_i} and μ_{2_i} . In addition, mechanical displacement, potential and adjoint vectors are calculated in global format while the sensitivity analysis should be performed on the elemental scale. To do so, $edofMat$ is utilized to convert the vector of mechanical displacement to the matrix $Uu_i(edofMat)$ in which each row is related to one element while the columns represent the mechanical displacement of that element. The same idea is also applied to the adjoint vectors, i.e., $\lambda_{1_i}(edofMat)$ and $\lambda_{2_i}(edofMat)$. On the other hand, due to equipotential condition, it is obvious that the potential of each element is equal to Up , and as mentioned before, values of μ_1 and μ_2 are equal for all elements.

Similar to the actuation code, here the sensitivities containing each piezoelectric matrix are calculated separately to facilitate the definition of different penalization factors.

4.2.6 Optimization algorithm

In the energy harvesting code, the MMA optimization algorithm is used. However, this code cannot be executed without having the external MMA code. The MMA implementation MATLAB code can be obtained by contacting Prof. Krister Svanberg. In this way, the interested readers will obtain two MATLAB codes to implement the MMA optimization code which includes `mmasub.m` and `subsolv.m`. The version 2007 of these codes is used in this paper. Supposing that these two MATLAB codes are already available, to implement the MMA, first initial parameters are defined in MMA Preparation part based on what is mentioned in Section 3.4.2. Then, optimization of variables will be done in MMA OPTIMIZATION OF DESIGN VARIABLES part. The important point in this part is that, due to the scale difference between the density sensitivities and polarization sensitivity, it is suggested here to normalize the polarization sensitivity with the following line of code

```

180 dp = dp/max(abs(dp(:))); % Normalizing the
    polarization sensitivity

```

Then, the main lines of MMA optimization implementation are

```

186 [xm, ~, ~, ~, ~, ~, ~, ~, ~, low, upp] =
    mmasub(mc, nVar, loop, xval, xmin, xmax
    , xold1, xold2, ...
187 f0val, df0dx, fval, dfdx, low, upp, a0, ai, ci, di)
    ; % MMA optimization

```

The output `xmma` is the vector of all optimization variables. Therefore, the upper half of this vector is the updated density and lower half is the polarization values.

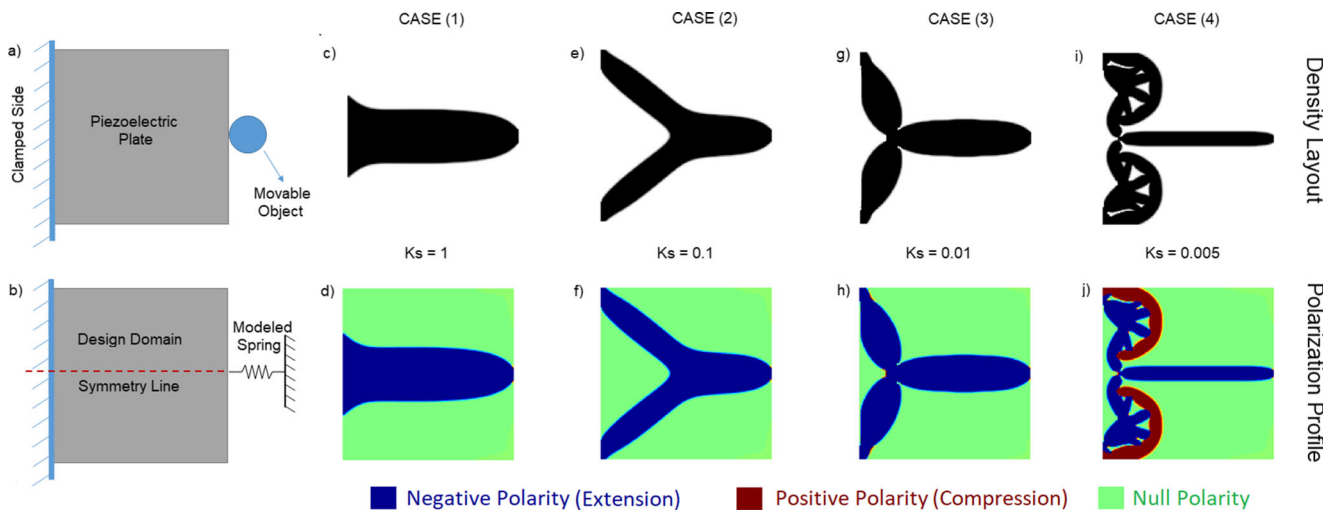


Fig. 3 Topology optimization of a piezoelectric actuator (pusher) for different stiffness of the modeled spring. Panel **a** presents the specification for a practical use and panel **b** is the mechanical model for implementation in the finite element software. Panels **c**, **e**, **g**, **i**

and **d**, **f**, **h**, **j** respectively present the density and polarization profile of the design for the specified output stiffness after convergence. For the polarization profile, blue, red and green represent respectively negative, positive and null polarization

After updating the optimization variables and plotting the results, this iteration of optimization will be finished and the optimization will be started from the beginning of the loop for the next iteration.

5 Numerical examples

In this section, the goal is to investigate the performances of the codes in different application cases of actuation and energy harvesting with different configurations. First, different examples of actuation will be investigated. Thereafter, different configurations of energy harvesting are explored.

5.1 Actuation

5.1.1 Pusher

The first example in the actuation part is a simple pusher as can be seen in Fig. 3a. The gray area shows the design domain which can be optimized by the optimization algorithm. The actuation optimization code which is mentioned in the Appendix is written for this example. As it can be seen from the code, the aspect ratio of the elements in the x and y directions is following the aspect ratio between the length and width of the plate which produces square elements for discretization of the design domain. This is not mandatory, but it is known that increasing the aspect ratio of

width and length of each element increases the inaccuracy of the finite element model (Logan 2000). Therefore, it is recommended to follow the aspect ratio of the plate in defining the number of elements in x and y directions.

The chosen penalization factors are $p_{uu} = 3$ and $p_{u\phi} = 4$, which satisfy the conditions mentioned in (52). These penalization factors are the same for all of the actuation examples.

For having a completely symmetrical response with respect to the horizontal dotted line in Fig. 3a and to decrease the number of elements in the design domain, the defined design domain in the code is the upper half of the piezoelectric plate. Therefore, in the symmetry line of the design domain, the roller mechanical boundary condition is applied with the following line of code,

```
71 fixeddofs2 = 2*[(nely+1):(nely+1):(nely+1)
    *(nelx+1)]; % Applying symmetry
```

With this mechanical boundary condition, the nodes connected to the symmetry line can have displacement in the x direction but not in the y direction.

As can be seen in Fig. 3, the results of topology optimization for different spring stiffness are plotted. The upper row of the figure shows the density layout while the lower row shows the polarization profile. By mirroring the obtained result with respect to the symmetry line, the results are illustrated for whole piezoelectric plate.

The numerical results for different spring stiffness are also reported in Table 1. The numerical results show the

Table 1 Displacement amplification ratio of optimized actuators with respect to full plate

	Pusher		Gripper	
	k_s	Amplification ratio	k_s	Amplification ratio
CASE (1)	1	0.76	1	3.95
CASE (2)	0.1	0.99	0.1	11.42
CASE (3)	0.01	1.54	0.01	34.04
CASE (4)	0.005	2.75	0.003	60.11

amplification ratio of the optimized design with respect to the full plate under application of same value of voltage. The objective value which is reported by the code is not showing this amplification ratio. To calculate the amplification ratio, the final value of the objective function after finishing the optimization should be divided by the objective function value of the full plate. To find the objective function of the full plate, it is possible to define the initial values of density equal to 1. To do so, the following line of code

```
97 x = repmat(volfrac,nely,nelx); % Initial
    values for density ratios
```

put 1 instead of volfrac, then stop the code after the calculation of objective function. In this case, the value of objective function for the full plate is obtained.

From the plots in Fig. 3c and d, it is clear that when the spring stiffness is one, the optimal layout is a very simple lumped design with uniform polarization profile. Based on Table 1, for this case, the full plate is having more displacement. On the other hand, by considering very low stiffness ($K_s = 0.005$) and then based on Fig. 3i and j, the density layout is more complicated and polarization profile is not uniform anymore. In fact, it is obvious that the blue region in the polarization profile will have extension while the red part will have compression (shrinkage). The combination of this extension and compression will produce an amplification ratio with respect to full plate equal to 2.75 as reported in Table 1.

5.1.2 Gripper

The second example of piezoelectric actuation is a gripper, which is similar to the case discussed in Ruiz et al. (2017). The goal is to design a gripper to grab an object as it is shown in Fig. 4a and b. To do so, some modifications should

be done to the actuation code in the Appendix. First of all, the OUTPUT DISPLACEMENT DEFINITION part should be completely changed by replacing the following lines

```
%% OUTPUT DISPLACEMENT DEFINITION
L_grip=(ceil(nelx/5));
W_grip=(ceil(nely/8));
DMDOF = ndof-2*W_grip;
L = sparse(2*(nely+1)*(nelx+1),1); %
    Initialization of adjoint vector force
L(DMDOF,1) = 1; % Setting values in
    adjoint vector force
Uu = zeros(ndof,1); % Creation of null
    displacement vector
Adjoint = zeros(ndof,1); % Creation of
    null adjoint vector
Up(1:nele,1)= 1;% Actuation voltage
```

where L_{grip} and W_{grip} are the length and width of the empty box in the piezoelectric plate as shown in Fig. 4a and b.

Next, to enforce zero material in the desired box of the design domain, the passive elements should be defined. The strategy is the same as in 99 lines (Sigmund 2001) and in 88 lines (Andreassen et al. 2011) of code. The following part should be added after the part INITIALIZE ITERATION and before the part START ITERATION

```
%% PASSIVE ELEMENTS
passive = zeros(nely,nelx);
for i = 1:nelx
for j = 1:nely
if i > nelx-L_grip
if j > nely-W_grip
passive(j,i)=1;
end
end
end
end
x(find(passive)) = 0.001;
```

Then to apply the passive material in each iteration, the following line should be added after the OC update line

```
137 xnew = max(0.001,max(x-move,min(1.,min(x+
    move,x.*(max(1e-30,-dc./dv/lmid)
    .^0.3))))); % OC update of density
xnew(find(passive)) = 0.001;
```

Now by executing the code, the results of Fig. 4 for different spring stiffness will be obtained and the numerical results are reported in Table 1. It is interesting to note that for the gripper, the amplification ratios in optimized designs are much higher than the amplification ratios for optimized pushers. Indeed, the polarization optimization plays a major role in designing the gripper. That is why for any chosen values of K_s , the polarization profile is not uniform and the

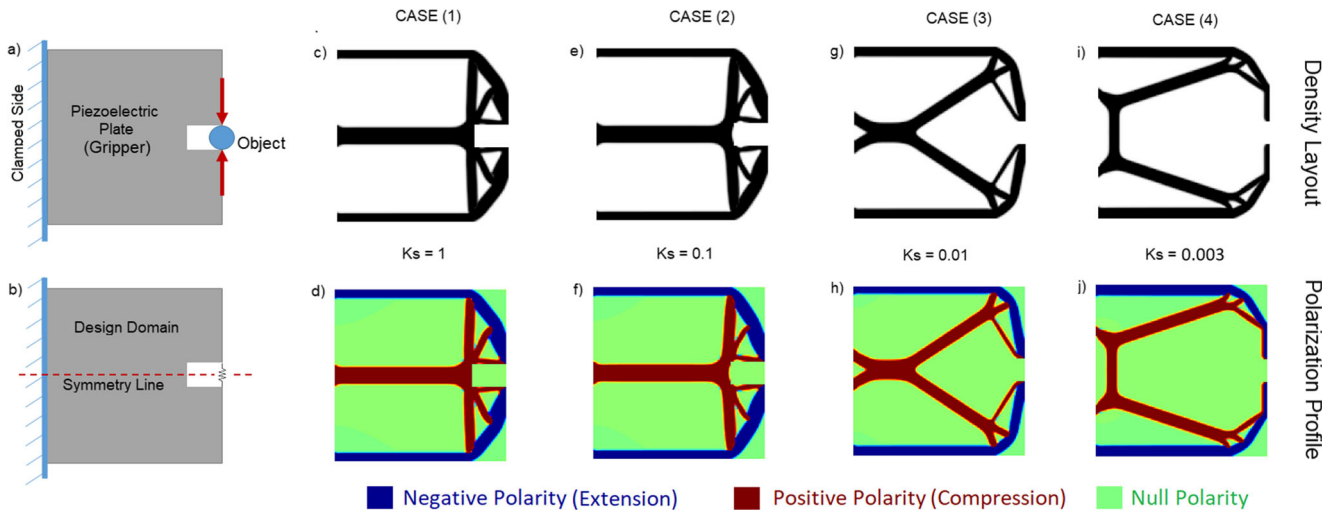


Fig. 4 Topology optimization of a piezoelectric gripper for different stiffness of the modeled spring. The panels follow the same presentation as Fig.3

gripper needs the combination of expansion and retraction for increasing the amplification ratio.

5.2 Energy harvesting

5.2.1 Lateral force

The first example of the energy harvesting code is a plate under a lateral force excitation as it is shown in Fig. 5. The code in the Appendix is written for this case. Here, the goal is to maximize the output electrical energy while minimizing the mechanical energy of the system. For this case, the problem is static and the excitation frequency is considered to be zero. The chosen penalization factors

for the energy harvesting code in contrast to actuation part are not the same for all cases. As such in Table 2, the penalization factors are reported for each case. But, for all cases, the chosen penalization factors are satisfying the conditions mentioned in (51) and (52). The reason for different penalization factors for each case is that the energy harvesting optimization is more complicated than actuation due to existence of the coupling effect. This coupling effect is highly affected by the chosen penalization factors in particular $p_{u\phi}$ and $p_{\phi\phi}$. By choosing proper penalization factors, it is possible to avoid the nonsymmetric results or to improve the convergence.

The results of optimization for different values of the weighting factor (w_j) are illustrated in Fig. 5. For the first

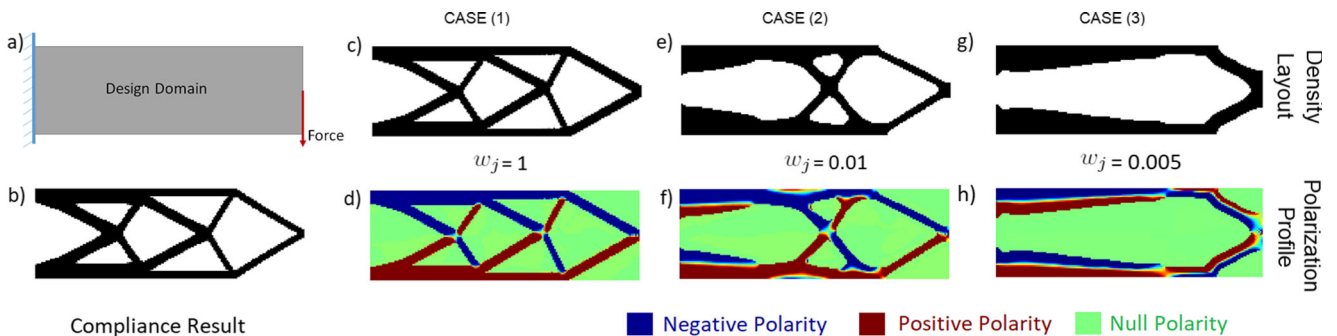


Fig. 5 Topology optimization of a piezoelectric energy harvester under application of a lateral static force for different values of w_j . Panel a presents the specification for a practical use and panel b is the result that can be obtained through classical compliance optimization. Panels

c, e, g, and d, f, h respectively present the density and polarization profile of the obtained design. For the polarization profile, blue, red and green represent respectively negative, positive and null polarization

Table 2 Output energies of the optimized designs for different weighting factors

	w_j	Π^S	Π^E	p_{uu}	$p_{u\phi}$	$p_{\phi\phi}$	p_P	Ω	$x_{\{0\}}$	MMA_move
Lateral force										
Compliance	1	116.66	0.00	3	6	4	0	0	volfrac	1
CASE (1)	1	99.90	1.25	3	6	4	1	0	volfrac	1
CASE (2)	0.01	152.87	1.70	3	4	4	1	0	volfrac	1
CASE (3)	0.005	741.09	7.77	3	4	4	1	0	volfrac	1
CASE (4)	1	106.04	1.38	3	6	6	1	1 kHz	volfrac	1
CASE (5)	0.01	123.15	1.43	3	6	6	1	1 kHz	volfrac	0.1
CASE (6)	0.02	240.50	2.63	3	6	4	1	3.5 kHz	volfrac	0.1
2 load case										
Compliance	1	106	0.03	3	6	6	0	0	1	1
CASE (1)	0.005	168.54	1.20	3	6	6	1	0	1	1
CASE (2)	1	107.88	0.86	3	6	6	1	1 kHz	1	0.1
CASE (3)	0.005	163.43	1.04	3	6	4	1	1 kHz	1	0.1
CASE (4)	0.05	120.06	0.95	3	6	6	1	1.5 kHz	1	0.1

case, the weighting (w_j) is equal to 1. As such, the problem is now a compliance problem in which minimization of deflection is the target. In this case, the optimization is done without polarization optimization. To do optimization without polarization, it is possible to simply put the penalization factor for the polarization equal to 0, i.e., $\text{penalPol} = 0$ in GENERAL DEFINITIONS part of the code. As can be seen in Fig. 5b, the obtained density layout for this case is similar to the results of the topology optimization of passive materials as reported by 99 lines (Sigmund 2001) or 88 lines (Andreassen et al. 2011) of MATLAB code. This was expected since the PZT materials have the plane isotropic behavior. The numerical results of the optimizations are given in Table 2. It is reported for the aforementioned case that the output electrical energy is zero which is due to the charge cancellation. In fact, lateral force induces tension and compression in different parts of the piezoelectric plate which produces voltages with opposite sign on the surface of the electrode. The opposite signs of voltages nullify each other.

For the next case, polarization is also optimized by putting $\text{penalPol} = 1$. In Fig. 5c and d, it is obvious that the density layout did not change and the polarization profile is not uniform anymore. By this polarization optimization, based on Table 2, not only the mechanical energy of the piezoelectric plate is reduced in comparison with the first case, but also the problem of charge cancellation is suppressed and we have a non-zero electrical energy.

In the next case, the goal is to increase the electrical energy due to the same amount of force. To do so, the weighting factor is decreased to 0.01. As can be seen in Fig. 5e and f, the density layout is changed and the polarization profile is changed accordingly as well. By observing the obtained numerical results in Table 2, the electrical energy is increased in comparison with the previous cases with the cost of increasing the mechanical energy as well. By further increasing the weighting factor, the results of Fig. 5g and h will be obtained in which a jump in both mechanical energy and electrical energy of the design can be seen in Table 2.

It can be noticed that in the optimized polarization profiles, there are areas with null polarity (green color) while there are materials (non-void). This null polarity is mostly at places where there is a transition between the polarization direction. It is possible to take these null polarity areas into consideration in the optimization problem formulation as discussed in Donoso and Sigmund (2016).

In the next example, the piezoelectric plate with the same configuration of boundary and load condition is considered while the force is considered to be harmonic. In Fig. 6, the results of topology optimization under harmonic force can be seen. The related numerical results are also reported in Table 2. In this figure, the excitation frequency is primarily considered to be 1 kHz. At this frequency, the optimization converges to the final layout for different values of w_j . However, by increasing the frequency of excitation, convergence problems begin. The problems are

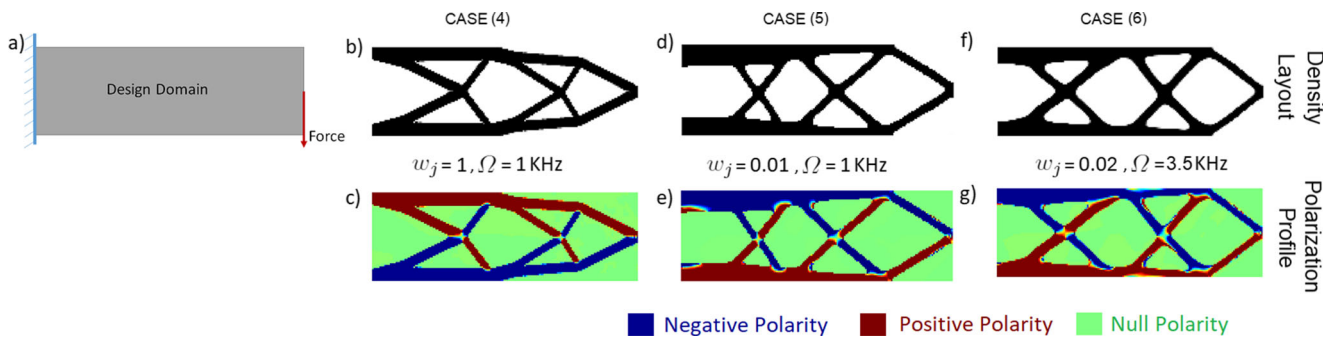


Fig. 6 Topology optimization of a piezoelectric energy harvester under application of a lateral harmonic force for different values of w_j . Panel **a** presents the specification for a practical use and panels **b**, **d**,

and **c**, **e**, **g** respectively present the density and polarization profile of the obtained design. For the polarization profile, blue, red and green represent respectively negative, positive and null polarization.

parasitic effects of the material layout and the force will be disjointed from the material. These problems are due to the fact that the excitation frequency is close to the resonance and antiresonance frequency of the piezoelectric plate. The resonances are the natural frequencies when the electrodes are short-circuited and antiresonances are the ones when the electrodes are in open-circuit condition (Lerch 1990). During the optimization, these frequencies are changing in each iteration. Therefore, it is possible that the excitation frequency comes close to the resonance frequency during the optimization which will introduce singularity in the FEM (23) and the amplitude of the displacement vector reaches to infinity. Even by defining the damping, still the jump in the displacement vector will result in numerical instabilities and disjoint problem in which the force will be disconnected from material as it is reported by Noh and Yoon (2012). One solution to reduce the numerical instabilities is to restrict the *move limit* of the MMA code. Indeed, the *move limit* of the MMA optimization code is by default set to 1 which means the density of material can jump from 0 (void) to 1 (material) in a single iteration. To modify this *move limit*, in the `mmasub.m`, one needs to change the value of `move` from 1 to 0.1. With this restriction on the *move limit*, convergence to a black and white final layout is achieved for 3.5 kHz excitation frequencies as it is shown in Fig. 6f and g. However, by reducing the MMA *move limit*, the final result can be trapped in the local optima. This is the maximum excitation frequency that the convergence can be achieved. After this frequency, again the force will be disjointed from the material and the convergence problem can be seen. To overcome the challenges of dynamic topology optimization of mechanical structures, several methods can be found in the literature. For example, Olhoff and Du (2005) suggest that the excitation frequency can be

increased gradually during the optimization iterations. Liu et al. (2015) and Jensen (2007) modeled the damping in the dynamic system and optimized the structure for an interval of frequencies including the resonance ones. For mechanical structures, the goal is to reduce the displacement or stored mechanical energy in the system. For piezoelectric energy harvesters in which the maximization of electrical output regarding the mechanical input is desired, Noh and Yoon (2012) solved the disjoint problem of force and material in dynamic topology optimization by defining a constraint on the mechanical energy of the piezoelectric structure and they defined the objective function as maximization of the electrical output of the system.

Reducing the weighting factor w_j brings the resonance frequency closer to the excitation frequency to some extent. However, a very low weighing factor can introduce convergence problems as well. On the other hand, for piezoelectric energy harvesters, matching the resonance frequency and excitation frequency is favorable. To do so, the problem formulation should be changed. For example, Kim and Shin (2013) did eigenfrequency optimization to increase the electromechanical coupling coefficient of the design and to match the resonance frequency and the excitation frequency. Wang et al. (2017) and Nakasone and Silva (2010) formulated the optimization problem to optimize the eigenmodes in addition to optimization of the eigenfrequency. It is worth mentioning that for these problem formulations, alternative material interpolation functions should be used to avoid the artificial local modes in the low-density regions (Pedersen 2000). The alternative material interpolation functions can be the one introduced by Huang et al. (2010) for the stiffness matrix or the RAMP interpolation function which is used by Nakasone and Silva (2010) in combination with the PEMAP-P interpolation function.

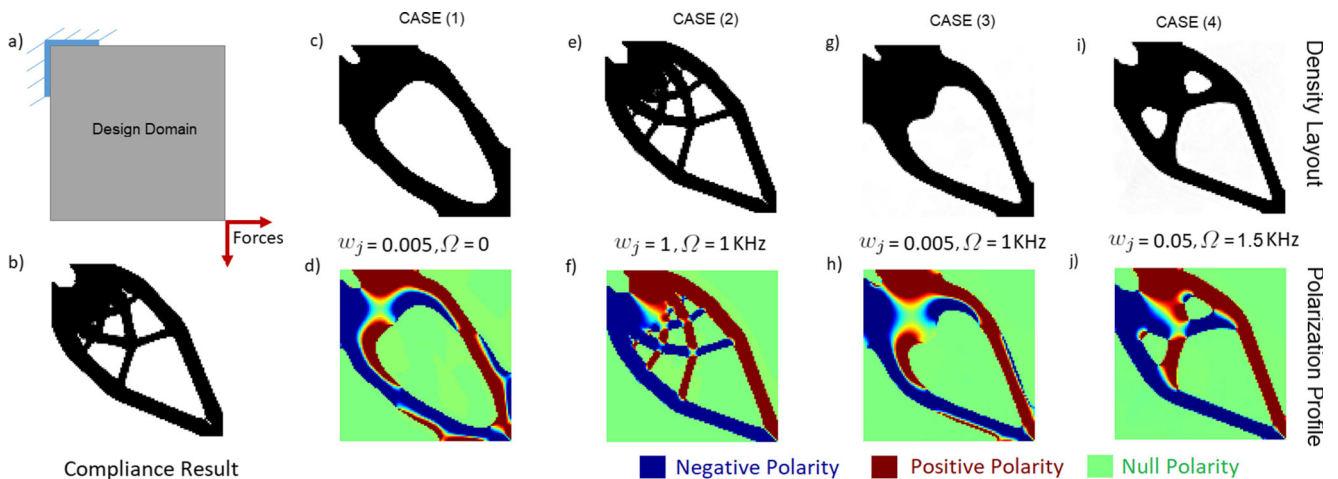


Fig. 7 Topology optimization of a piezoelectric energy harvester under application of 2 load cases for different values of w_j and Ω . Panel **a** presents the specification for a practical use and panel **b** is the result that can be obtained through classical compliance optimization. Panels

c, e, g, i and **d, f, h, j** respectively present the density and polarization profile of the obtained design. For the polarization profile, blue, red and green represent respectively negative, positive and null polarization

5.2.2 Two loads case

In this example, the goal is to optimize piezoelectric energy harvesters for two loads case as shown in Fig. 7a. In fact, the piezoelectric plate will be optimized for in-plane forces that can come from different directions as it is discussed in Homayouni-Amlashi et al. (2020b) since any in-plane force can be decomposed to the load cases shown in Fig. 7a. The mechanical boundary condition of this figure is proposed to make the harvested energy symmetric with respect to the forces in each direction as it is possible.

To implement the two loads case, the following changes should be made to the energy harvesting code mentioned in the Appendix. The parameters in the GENERAL DEFINITIONS part should be changed by putting $L_p = 3e-2$, $W_p = 3e-2$, $n_{elx} = 100$, and $n_{ely} = 100$. To define the boundary condition, the DEFINITION OF BOUNDARY CONDITION part will be changed completely as follows

```

%% DEFINITION OF BOUNDARY CONDITION
BCRatio=.2;
fixe = [1:2*(nely+1)];
fixe2 =2*[1:(nely+1):((nelx+1)*(nely+1)-(
    nely-1))];
BCT=ceil(BCRatio*length(fixe));
BCT2=ceil(BCRatio*length(fixe2));
fixeddofs=[fixe(1:BCT),fixe2(1:BCT2),fixe2
    (1:BCT2)-1];
freedofs = setdiff(1:ndof,fixeddofs);
lf = length(freedofs);

```

BCRatio is the ratio of the clamped part to the total length of the edge. To define the two load case the part FORCE DEFINITION will be changed with the following lines:

```

%% FORCE DEFINITION
nf = 2;
F = sparse(ndof,nf);
Fe=ndof;
F(Fe,1) = +1;
F(Fe-1,2) = +1;
Ftot = [F(freedofs,:);zeros(1,nf)];

```

In contrast to previous case, the initial values for the densities ($x_{\{0\}}$) are not equal to volume fraction (volfrac). It is observed that by changing the initial values to one, the obtained results are more symmetric. Therefore, the following line of energy harvesting code is updated,

```

101 x = repmat(1,nely,nelx); % Initial values
    for density ratios

```

The results for the defined load and boundary condition are illustrated in Fig. 7. Figure 7b shows the compliance result due to static force without polarization optimization. In Fig. 7c and d the excitation frequency is zero, w_j is decreased and polarization is also optimized. Other cases are optimized for different excitation frequencies and weighting factors. The numerical results are reported in Table 2. For defined geometry, load and boundary condition of this example, the highest frequency that the convergence to black and white is achieved is 1.5 kHz. This is due to the

fact that with the defined geometry and boundary condition, the first resonance frequency is lower than the previous example of Fig. 6.

6 Discussion

In the actuation code, the optimization algorithm is OC. However, it is very simple to apply the MMA to the actuation code. One has just to copy the MMA Preparation part from the energy harvesting code and paste it after the INITIALIZE ITERATION part of the actuation code. Then, one has to remove the OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES part and substitute the MMA OPTIMIZATION OF DESIGN VARIABLES part from the energy harvesting code. It will be observed that the obtained result remains the same.

It is possible to implement the OC in the energy harvesting code as well. This can be simply done by substituting the MMA OPTIMIZATION OF DESIGN VARIABLES part with the OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES part of the actuation code. However, due to coupling effect, sometimes the convergence problem appears.

It should be noted that the results presented in this paper are not the best results that can be obtained from the codes. In fact by changing the penalization factors, volume fraction, filter radius and initial values, different results can be obtained. However, finding the best results or performing the parameters analysis on the final results is not the subject of this educational paper.

There are similar aspects between the proposed codes here and the 88 lines of MATLAB code (Andreassen et al. 2011), such as the use of different filtering methods, different boundary conditions, and different load cases. These aspects are not discussed here since the procedures of implementation are similar.

Extension of the codes for different goals such as different objective functions, multi-material optimization, and 3D finite element modeling is straightforward. For example, the 169 lines of code for 3D topology optimization (Liu and Tovar 2014) of passive material are the extension of the 88 lines of code. The same strategy can be used for the proposed code here to extend the code for 3D finite elements to consider multi-layer piezoelectric plates and out of plane forces.

7 Conclusion

Two MATLAB codes are proposed for topology optimization of piezoelectric actuators and energy harvesters. The

codes are developed based on the finite element modeling of piezoelectric materials. The PEMAP-P as an extension of SIMP approaches is used for material interpolation scheme. Optimality criteria and method of moving asymptotes are used for optimization of element's density and polarization direction. Different parts of the codes are explained in detail to make the implementation and extension of the codes straightforward. Some basic and general examples are chosen to show the effectiveness of the codes. The aim of the codes is to help the students and newcomers in the field of topology optimization of smart materials in particular the piezoelectric material. While the codes have been and can be used for energy harvesting and actuation applications, perspective works include their application to optimization of piezoelectric sensors as well as piezoelectric sensors-actuators, also named as self-sensing (Bafumba Liseli and Agnus 2019; Rakotondrabe 2013; Aljanaideh et al. 2018).

Funding This work has been supported by the national CODE-TRACK project (ANR-17-CE05-0014-01, Control theory tools for optimal design of piezoelectric energy harvesters devoted to birds tracking devices). This work has also been partially supported by the Bourgogne Franche-Comté region project COMPACT.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results All the results presented in this work can be reproduced with the MATLAB code available within the paper and by adding reference of this latter.

Appendix: Proof of normalization

Here the goal is to prove the normalization which is proposed by authors in Homayouni-Amlashi (2019) and Homayouni-Amlashi et al. (2020b). To normalize the global FEM equation of the piezoelectric plate

$$\begin{bmatrix} K_{uu} - M\Omega^2 & K_{u\phi} \\ K_{\phi u} & -K_{\phi\phi} \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (\text{A.1})$$

the first step of normalization is to factorize the values defined in (20) and (22)

$$\begin{aligned} K_{uu} &= k_0 \tilde{K}_{uu}, & M &= m_0 \tilde{M}, & K_{u\phi} &= \alpha_0 \tilde{K}_{u\phi}, \\ K_{\phi\phi} &= \beta_0 \tilde{K}_{\phi\phi}, & U &= u_0 \tilde{U}, & \Phi &= \phi_0 \tilde{\Phi}, & F &= f_0 \tilde{F} \end{aligned} \quad (\text{A.2})$$

It should be noted that since the responses of the system, i.e., U and Φ are unknown, their factorization values will be defined later.

With the help of factorization defined in (A.2), (A.1) can be rewritten as follows,

$$\begin{bmatrix} k_0 \tilde{K}_{uu} - m_0 \tilde{M} \Omega^2 & \alpha_0 \tilde{K}_{u\phi} \\ \alpha_0 \tilde{K}_{\phi u} & -\beta_0 \tilde{K}_{\phi\phi} \end{bmatrix} \begin{bmatrix} u_0 \tilde{U} \\ \phi_0 \tilde{\Phi} \end{bmatrix} = \begin{bmatrix} f_0 \tilde{F} \\ 0 \end{bmatrix} \quad (\text{A.3})$$

This is two linearly coupled equations. The second equation can be written as

$$\alpha_0 u_0 \tilde{K}_{\phi u} \tilde{U} - \beta_0 \phi_0 \tilde{K}_{\phi\phi} \tilde{\Phi} = 0 \quad (\text{A.4})$$

Then, the $\tilde{\Phi}$ will be

$$\tilde{\Phi} = \frac{\alpha_0 u_0}{\beta_0 \phi_0} \tilde{K}_{\phi\phi}^{-1} \tilde{K}_{\phi u} \tilde{U} \quad (\text{A.5})$$

The first linear equation from (A.3) can be written as

$$(k_0 \tilde{K}_{uu} - m_0 \tilde{M} \Omega^2) u_0 \tilde{U} + \alpha_0 \phi_0 \tilde{K}_{u\phi} \tilde{\Phi} = f_0 \tilde{F} \quad (\text{A.6})$$

By substituting (A.5) to (A.6) and dividing the resulted equation by $k_0 u_0$, one will have

$$\begin{aligned} & \left(\tilde{K}_{uu} - \frac{m_0}{k_0} \tilde{M} \Omega^2 \right) \tilde{U} + \frac{\alpha_0 \phi_0}{u_0 k_0} \frac{\alpha_0 u_0}{\beta_0 \phi_0} \tilde{K}_{u\phi} \tilde{K}_{\phi\phi}^{-1} \tilde{K}_{\phi u} \tilde{U} \\ & = \frac{f_0}{k_0 u_0} \tilde{F} \end{aligned} \quad (\text{A.7})$$

Now, it is possible to define,

$$\tilde{\Omega}^2 = \frac{m_0}{k_0} \Omega^2, \quad u_0 = \frac{f_0}{k_0} \quad (\text{A.8})$$

By substituting (A.8) to (A.7)

$$(\tilde{K}_{uu} - \tilde{M} \tilde{\Omega}^2) \tilde{U} + \frac{\alpha_0^2}{\beta_0 k_0} \tilde{K}_{u\phi} \tilde{K}_{\phi\phi}^{-1} \tilde{K}_{\phi u} \tilde{U} = \tilde{F} \quad (\text{A.9})$$

Now, the normalization factor can be defined as

$$\frac{\beta_0 k_0}{\alpha_0^2} = \gamma \quad (\text{A.10})$$

Again, the (A.9) can be written as,

$$(\tilde{K}_{uu} - \tilde{M} \tilde{\Omega}^2) \tilde{U} + \frac{1}{\gamma} \tilde{K}_{u\phi} \tilde{K}_{\phi\phi}^{-1} \tilde{K}_{\phi u} \tilde{U} = \tilde{F} \quad (\text{A.11})$$

Now, it can be considered that

$$\frac{1}{\gamma} \tilde{K}_{\phi\phi}^{-1} \tilde{K}_{\phi u} \tilde{U} = \tilde{\Phi} \quad (\text{A.12})$$

If one write (A.12) as

$$\tilde{K}_{\phi u} \tilde{U} - \gamma \tilde{K}_{\phi\phi} \tilde{\Phi} = 0 \quad (\text{A.13})$$

then the following linearly coupled equation can be written with the help of (A.11) and (A.13)

$$\begin{bmatrix} \tilde{K}_{uu} - \tilde{M} \tilde{\Omega}^2 & \tilde{K}_{u\phi} \\ \tilde{K}_{\phi u} & -\gamma \tilde{K}_{\phi\phi} \end{bmatrix} \begin{bmatrix} \tilde{U} \\ \tilde{\Phi} \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ 0 \end{bmatrix} \quad (\text{A.14})$$

which is the same as (23) of the paper.

However, right now we have two equations for $\tilde{\Phi}$ in (A.12) and (A.5). By equating these two equations results in

$$\tilde{\Phi} = \frac{\alpha_0 u_0}{\beta_0 \phi_0} \tilde{K}_{\phi\phi}^{-1} \tilde{K}_{\phi u} \tilde{U} = \frac{1}{\gamma} \tilde{K}_{\phi\phi}^{-1} \tilde{K}_{\phi u} \tilde{U} \quad (\text{A.15})$$

We can then simplify both sides and replace the γ coefficient

$$\frac{\alpha_0 u_0}{\beta_0 \phi_0} = \frac{\alpha_0^2}{\beta_0 k_0} \quad (\text{A.16})$$

We will find the following equation for ϕ_0 ,

$$\phi_0 = \frac{u_0 k_0}{\alpha_0} \quad (\text{A.17})$$

But from (A.8) u_0 can be replaced and the final value for ϕ_0 will be

$$\phi_0 = \frac{f_0}{\alpha_0} \quad (\text{A.18})$$

Now, by finding the values of γ , u_0 and ϕ_0 , the normalization of global FEM equations of piezoelectric material is finished. Actuation is the sub problem for the aforementioned normalization. Only (16) should be normalized. This equation with the help of the normalization factors defined in (20) and (24) can be written as

$$k_0 \tilde{K}_{uu} u_0 \tilde{U} + \alpha_0 \tilde{K}_{u\phi} \phi_0 \tilde{\Phi} = f_0 \tilde{F} \quad (\text{A.19})$$

By dividing each side of equation by f_0 ,

$$\frac{k_0}{f_0} \tilde{K}_{uu} u_0 \tilde{U} + \frac{\alpha_0}{f_0} \tilde{K}_{u\phi} \phi_0 \tilde{\Phi} = \tilde{F} \quad (\text{A.20})$$

Now by considering u_0 and ϕ_0 equal to values in (A.8) and (A.18), the normalized (21) will be obtained.

$$\tilde{K}_{uu} \tilde{U} + \tilde{K}_{u\phi} \tilde{\Phi} = \tilde{F} \quad (\text{A.21})$$

MATLAB TOPOLOGY OPTIMIZATION CODE FOR PIEZOELECTRIC ACTUATORS

```

1 % A TOPOLOGY OPTIMIZATION MATLAB CODE FOR PIEZOELECTRIC ACTUATOR
2 function Piezo_Actuator
3 %% GENERAL DEFINITIONS
4 Lp = 1e-2; % Piezoelectric plate length (m) in x direction
5 Wp = 0.5e-2; % Piezoelectric plate width (m) in y direction
6 h = 1e-4; % Piezoelectric plate Thickness (m) in z direction
7 nelx = 150; % Number of element in x direction
8 nely = 75; % Number of element in y direction
9 penalKuu = 3;penalKup = 4;penalPol = 1; % Penalization factors
10 Ks = 0.005; %2e-2/3 End spring stiffness
11 volfrac = 0.3; % Volume fraction
12 rmin = 2.5; % Filter radius
13 ft = 1; % Filter type - 1 for sensitivity, 2 for density
14 Max_loop = 1000; % Maximum number of Iteration
15 %% MATERIAL PROPERTIES (PZT 4)
16 e31 = -14.9091; % e31 Coupling coefficient
17 C = zeros(3,3); % Creation of null mechanical stiffness tensor
18 C(1,1) = 9.1187e+10;C(2,2) = C(1,1);
19 C(1,2) = 3.0025e+10;C(2,1) = C(1,2);
20 C(3,3) = 3.0581e+10;
21 %% PREPARE FINITE ELEMENT ANALYSIS
22 le = Lp/nelx; % Element length
23 we = Wp/nely; % Element width
24 e = [e31,e31,0]; % Piezoelectric matrix
25 x1 = 0;y1 = 0;x2 = le;y2 = 0;x3 = le;y3 = we;x4 = 0;y4 = we; % Element node coordinate
26 GP = [-1/sqrt(3) -1/sqrt(3);1/sqrt(3) -1/sqrt(3);1/sqrt(3) 1/sqrt(3);-1/sqrt(3) 1/sqrt(3)]; %
    Gauss quadrature DMDOFs
27 kuu = 0;kup = 0; % Initial values for piezoelectric matrices
28 for i = 1:4
29     s = GP(i,1);t = GP(i,2); % Natural coordinates
30     n1 = (1/4)*(1-s)*(1-t);
31     n2 = (1/4)*(1+s)*(1-t);
32     n3 = (1/4)*(1+s)*(1+t);
33     n4 = (1/4)*(1-s)*(1+t);
34     a = (y1*(s-1)+y2*(-1-s)+y3*(1+s)+y4*(1-s))/4;
35     b = (y1*(t-1)+y2*(1-t)+y3*(1+t)+y4*(-1-t))/4;
36     c = (x1*(t-1)+x2*(1-t)+x3*(1+t)+x4*(-1-t))/4;
37     d = (x1*(s-1)+x2*(-1-s)+x3*(1+s)+x4*(1-s))/4;
38     B1 = [a*(t-1)/4-b*(s-1)/4 0 ; 0 c*(s-1)/4-d*(t-1)/4 ;c*(s-1)/4-d*(t-1)/4 a*(t-1)/4-b*(s-1)
    /4];
39     B2 = [a*(1-t)/4-b*(-1-s)/4 0 ; 0 c*(-1-s)/4-d*(1-t)/4;c*(-1-s)/4-d*(1-t)/4 a*(1-t)/4-b
    *(-1-s)/4];
40     B3 = [a*(t+1)/4-b*(s+1)/4 0 ; 0 c*(s+1)/4-d*(t+1)/4 ;c*(s+1)/4-d*(t+1)/4 a*(t+1)/4-b*(s+1)
    /4];
41     B4 = [a*(-1-t)/4-b*(1-s)/4 0 ; 0 c*(1-s)/4-d*(-1-t)/4 ;c*(1-s)/4-d*(-1-t)/4 a*(-1-t)/4-b
    *(1-s)/4];
42     Bfirst = [B1 B2 B3 B4];
43     Jfirst = [0 1-t t-s s-1 ; t-1 0 s+1 -s-t ;s-t -s-1 0 t+1 ; 1-s s+t -t-1 0];
44     J = [x1 x2 x3 x4]*Jfirst*[y1 ; y2 ; y3 ; y4]/8; % Determinant of jacobian matrix
45     Bu = Bfirst/J;
46     Bphi = 1/h;
47     kuu = kuu+h*J*transpose(Bu)*C*Bu; % Mechanical stiffness matrix:
48     kup = kup+ h*J*transpose(Bu)*e'*Bphi; % Piezoelectric coupling matrix:
49 end
50 k0 = max(abs(kuu(:)));alpha = max(kup(:));% Normalization Factors
51 kuu = kuu/k0;kup = kup/alpha; % Normalization
52 ndof = 2*(nely+1)*(nelx+1); % Mechanical degrees of freedom
53 nele = nelx*nely; % Number of elements
54 nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
55 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nele,1);
56 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nele,1);
57 edofMatPZT = 1:nele;
58 iK = kron(edofMat,ones(8,1))';
59 jK = kron(edofMat,ones(1,8))';
60 iKup = edofMat';
61 jKup = kron(edofMatPZT,ones(1,8))';
62 %% OUTPUT DISPLACEMENT DEFINITION
63 DMDOF = ndof-1; % Desired mechanical degree of freedom
64 L = sparse(2*(nely+1)*(nelx+1),1);
65 L(DMDOF,1) = -1;

```

```

66 Uu = zeros(ndof,1); % Creation of null displacement vector
67 Adjoint = zeros(ndof,1); % Creation of null adjoint vector
68 Up(1:nele,1)= 1;% Actuation voltage
69 %% DEFINITION OF BOUNDARY CONDITION
70 fixeddofs1 = 1:2*(nely+1); % Main supports
71 fixeddofs2 = 2*[(nely+1):(nely+1):(nely+1)*(nelx+1)]; % Applying symmetry
72 fixeddofs = [fixeddofs1,fixeddofs2]; % Fusion of every supports
73 freedofs = setdiff(1:ndof,fixeddofs); % Computation of freedofs
74 lf = length(freedofs); % Number of free dofs
75 %% PREPARE FILTER
76 iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
77 jH = ones(size(iH));
78 sH = zeros(size(iH));
79 k = 0;
80 for i1 = 1:nelx
81     for j1 = 1:nely
82         e1 = (i1-1)*nely+j1;
83         for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
84             for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
85                 e2 = (i2-1)*nely+j2;
86                 k = k+1;
87                 iH(k) = e1;
88                 jH(k) = e2;
89                 sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
90             end
91         end
92     end
93 end
94 H = sparse(iH,jH,sH);
95 Hs = sum(H,2);
96 %% INITIALIZE ITERATION
97 x = repmat(volfrac,nely,nelx); % Initial values for density ratios
98 pol = repmat(0.1,[nely,nelx]); % Initial values for polarization
99 xPhys = x;
100 loop = 0;
101 Density_change = 1;
102 E0 = 1; Emin = 1e-9;
103 e0 = 1; eMin = 1e-9;
104 %% START ITERATION
105 while Density_change > 0.01 && loop < Max_loop
106     tic
107     loop = loop + 1;
108     % FE-ANALYSIS
109     sKuu = kuu(:)*(Emin+xPhys(:)'.^penalKuu*(E0-Emin));
110     sKup = kup(:)*(eMin+xPhys(:)'.^penalKup*(e0-eMin).*((2*pol(:)-1)'.^penalPol));
111     Kuu = sparse(iK,jK,sKuu); % Global stiffness matrix
112     Kup = sparse(iKup(:),jKup(:),sKup(:)); % Global piezoelectric coupling matrix
113     Kuu(DMDOF,DMDOF) = Kuu(DMDOF,DMDOF)+Ks; % Assembling the stiffness of the modeled spring
114     Uu(freedofs,:) = Kuu(freedofs,freedofs)\(-Kup(freedofs,)*Up); % Mechanical displacement
115     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
116     CE = -(sum(L(edofMat).*Uu(edofMat),2));
117     c = full(sum(CE)); % Objective Function
118     Adjoint(freedofs,:) = Kuu(freedofs,freedofs)\L(freedofs,:); % Adjoint vector
119     DCKuuE = sum((Adjoint(edofMat)*kuu).*Uu(edofMat),2);
120     DCKupE = (Adjoint(edofMat)*kup).*Up(edofMatPZT);
121     DCKuu = reshape(DCKuuE,[nely,nelx]);
122     DCKup = reshape(DCKupE,[nely,nelx]);
123     dc = penalKuu*(E0-Emin)*xPhys.^(penalKuu-1).*DCKuu+...
124         penalKup*(E0-Emin).*((2*pol-1).^(penalPol)).*xPhys.^(penalKup-1).*DCKup; % Sensitivity with
        respect to x
125     dp = 2*penalPol*((2*pol-1).^(penalPol-1)).*xPhys.^(penalKup).*DCKup; % Sensitivity with
        respect to p
126     dv = ones(nely,nelx); % Volume sensitivity
127     % FILTERING/MODIFICATION OF SENSITIVITIES
128     if ft == 1
129         dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
130     elseif ft == 2
131         dc(:) = H*(dc(:))./Hs; dv(:) = H*(dv(:))./Hs;
132     end
133     %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES
134     l1 = 0; l2 = 1e9; move = 0.2;

```

```

135 while (l2-l1)/(l1+l2) > 1e-3
136     lmid = 0.5*(l2+l1);
137     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*(max(1e-30,-dc./dv/lmid)).^0.3)))); % OC
update of density
138     if ft == 1
139         xPhys = xnew;
140     elseif ft == 2
141         xPhys(:) = (H*xnew(:))./Hs;
142     end
143     if sum(xPhys(:)) > volfrac*nele, l1 = lmid; else l2 = lmid; end
144 end
145 pol = max(0,max(pol-move,min(1.,min(pol+move,sign(-dp))))); % OC update of polarization
146 Density_change = max(abs(xnew(:)-x(:)));
147 x = xnew;
148 %% PLOT DENSITIES & POLARIZATION
149 figure(1);colormap(gray); imagesc(1-[xPhys;flip(xPhys)]); caxis([0 1]); axis equal; axis off;
drawnow;
150 figure(2);colormap(jet); imagesc([(xPhys.*(pol*2-1))+1]/2;((flip(xPhys).*flip(pol*2-1))+1)
/2]); caxis([0 1]); axis equal; axis off; drawnow;
151 fprintf(' It:%2.0i Time:%3.2fs Obj:%3.3f Vol:%3.3f ch:%3.3f\n ',loop,toc,c,mean(xPhys
(:)),Density_change);
152 end
153 % ||=====||
154 % || THIS CODE IS WRITTEN BY ABBAS HOMAYOUNI-AMLAHI, THOMAS SCHLINQUER, ||
155 % || ABDENBI MOHAND-OUSAIID AND MICKY RAKOTONDRAHE ||
156 % ||=====||

```

MATLAB TOPOLOGY OPTIMIZATION CODE FOR PIEZOELECTRIC ENERGY HARVESTERS

```

1 % A 2D TOPOLOGY OPTIMIZATION CODE FOR PIEZOELECTRIC ENERGY HARVESTER
2 function Piezo_EnergyHarvester
3 %% GENERAL DEFINITIONS
4 Lp = 3e-2; % Piezoelectric plate length (m) in x direction
5 Wp = 1e-2; % Piezoelectric plate width (m) in y direction
6 h = 1e-4; % Piezoelectric plate Thickness (m) in z direction
7 nelx = 180; % Number of element in x direction
8 nely = 60; % Number of element in y direction
9 penalKuu = 3; penalKup = 6;penalKpp = 4;penalPol = 1; % Penalization factors
10 omega = 0; % Excitation frequency (Hz)
11 wj = 1; % Objective function weighing factor
12 volfrac = 0.4; % Volume fraction
13 rmin = 3; % Filter radius
14 ft = 2; % Filter type - 1 for sensitivity, 2 for density
15 Max_loop = 400; % Maximum number of Iteration
16 %% MATERIAL PROPERTIES (PZT 4)
17 ro = 7500; % Density of piezoelectric material
18 e31 = -14.9091; % e31 Coupling coefficient
19 ep33 = 7.8374e-09; % Piezoelectric permittivity epsilon33
20 C = zeros(3,3); % Creation of null mechanical stiffness tensor
21 C(1,1) = 9.1187e+10; C(2,2) = C(1,1);
22 C(1,2) = 3.0025e+10; C(2,1) = C(1,2);
23 C(3,3) = 3.0581e+10;
24 %% PREPARE FINITE ELEMENT ANALYSIS
25 le = Lp/nelx; % Element length
26 we = Wp/nely; % Element width
27 e = [e31,e31,0]; % Piezoelectric matrix
28 x1 = 0;y1 = 0;x2 = le;y2 = 0;x3 = le;y3 = we;x4 = 0;y4 = we; % Element node coordinate
29 GP = [-1/sqrt(3) -1/sqrt(3);1/sqrt(3) -1/sqrt(3);1/sqrt(3) 1/sqrt(3);-1/sqrt(3) 1/sqrt(3)]; %
Gauss quadrature points
30 kuu = 0;kpp = 0;kup = 0;m = 0; % Initial values for piezoelectric matrices
31 for i = 1:4
32     s = GP(i,1);t = GP(i,2); % Natural coordinates
33     n1 = (1/4)*(1-s)*(1-t);
34     n2 = (1/4)*(1+s)*(1-t);
35     n3 = (1/4)*(1+s)*(1+t);
36     n4 = (1/4)*(1-s)*(1+t);
37     a = (y1*(s-1)+y2*(-1-s)+y3*(1+s)+y4*(1-s))/4;
38     b = (y1*(t-1)+y2*(1-t)+y3*(1+t)+y4*(-1-t))/4;
39     c = (x1*(t-1)+x2*(1-t)+x3*(1+t)+x4*(-1-t))/4;
40     d = (x1*(s-1)+x2*(-1-s)+x3*(1+s)+x4*(1-s))/4;
41     B1 = [a*(t-1)/4-b*(s-1)/4 0 ; 0 c*(s-1)/4-d*(t-1)/4 ;c*(s-1)/4-d*(t-1)/4 a*(t-1)/4-b*(s-1)
/4];

```

```

42 B2 = [a*(1-t)/4-b*(-1-s)/4 0 ; 0 c*(-1-s)/4-d*(1-t)/4;c*(-1-s)/4-d*(1-t)/4 a*(1-t)/4-b
      *(-1-s)/4];
43 B3 = [a*(t+1)/4-b*(s+1)/4 0 ; 0 c*(s+1)/4-d*(t+1)/4 ;c*(s+1)/4-d*(t+1)/4 a*(t+1)/4-b*(s+1)
      /4];
44 B4 = [a*(-1-t)/4-b*(1-s)/4 0 ; 0 c*(1-s)/4-d*(-1-t)/4 ;c*(1-s)/4-d*(-1-t)/4 a*(-1-t)/4-b
      *(1-s)/4];
45 Bfirst = [B1 B2 B3 B4];
46 Jfirst = [0 1-t t-s s-1 ; t-1 0 s+1 -s-t ;s-t -s-1 0 t+1 ; 1-s s+t -t-1 0];
47 J = [x1 x2 x3 x4]*Jfirst*[y1 ; y2 ; y3 ; y4]/8; % Determinant of jacobian matrix
48 Bu = Bfirst/J;
49 Bphi = 1/h;
50 kuu = kuu + h*J*transpose(Bu)*C*Bu; % Mechanical stiffness matrix
51 kup = kup + h*J*transpose(Bu)*e'*Bphi; % Piezoelectric coupling matrix
52 kpp = kpp + h*J*transpose(Bphi)*ep33*Bphi; % Dielectric stiffness matrix
53 N = [n1,0,n2,0,n3,0,n4,0;0,n1,0,n2,0,n3,0,n4]; % Matrix of interpolation functions
54 m = m+J*ro*h*(N')*N; % Mass matrix
55 end
56 k0 = max(abs(kuu(:)));beta = max(kpp(:));alpha = max(kup(:));M0 = max(m(:)); % Normalization
      Factors
57 kuu = kuu/k0;kup = kup/alpha;kpp = kpp/beta;gamma = (k0*beta)/(alpha^2);m = m/M0; omega = M0*(
      omega*2*pi)^2/k0; % Normalization
58 ndof = 2*(nely+1)*(nelx+1); % mechanical degrees of freedom
59 nele = nelx*nely; % number of elements
60 nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
61 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nele,1);
62 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nele,1);
63 edofMatPZT = 1:nele;
64 iK = kron(edofMat,ones(8,1))';
65 jK = kron(edofMat,ones(1,8))';
66 iKup = edofMat';
67 jKup = kron(edofMatPZT,ones(1,8))';
68 B = ones(nele,1); % Boolean Matrix defined as a vector of ones
69 %% DEFINITION OF BOUNDARY CONDITION
70 fixeddofs = 1:2*(nely+1); % Clamped-Free
71 freedofs = setdiff(1:ndof, fixeddofs);
72 lf = length(freedofs);
73 %% FORCE DEFINITION
74 nf = 1; % Number of forces
75 F = sparse(ndof,nf);
76 Fe = ndof-(nely); % Definition of desired Dof for application of force
77 F(Fe,1) = +1; % Amplitude of the force
78 Ftot = [F(freedofs,:);zeros(1,nf)];
79 %% PREPARE FILTER
80 iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
81 jH = ones(size(iH));
82 sH = zeros(size(iH));
83 k = 0;
84 for il = 1:nelx
85     for j1 = 1:nely
86         e1 = (il-1)*nely+j1;
87         for i2 = max(il-(ceil(rmin)-1),1):min(il+(ceil(rmin)-1),nelx)
88             for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
89                 e2 = (i2-1)*nely+j2;
90                 k = k+1;
91                 iH(k) = e1;
92                 jH(k) = e2;
93                 sH(k) = max(0,rmin-sqrt((il-i2)^2+(j1-j2)^2));
94             end
95         end
96     end
97 end
98 H = sparse(iH,jH,sH);
99 Hs = sum(H,2);
100 %% INITIALIZE ITERATION
101 x = repmat(volfrac,nely,nelx); % Initial values for density ratios
102 pol = repmat(0.1,[nely,nelx]); % Initial values for polarization
103 xPhys = x;
104 loop = 0;
105 Density_change = 1;
106 E0 = 1; Emin = 1e-9;
107 e0 = 1; eMin = 1e-9;

```

```

108 eps0 = 1; epsMin = 1e-9;
109 %% MMA Preparation
110 mc = 1; % Number of constraints
111 nVar = 2*nele; % Number of variables
112 xmin = (1e-9)*ones(nele,1); % Minimum possible density
113 polmin = 0*ones(nele,1); % Minimum possible polarization
114 xmin = [xmin;polmin]; % Vector of minimum optimization variables
115 xmax = ones(nVar,1); % Vector of maximum optimization variables
116 xold1 = [x(:);pol(:)]; % Vector of variables for previous iteration
117 xold2 = [x(:);pol(:)]; % Vector of variables for 2nd previous iteration
118 low = xmin; % Initial vector of lower asymptotes
119 upp = xmax; % Initial vector of upper asymptotes
120 a0 = 1;
121 ai = zeros(mc,1);
122 ci = (1e5)*ones(mc,1);
123 di = zeros(mc,1);
124 %% START ITERATION
125 while Density_change > 0.005 && loop < Max_loop
126     tic
127     loop = loop + 1;
128     % FE-ANALYSIS
129     sM = m(:)*xPhys(:)';
130     sKuu = kuu(:)*(Emin+xPhys(:)'.^penalKuu*(E0-Emin));
131     sKup = kup(:)*(eMin+xPhys(:)'.^penalKup*(e0-eMin).*((2*pol(:)-1)'.^penalPol));
132     sKpp = kpp(:)*(epsMin+xPhys(:)'.^penalKpp*(eps0-epsMin));
133     % Creation of matrices
134     M = sparse(iK(:),jK(:),sM(:)); % Global mass matrix
135     Kuu = sparse(iK,jK,sKuu)-omega*M;
136     Kup = sparse(iKup(:),jKup(:),sKup(:)); % Global piezoelectric coupling matrix
137     Kpp = sparse(edofMatPZT(:),edofMatPZT(:),sKpp(:)); % Global dielectric stiffness matrix
138     KupEqui = Kup(freedofs,:)*B; KppEqui = B'*Kpp*B; % Equipotential Condition
139     Ktot = [Kuu(freedofs,freedofs),KupEqui;KupEqui',-gamma*KppEqui]; % Creation of total matrix
140     % with equipotential hypothesis
141     Ktot = 1/2*(Ktot + Ktot'); % Numerical symmetry enforcement
142     U = Ktot\Ftot; % Response vector
143     Uu(freedofs,:) = U(1:lf,:); Up = U(lf+1:end,:); % Separation of mechanical displacement and
144     % electrical Potential
145     ADJ1 = Ktot\[-Kuu(freedofs,freedofs)*Uu(freedofs,:);zeros(1,nf)]; % First adjoint vector
146     lambda1(freedofs,:) = ADJ1(1:lf,:); mu1 = B*ADJ1(lf+1:end,:);
147     ADJ2 = Ktot\[zeros(lf,nf);-KppEqui*Up]; % Second adjoint vector
148     lambda2(freedofs,:) = ADJ2(1:lf,:); mu2 = B*ADJ2(lf+1:end,:);
149     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
150     c = 0; Wm = 0; We = 0;
151     dc = zeros(nely,nelx);
152     dp = zeros(nely,nelx);
153     for i = 1:nf % nf is the total number of forces
154         Uu_i = Uu(:,i);Up_i = B*Up(:,i);
155         lambda1_i = lambda1(:,i); lambda2_i = lambda2(:,i);
156         mu1_i = mu1(:,i);mu2_i = mu2(:,i);
157         Wm = Wm+ reshape(sum((Uu_i(edofMat)*kuu).*Uu_i(edofMat),2),nely,nelx);
158         We = We+ reshape(sum((Up_i*kpp).*Up_i,2),nely,nelx);
159         dcKuuE = wj*(((1/2)*Uu_i(edofMat) + lambda1_i(edofMat))*kuu).*Uu_i(edofMat))- (1-wj)*((
160         lambda2_i(edofMat)*kuu).*Uu_i(edofMat));
161         dcKupE = wj*((lambda1_i(edofMat)*kup).*Up_i + ((Uu_i(edofMat))*kup).*mu1_i)- (1-wj)*((
162         lambda2_i(edofMat)*kup).*Up_i + ((Uu_i(edofMat))*kup).*mu2_i);
163         dcKppE = wj*((-mu1_i*kpp).*Up_i)- (1-wj)*((1/2)*(Up_i*kpp).*Up_i - (mu2_i*kpp).*Up_i);
164         dcME = wj*(((1/2)*Uu_i(edofMat) + lambda1_i(edofMat))*(-m*omega)).*Uu_i(edofMat))- (1-wj)
165         *((lambda2_i(edofMat))*(-m*omega)).*Uu_i(edofMat));
166         dcKuu = reshape(sum(dcKuuE,2),[nely,nelx]);
167         dcKup = reshape(sum(dcKupE,2),[nely,nelx]);
168         dcKpp = gamma*reshape(sum(dcKppE,2),[nely,nelx]);
169         dcM = reshape(sum(dcME,2),[nely,nelx]);
170         dc = dc + penalKuu*(E0-Emin)*xPhys.^penalKuu-1).*dcKuu+penalKup*(e0-eMin)*xPhys.^(
171         penalKup-1).*dcKup.*((2*pol-1).^penalPol))+penalKpp*(eps0-epsMin)*xPhys.^penalKpp-1).*dcKpp
172         +dcM;
173         dp = dp + (e0-eMin)*2*penalPol*((2*pol-1).^penalPol-1)).*xPhys.^penalKup.*dcKup; %
174         % Polarization sensitivity
175     end
176     Wm = sum(sum((Emin+xPhys.^penalKuu*(E0-Emin)).*Wm)); % Mechanical energy
177     We = sum(sum((epsMin+xPhys.^penalKpp*(eps0-epsMin)).*We)); % Electrical energy
178     c = wj*Wm- (1-wj)*We; % Objective function

```

```

171     dv = ones(nely,nelx);
172     % FILTERING/MODIFICATION OF SENSITIVITIES
173     if ft == 1
174         dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
175     elseif ft == 2
176         dc(:) = H*(dc(:))./Hs;
177         dv(:) = H*(dv(:))./Hs;
178     end
179     %% MMA OPTIMIZATION OF DESIGN VARIABLES
180     dp = dp/max(abs(dp(:))); % Normalizing the polarization sensitivity
181     xval = [x(:);pol(:)]; % Vector of current optimization variables
182     f0val = c; % Current objective function value
183     df0dx = [dc(:);dp(:)]; % Vector of Sensitivities
184     fval = [sum(xPhys(:))/(volfrac*nele) - 1]; % Constraint value
185     dfdx = [dv(:)' / (volfrac*nele),0*pol(:)']; % Constraint's Sensitivities
186     [xmma,~,~,~,~,~,~,~,low,upp] = mmasub(mc, nVar, loop, xval, xmin, xmax, xold1,
xold2, ...
187         f0val,df0dx,fval,dfdx,low,upp,a0,ai,ci,di); % MMA optimization
188     xnew = reshape(xmma(1:nele,1),nely,nelx); % Vector of updated density variables
189     if ft == 1
190         xPhys = xnew;
191     elseif ft == 2
192         xPhys(:) = (H*xnew(:))./Hs;
193     end
194     Density_change = max(abs(xnew(:)-x(:)));
195     xold2 = xold1(:);
196     xold1 = [x(:);pol(:)];
197     pol = reshape(xmma(nele+1:2*nele,1),nely,nelx); % Vector of updated polarization variables
198     x = xnew;
199     %% PLOT DENSITIES & POLARIZATION
200     figure(1);colormap(gray); imagesc(1-x); caxis([0 1]); axis equal; axis off; drawnow;
201     figure(2);colormap(jet); imagesc((x.*(pol*2-1))+1)/2); caxis([0 1]); axis equal; axis off;
drawnow;
202     fprintf(' It:%2.0i Time:%3.2fs Obj:%3.4f Wm.:%3.4f We.:%3.4f Vol:%3.3f ch:%3.3f\n ', loop,toc,c
,Wm,We,mean(xPhys(:)),Density_change);
203 end
204 % ||=====||
205 % || THIS CODE IS WRITTEN BY ABBAS HOMAYOUNI-AMLASHI, THOMAS SCHLINQUER, ||
206 % || ABDENBI MOHAND-OUSSAID AND MICKY RAKOTONDRABE ||
207 % ||=====||

```

References

- Alexandersen J, Andreassen CS (2020) A review of topology optimisation for fluid-based problems. *Fluids* 5(1):29
- Aljanaideh O et al (2018) Observer and robust h-inf control of a 2-dof piezoelectric actuator equipped with self-measurement. *IEEE Robot Autom Lett* 3(2):1080–1087
- Andreassen CS, Elingaard MO, Aage N (2020) Level set topology and shape optimization by density methods using cut elements with length scale control. *Struct Multidiscip Optim* 1–23
- Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in matlab using 88 lines of code. *Struct Multidiscip Optim* 43(1):1–16
- Bafumba Liseli J, Agnus J (2019) Lutz p. Optimal design of piezoelectric cantilevered actuators for charge-based self-sensing application. *MDPI Sensors*, Rakotondrabe M. <https://doi.org/10.3390/s19112582>
- Bendsøe MP (1989) Optimal shape design as a material distribution problem. *Struct Optim* 1(4):193–202
- Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method, *Comput Methods Appl Mech Eng*
- Bendsøe MP, Sigmund O (1995) Optimization of structural topology, shape, and material, vol 414. Springer, Berlin
- Bendsøe MP (2013) Topology optimization theory, methods and applications. Springer Science and Business Media, Berlin
- Challis VJ (2010) A discrete level-set topology optimization code written in matlab. *Struct Multidiscip Optim* 41(3):453–464
- Chen Q, Zhang X, Zhu B (2019) A 213-line topology optimization code for geometrically nonlinear structures. *Struct Multidiscip Optim* 59(5):1863–1879
- Chen S, Gonella S, Chen W, Liu WK (2010) A level set approach for optimal design of smart energy harvesters. *Comput Methods Appl Mech Eng* 199(37-40):2532–2543
- Clausen A, Aage N, Sigmund O (2015) Topology optimization of coated structures and material interface problems. *Comput Methods Appl Mech Eng* 290:524–541
- Cook RD et al (2007) Concepts and applications of finite element analysis. Wiley, New York
- de Almeida BV (2019) Topology optimization of bimorph piezoelectric energy harvesters considering variable electrode location. *Smart Mater Struct*, Pavanello, R
- Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Struct Multidiscip Optim* 49(1):1–38
- Donoso A, Sigmund O (2016) Topology optimization of piezo modal transducers with null-polarity phases. *Struct Multidiscip Optim* 53(2):193–203
- Gersborg-Hansen A, Bendsøe MP, Sigmund O (2006) Topology optimization of heat conduction problems using the finite volume method. *Struct Multidiscip Optim* 31(4):251–259

- Gonçalves JF, De Leon DM, Perondi EA (2018) Simultaneous optimization of piezoelectric actuator topology and polarization. *Struct Multidiscip Optim* 58(3):1139–1154
- Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. *J Appl Mech* 81(8)
- Hassani B, Hinton E (1998) A review of homogenization and topology optimization iii—topology optimization using optimality criteria. *Comput Struct* 69(6):739–756
- Homayouni-Amlashi A (2019) Multi directional piezoelectric plate energy harvesters designed by topology optimization algorithm. *IEEE Robot Autom Lett, Rakotondrabe M*
- Homayouni-Amlashi A, Mohand-Ousaid A, Rakotondrabe M (2020a) Analytical modelling and optimization of a piezoelectric cantilever energy harvester with in-span attachment. *Micromachines* 11(6):591
- Homayouni-Amlashi A, Mohand-Ousaid A, Rakotondrabe M (2020b) Topology optimization of 2dof piezoelectric plate energy harvester under external in-plane force. *J Micro-Bio Robot* 1–13
- Huang X, Zuo Z, Xie Y (2010) Evolutionary topological optimization of vibrating continuum structures for natural frequencies. *Comput Struct* 88(5-6):357–364
- Hutton DV, Wu J (2004) *Fundamentals of finite element analysis, vol 1*. McGraw-hill, New York
- Jensen JS (2007) Topology optimization of dynamics problems with padé approximants. *Int J Numer Methods Eng* 72(13):1605–1630
- Junior CDM, Erturk A, Inman DJ (2009) An electromechanical finite element model for piezoelectric energy harvester plates. *J Sound Vib* 327(1-2):9–25
- Kattan PI (2010) *MATLAB guide to finite elements: an interactive approach*. Springer Science and Business Media, Berlin
- Khadraoui S, Rakotondrabe M, Lutz P (2014) Optimal design of piezoelectric cantilevered actuators with guaranteed performances by using interval techniques. *IEEE ASME Trans Mechatron* 19(5):1660–1668
- Kim C, Shin JW (2013) Topology optimization of piezoelectric materials and application to the cantilever beams for vibration energy harvesting. *Int J Precis Eng Manuf* 14(11):1925–1931
- Kim JE, Kim DS, Ma PS, Kim YY (2010) Multi-physics interpolation for the topology optimization of piezoelectric systems. *Comput Methods Appl Mech Eng* 199(49-52):3153–3168
- Kögl M, Silva EC (2005) Topology optimization of smart structures: design of piezoelectric plate and shell actuators. *Smart Mater Struct* 14(2):387
- Lerch R (1990) Simulation of piezoelectric devices by two-and three-dimensional finite elements. *IEEE Trans Ultrason Ferroelectr Freq Control* 37(3):233–247
- Liu H, Zhang W, Gao T (2015) A comparative study of dynamic analysis methods for structural topology optimization under harmonic force excitations. *Struct Multidiscip Optim* 51(6):1321–1333
- Liu K, Tovar A (2014) An efficient 3d topology optimization code written in matlab. *Struct Multidiscip Optim* 50(6):1175–1196
- Logan DL (2000) *A first course in the finite element method using Algor*. Brooks/Cole Publishing Co, Pacific Grove
- Maute K, Sigmund O (2013) Topology optimization approaches: a comparative review. *Struct Multidiscip Optim* 6
- Menuzzi O, Fonseca JS, Perondi EA, Gonçalves JF, Padoin E, Silveira OA (2018) Piezoelectric sensor location by the observability gramian maximization using topology optimization. *Comput Appl Math* 37(1):237–252
- Moretti M, Silva EC (2019) Topology optimization of piezoelectric bi-material actuators with velocity feedback control. *Front Mech Eng* 14(2):190–200
- Muthalif AG, Nordin ND (2015) Optimal piezoelectric beam shape for single and broadband vibration energy harvesting: Modeling, simulation and experimental results. *Mech Syst Signal Process* 54:417–426
- Nakasone P, Silva E (2010) Dynamic design of piezoelectric laminated sensors and actuators using topology optimization. *J Intell Mater Syst Struct* 21(16):1627–1652
- Noh JY, Yoon GH (2012) Topology optimization of piezoelectric energy harvesting devices considering static and harmonic dynamic loads. *Adv Eng Softw* 53:45–60
- Olhoff N, Du J (2005) Topological design of continuum structures subjected to forced vibration. In: *Proceedings of 6th world congresses of structural and multidisciplinary optimization, Rio de Janeiro, Brazil*
- Pedersen NL (2000) Maximization of eigenvalues using topology optimization. *Structural Multidiscip Optim* 20(1):2–11
- Piefort V (2001) *Finite element modelling of piezoelectric active structures*. PhD thesis, Ph. D. thesis. Bruxelles, Belgium: Université Libre de Bruxelles, Department ...
- Rabenorosa K et al (2015) Performances analysis of piezoelectric cantilever based energy harvester devoted to mesoscale intra-body robot
- Rakotondrabe M (2011) Performances inclusion for stable interval systems. In: *American control conference*, pp 4367–4372
- Rakotondrabe M (2013) Combining self-sensing with an unknown-input-observer to estimate the displacement, the force and the state in piezoelectric cantilevered actuator. In: *American control conference*, pp 4523–4530
- Rakotondrabe M, Khadraoui S (2013) Design of piezoelectric actuators with guaranteed performances using the performances inclusion theorem and interval tools. In: *Smart materials-based actuators at the micro/nano-scale*. Springer
- Ruiz D, Díaz-Molina A, Sigmund O, Donoso A, Bellido J, Sánchez-Rojas J (2017) Optimal design of a microgripper-type actuator based on aln/si heterogeneous bimorph. In: *Smart sensors, actuators, and MEMS VIII, International society for optics and photonics*, vol 10246, p 1024606
- Rupp CJ, Evgrafov A, Maute K, Dunn ML (2009) Design of piezoelectric energy harvesting systems: a topology optimization approach based on multilayer plates and shells. *J Intell Mater Syst Struct* 20(16):1923–1939
- Salas R, Ramírez F, Montealegre-Rubio W, Silva E, Reddy J (2018) A topology optimization formulation for transient design of multi-entry laminated piezocomposite energy harvesting devices coupled with electrical circuit. *Int J Numer Methods Eng* 113(8):1370–1410
- Schlinquer T, Mohand-Ousaid A, Rakotondrabe M (2017) Optimal design of a unimorph piezoelectric cantilever devoted to energy harvesting to supply animal tracking devices. *IFAC-PapersOnLine* 50(1):14600–14605
- Schmit LA (1960) *Structural design by systematic synthesis*. In: *Proceedings of the second national conference on electronic computation*. ASCE, Sept. 1960
- Sigmund O (2001) A 99 line topology optimization code written in matlab. *Struct Multidiscip Optim* 21(2):120–127
- Sigmund O (2007) Morphology-based black and white filters for topology optimization. *Struct Multidiscip Optim* 33(4-5):401–424
- Sigmund O, Maute K (2013) Topology optimization approaches. *Struct Multidiscip Optim* 48(6):1031–1055
- Sigmund O, Torquato S (1999) Design of smart composite materials using topology optimization. *Smart Mater Struct* 8(3):365
- Sigmund O, Torquato S, Aksay IA (1998) On the design of 1–3 piezocomposites using topology optimization. *J Mater Res* 13(4):1038–1048

- Silva EN, Fonseca JO, Kikuchi N (1997) Optimal design of piezoelectric microstructures. *Comput Mech* 19(5):397–410
- Smith H, Norato JA (2020) A matlab code for topology optimization using the geometry projection method. *Struct Multidiscip Optim* 1–16
- Stolpe M, Svanberg K (2001) An alternative interpolation scheme for minimum compliance topology optimization. *Struct Multidiscip Optim* 22(2):116–124
- Suzuki K, Kikuchi N (1991) A homogenization method for shape and topology optimization. *Comput Methods Appl Mech Eng*
- Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. *Int J Numer Methods Eng* 24(2):359–373
- Svanberg K (2007) Mma and gcmma-two methods for nonlinear optimization. 1: 1–15
- Townsend S, Grigg S, Picelli R, Featherston C, Kim HA (2019) Topology optimization of vibrational piezoelectric energy harvesters for structural health monitoring applications. *J Intell Mater Syst Struct* 30(18–19):2894–2907
- van Dijk NP, Maute K, Langelaar M, Van Keulen F (2013) Level-set methods for structural topology optimization: a review. *Struct Multidiscip Optim* 48(3):437–472
- Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. *Struct Multidiscip Optim* 43(6):767–784
- Wang X, Lin Z, Ren Y (2017) Topology optimization of piezocomposite resonator for maximizing excitation strength and synthesizing desired eigenmodes. *Acta Mechanica Solida Sinica* 30(5):531–539
- Wei P, Li Z, Li X, Wang MY (2018) An 88-line matlab code for the parameterized level set method based topology optimization using radial basis functions. *Struct Multidiscip Optim* 58(2):831–849
- Wein F, Kaltenbacher M, Baensch E, Leugering G, Schury F (2009) Topology optimization of a piezoelectric-mechanical actuator with single-and multiple-frequency excitation. *Int J Appl Electromagn Mech* 30(3–4):201–221
- Wein F, Kaltenbacher M, Stingl M (2013) Topology optimization of a cantilevered piezoelectric energy harvester using stress norm constraints. *Struct Multidiscip Optim* 48(1):173–185
- Xia L, Xia Q, Huang X, Xie YM (2018) Bi-directional evolutionary structural optimization on advanced structures and materials: a comprehensive review. *Arch Comput Methods Eng* 25(2):437–478
- Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. *Comput Struct* 49(5):885–896
- Yaghmaei M, Ghoddosian A, Khatibi MM (2020) A filter-based level set topology optimization method using a 62-line matlab code. *Struct Multidiscip Optim*
- Zhang W, Chen J, Zhu X, Zhou J, Xue D, Lei X, Guo X (2017) Explicit three dimensional topology optimization via moving morphable void (mmv) approach. *Comput Methods Appl Mech Eng* 322:590–614
- Zheng B, Chang CJ, Gea HC (2009) Topology optimization of energy harvesting devices using piezoelectric materials. *Struct Multidiscip Optim* 38(1):17–23
- Zhu B, Zhang X, Zhang H, Liang J, Zang H, Li H, Wang R (2020) Design of compliant mechanisms using continuum topology optimization: a review. *Mech Mach Theory* 143:103622

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