Modelling structures with piezoelectric materials : full, reduced order, and state-space models

Theory and SDT Tutorial

Etienne Balmes
Arnaud Deraemaeker

SDTools
44 rue Vergniaud
75013 Paris (France)

Tel.	+33 1 44 24 63 71
General info	http://www.sdtools.com
Support	http://support.sdtools.com

Contents

1

Release notes

Contents

1.1 Release notes, SDT 7.6

Major modifications for current release

- New title, general reorganisation and major improvement of the documentation
- New format of graphics and figures with improved style sheets
- New section 4 on general sensors and actuators definition
- New section 5 on methods for meshing plates with piezoelectric patches (including automated methods)
- New section 6 on model reduction and I/O state-space models with emphasis on
  - State-space models with imposed displacement/acceleration/voltage
  - State-space models construction from Craig-Bampton reduced models (also for imposed voltage)
- General reorganisation and addition of numerous scripts to illustrate all concepts presented in the new sections, in particular related to the construction of state-space models (total of 28 tutorial scripts).
- Direct links to scripts for given steps of tutorial scripts in the documentation.
- All tutorial scripts are integrated in the `d_piezo('tuto')` interactive tool.

A total of 28 tutorial scripts are included in this documentation. The simplest way to run the tutorials is to open the gui environment:

```
% Open the gui interface
d_piezo('tuto');
```

Each tutorial can be expanded using the '+' on the left. The green arrows allow to run each step of the script and access that part of the script in the Matlab editor. The white arrows allows to only see the script.

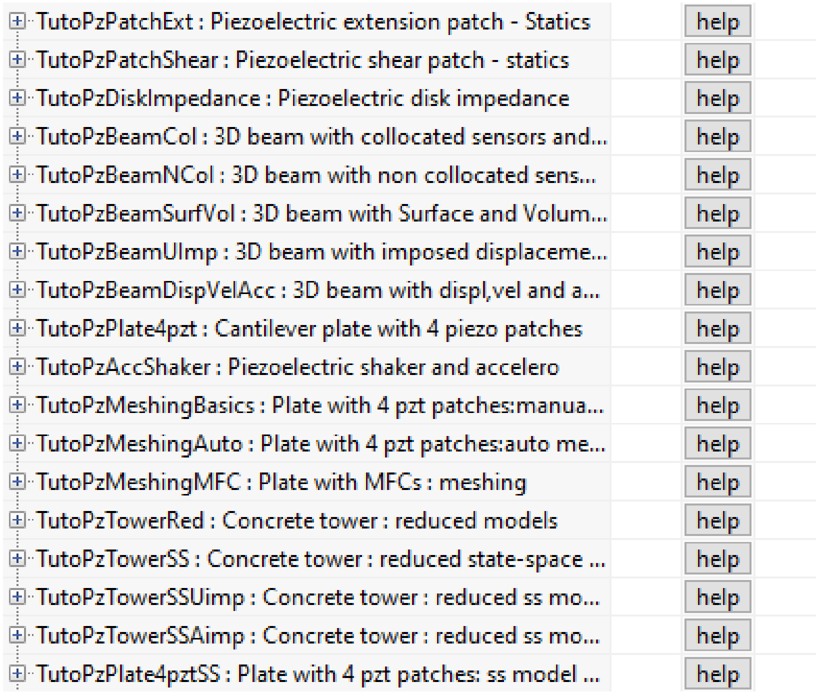


Figure 1.1: gui interface to run tutorials

`d_piezo('Scriptname')` is used to run a specific tutorial script in command line.
`d_piezo('Scriptname-sn')` allows to specify the step, where n is the step number.

```
d_piezo('tutopzpatchext-s1')
```

will run the first step of the demo script in section 3.5.1 .

1.2 Old release notes

Major modifications for SDT 7.1 are

- Inclusion of new materials (Ferroperm, Sonox, MFCs) in the `m_piezo` database.
- Introduction of tutorials in `d_piezo('Tuto')`.
- New script for Macro Fiber Composites in `d_piezo('TutoPlate_mfc')`.
- Theory and new script for point load actuator using a shaped triangular piezoelectric transducer in `d_piezo('TutoPlate_triang')`.

- Theory and new script for vibration damping using RL shunt and piezoelectric patches in `d_piezo('TutoPz_shunt')`.
- Theory and new script for piezoelectric homogenization on RVEs of piezocomposites (application to MFCs) in `d_piezo('TutoPz_P1_homo')` and `d_piezo('TutoPz_P2_homo')` .
- Color visualization of stress and strain added to IDE patch script `d_piezo('TutoPatch_num_IDE')`.

Major modifications for SDT 6.6 were

- Writing of the present manual
- Significant generalization of `p_piezo('Electrode')` commands.
- Inclusion of elastic properties in the `m_piezo` database.
- Introduction of electrical and charge viewing illustrated in this manual.
- Specialized meshing capabilities and examples are grouped in `d_piezo('Mesh')`.

1.3 External links

References to external documents. In SDT use `sdtweb('ref')` to open the page.

- `fe2ss` FEM model to state-space transformation `base/fe2ss`.
- `scell` shape at DOF `base/scell`.
- `general` general definition of sensors `base/scell#general`.
- `resultant` resultant force sensor `base/sstruct#resultant`.
- `2` properties of composite shell `base/p_shell#2`

2

Basics of piezoelectricity

Contents

Polarization consists in the separation of positive and negative electric charges at different ends of the dielectric material on the application of an external electric field (Figure 2.1).

Spontaneous polarization is the phenomenon by which polarization appears without the application of an external electric field. Spontaneous polarization has been observed in certain crystals in which the centers of positive and negative charges do not coincide. Spontaneous polarization can occur more easily in perovskite crystal structures.

The level and direction of the polarization is described by the electric displacement vector D:

$$D = \varepsilon E + P \tag{2.1}$$

where P is the *permanent polarization* which is retained even in the absence of an external electric field, and εE represents the polarization induced by an applied electric field. ε is the dielectric permittivity. If no spontaneous polarization exists in the material, the process through which permanent polarization is induced in a material is known as *poling*.

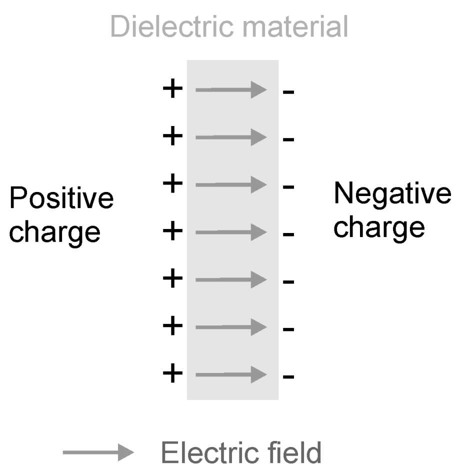


Figure 2.1: Polarization: separation of positive and negative electric charges on the two sides of a dielectric material

Ferroelectric materials have permanent polarization that can be altered by the application of an external electric field, which corresponds to poling of the material. As an example, perovskite structures are ferroelectric below the *Curie temperature*. In the ferroelectric phase, polarization can therefore be induced by the application of a (large) electric field.

Piezoelectricity was discovered by Pierre and Jacques Curie in 1880. The *direct piezoelectric effect* is the property of a material to display electric charge on its surface under the application of an external mechanical stress (i.e. to change its polarization). (Figure 2.2a). *The converse piezoelectric effect* is the production of a mechanical strain due to a change in polarization (Figure 2.2b).

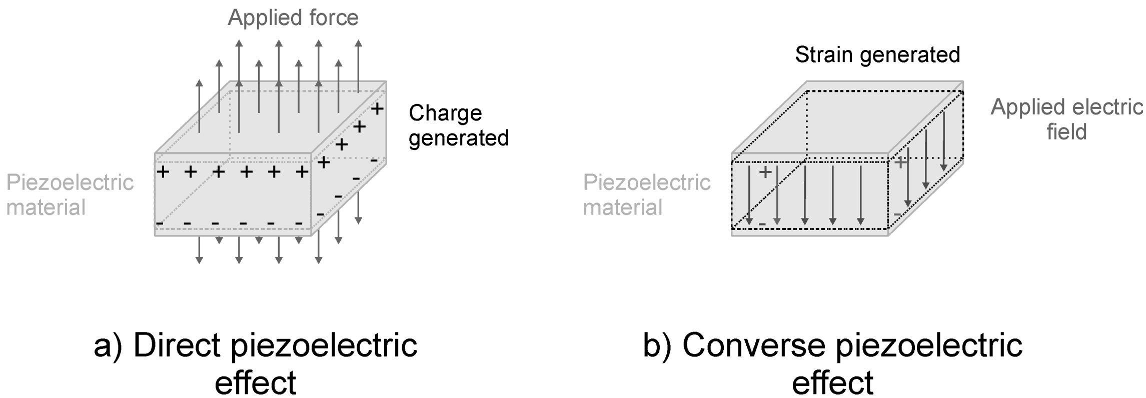


Figure 2.2: Direct and converse piezoelectric effect

Piezoelectricity occurs naturally in non ferroelectric single crystals such as quartz, but the effect is not very strong, although it is very stable. The direct effect is due to a distortion of the crystal lattice caused by the applied mechanical stress resulting in the appearance of electrical dipoles. Conversely, an electric field applied to the crystal causes a distortion of the lattice resulting in an induced mechanical strain. In other materials, piezoelectricity can be induced through *poling*. This can be achieved in *ferroelectric crystals, ceramics or polymers.*

A piezoelectric ceramic is produced by pressing ferroelectric material grains (typically a few micrometers in diameter) together. During fabrication, the ceramic powder is heated (sintering process) above Curie temperature. As it cools down, the perovskite ceramic undergoes phase transformation from the paraelectric state to the ferroelectric state, resulting in the formation of randomly oriented ferroelectric domains. These domains are arranged in grains, containing either 90° or 180° domains (Figure 2.3a). This random orientation leads to zero (or negligible) net polarization and piezoelectric coefficients (Figure 2.3b).

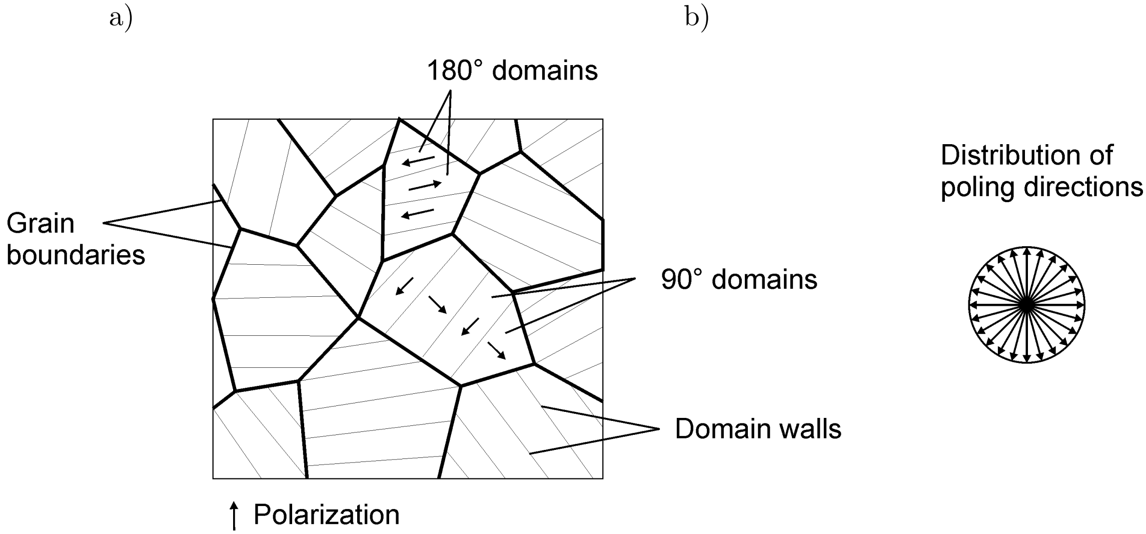


Figure 2.3: Piezoelectric ceramic : a) ferroelectric grains and domains, b) distribution of poling directions

The application of a sufficiently high electric field to the ceramic causes the domains to reorient in the direction of the applied electric field. Note however that the mobility of the domains is not such that all domains are perfectly aligned in the poling direction, but the total net polarization increases with the magnitude of the electric field (Figure 2.4). After removal of the applied electric field, the ferroelectric domains do not return in their initial orientation and a permanent polarization remains in the direction of the applied electric field (the poling direction). In this state, the application of a moderate electric field results in domain motions which are responsible for a deformation of the ceramic and are the source of the piezoelectric effect. The poling direction is therefore a very important material property of piezoelectric materials and needs to be known for a proper modeling.

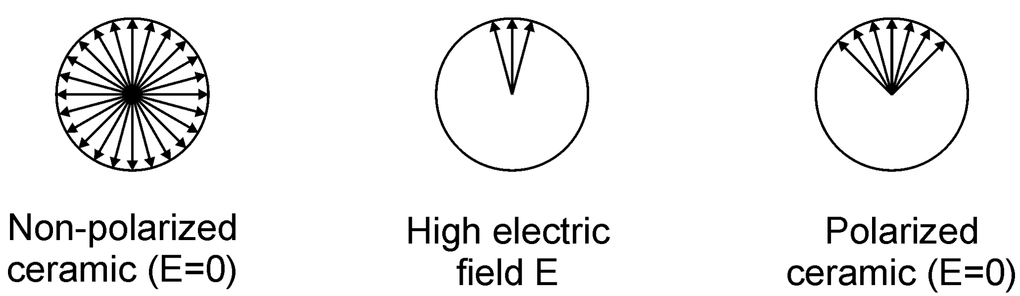


Figure 2.4: Orientation of the ferroelectric domains in non-polarized and polarized ceramics

Typical examples of simple perovskites are Barium titanate ($BaTiO3$) and lead titanate ($PbTiO3$). The most common perovskite alloy is lead zirconate titanate (PZT- PbZr TiO3). Nowadays, the most common ceramic used in piezoelectric structures for structural dynamics applications (active control, shape control, structural health monitoring) is PZT, which will be used extensively in the documented examples.

In certain polymers, piezoelectricity can be obtained by orienting the molecular dipoles within the polymer chain. Similarly to the ferroelectric domains in ceramics, in the natural state, the molecular dipole moments usually cancel each other resulting in an almost zero macroscopic dipole. Poling of the polymer is usually performed by stretching the polymer and applying a very high electric field, which causes the molecular dipoles to orient with the electric field, and remain orientated in this preferential direction after removal of the electric field (permanent polarization). This gives rise to piezoelectricity in the polymer. The technology of piezoelectric polymers has been largely dominated by ferroelectric polymers from the polyvinylidene fluoride (PVDF) family, discovered in 1969. The main advantage is the good flexibility, but their piezoelectric coefficients are much lower compared to ferroelectric ceramics.

2.1 Piezoelectric constitutive laws in 3D

Up to a certain level of electric field and strain, piezoelectric materials behave linearly. This tutorial is restricted to linear piezoelectricity, but the interested reader can refer to [1] for more details on non-linear piezoelectricity.

Assuming a linear piezoelectric material and adopting the notations of the IEEE Standards on piezoelectricity [2], the 3D constitutive equations are given by:

$$\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ D_1 \\ D_2 \\ D_3 \end{Bmatrix} = \begin{bmatrix} c_{11}^E & c_{12}^E & c_{13}^E & 0 & 0 & 0 & 0 & 0 & -e_{31} \\ c_{12}^E & c_{22}^E & c_{23}^E & 0 & 0 & 0 & 0 & 0 & -e_{32} \\ c_{13}^E & c_{23}^E & c_{33}^E & 0 & 0 & 0 & 0 & 0 & -e_{33} \\ 0 & 0 & 0 & c_{44}^E & 0 & 0 & 0 & -e_{24} & 0 \\ 0 & 0 & 0 & 0 & c_{55}^E & 0 & -e_{15} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66}^E & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_{15} & 0 & \varepsilon_{11}^S & 0 & 0 \\ 0 & 0 & 0 & e_{24} & 0 & 0 & 0 & \varepsilon_{22}^S & 0 \\ e_{31} & e_{32} & e_{33} & & 0 & 0 & 0 & 0 & \varepsilon_{33}^S \end{bmatrix} \begin{Bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ E_1 \\ E_2 \\ E_3 \end{Bmatrix} \tag{2.2}$$

where E_i and D_i are the components of the electric field vector and the electric displacement vector, and T_i and S_i are the components of stress and strain vectors, defined according to:

$$\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{Bmatrix} = \begin{Bmatrix} T_{11} \\ T_{22} \\ T_{33} \\ T_{23} \\ T_{13} \\ T_{12} \end{Bmatrix} \qquad \begin{Bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{Bmatrix} = \begin{Bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ 2\,S_{23} \\ 2\,S_{13} \\ 2\,S_{12} \end{Bmatrix} \tag{2.3}$$

Matrix notations are usually adopted leading to:

$$\begin{aligned} \{T\} &= \left[C^E\right]\{S\} - [e]^T\{E\} \\ \{D\} &= [e]\{S\} + \left[\varepsilon^S\right]\{E\} \end{aligned} \tag{2.4}$$

A widely used alternative and equivalent representation consists in writing the constitutive equations in the following form:

$$\begin{aligned} \{S\} &= \left[s^E\right]\{T\} + [d]^T\{E\} \\ \{D\} &= [d]\{T\} + \left[\varepsilon^T\right]\{E\} \end{aligned} \tag{2.5}$$

where the following relationships hold:

$$\left[s^E\right] = \left[c^E\right]^{-1} \tag{2.6}$$

$$[e] = [d]\left[c^E\right] \tag{2.7}$$

$$\left[\varepsilon^S\right] = \left[\varepsilon^T\right] - [d]\,[e]^T \tag{2.8}$$

There are also two additional possibilities to write these constitutive equations, which are less commonly used but are given here for completeness:

$$\begin{aligned} \{S\} &= \left[s^D\right]\{T\} + [g]^T\{D\} \\ \{E\} &= -[g]\{T\} + \left[\beta^T\right]\{D\} \end{aligned} \tag{2.9}$$

$$\begin{aligned} \{T\} &= \left[c^D\right]\{S\} - [h]^T\{D\} \\ \{E\} &= -[h]\{S\} + \left[\beta^S\right]\{D\} \end{aligned} \tag{2.10}$$

The following relationships hold:

$$\left[c^D\right]\left[s^D\right] = I_6 \tag{2.11}$$

$$\begin{aligned}
&\left[\beta^S\right]\left[\varepsilon^S\right] = \left[\beta^T\right]\left[\varepsilon^T\right] = I_3 \\
&\left[c^D\right] = \left[c^E\right] + [e]^T [h] \\
&\left[s^D\right] = \left[c^D\right] - [d]^T [g] \\
&\left[\beta^S\right] = \left[\beta^T\right] - [g]^T [h] \\
&[d] = \left[\varepsilon^T\right][g] \\
&[g] = [h]\left[s^D\right]
\end{aligned} \tag{2.12}$$

$$[h] = \left[\varepsilon^S\right][e] \tag{2.13}$$

The piezoelectric coefficients are contained in the matrix $[d]$ whose structure is specific to each type of piezoelectric material. The typical structure for a z-polarized PZT material is

$$[d] = \begin{bmatrix} 0 & 0 & 0 & 0 & d_{15} & 0 \\ 0 & 0 & 0 & d_{24} & 0 & 0 \\ d_{31} & d_{32} & d_{33} & 0 & 0 & 0 \end{bmatrix} \tag{2.14}$$

Regular PZT ceramics are isotropic in the plane perpendicular to the poling direction ($d_{31} = d_{32}$, $d_{15} = d_{24}$), but piezoelectric composites can have orthotropic properties [3]. PVDF material does not exhibit piezoelectricity in the shear mode, so that the typical structure is:

$$[d] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ d_{31} & d_{32} & d_{33} & 0 & 0 & 0 \end{bmatrix} \tag{2.15}$$

PVDF can be either isotropic or orthotropic in the plane perpendicular to the poling direction, depending on the fabrication process (uni-axial or bi-axial). Table 2.1 gives typical piezoelectric coefficients for PZT ceramics and PVDF films. Note that these properties can vary significantly from the figures in the table, as there are many different material types. The permittivity is usually given with its relative value which is the ratio of the permittivity by the permittivity of vacuum ($\varepsilon_0 = 8.854\,10^{-12} F/m$).

Material properties	PZT	PVDF (bi-axial)
Piezoelectric properties		
d_{33} (pC/N)	440	-25
d_{31} (pC/N)	-185	3
d_{32} (pC/N)	-185	3
Relative permittivity		
ε_r	1800	12
Young's Modulus		
$Y_1(GPa)$	54	3
$Y_2(GPA)$	54	3
$Y_3(GPA)$	48	10
ρ (kg/m^3)	7600	1800

Table 2.1: Typical piezoelectric properties of PZT ceramics and PVDF films

2.2 Piezoelectric constitutive laws in plates

When thin piezoelectric transducers are used with plate structures, the common plane stress hypothesis ($T_3 = 0$) must be used together with an hypothesis for the electric field. When the ceramic is poled through the thickness, the hypothesis commonly adopted is that the electric field is zero in the plane of the transducer ($E_1 = E_2 = 0$). The constitutive equations then reduce to:

$$\begin{Bmatrix} T_1 \\ T_2 \\ T_4 \\ T_5 \\ T_6 \\ D_3 \end{Bmatrix} = \begin{bmatrix} c_{11}^{E*} & c_{12}^{E*} & 0 & 0 & 0 & -e_{31}^{*} \\ c_{12}^{E*} & c_{22}^{E*} & 0 & 0 & 0 & -e_{32}^{*} \\ 0 & 0 & c_{44}^{E*} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{55}^{E*} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{66}^{E*} & 0 \\ e_{31}^{*} & e_{32}^{*} & 0 & 0 & 0 & \varepsilon_{33}^{S*} \end{bmatrix} \begin{Bmatrix} S_1 \\ S_2 \\ S_4 \\ S_5 \\ S_6 \\ E_3 \end{Bmatrix} \tag{2.16}$$

where the superscript * denotes the properties under the "piezoelectric plates" hypothesis ($T_3 = E_1 = E_2 = 0$). These properties are related to the 3D properties with the following relationships:

$$c_{11}^{E*} = \left[c_{11}^{E} - \frac{(c_{13}^{E})^2}{c_{33}^{E}} \right] \tag{2.17}$$

$$c_{12}^{E*} = \left[c_{12}^{E} - \frac{c_{13}^{E}\, c_{23}^{E}}{c_{33}^{E}} \right] \tag{2.18}$$

$$c_{22}^{E*} = \left[c_{22}^{E} - \frac{(c_{23}^{E})^2}{c_{33}^{E}} \right] \tag{2.19}$$

$$e_{31}^{*} = \left[e_{31} - \frac{c_{13}^{E}\, e_{33}}{c_{33}^{E}} \right] \tag{2.20}$$

$$e_{32}^{*} = \left[e_{32} - \frac{c_{23}^{E}\, e_{33}}{c_{33}^{E}} \right] \tag{2.21}$$

$$\varepsilon_{33}^{S*} = \left[\varepsilon_{33}^{S} + \frac{(e_{33})^2}{c_{33}^{E}} \right] \tag{2.22}$$

The distinction is very important, as it is often not well understood and many errors can arise from the confusion between plate and 3D properties of piezoelectric materials. Note however that the d_{ij}, s_{ij}^{E} and ε^{T} coefficients are equal for plate and 3D constitutive equations. It is therefore preferable to handle the material properties of piezoelectric materials in the form of (2.5).

Similarly to the 3D equations, the constitutive equations can be written in a matrix form, separating the mechanical and the electrical parts:

$$\begin{aligned} \{T\} &= \left[c^{E*}\right]\{S\} - \left[e^{*}\right]^{T}\{E\} \\ \{D\} &= \left[e^{*}\right]\{S\} + \left[\varepsilon^{S*}\right]\{E\} \end{aligned} \tag{2.23}$$

Using (2.7) in equations ((2.20),(2.21),(2.22)), one can further show that

$$\left[e^{*}\right] = \left[d^{*}\right]\left[c^{E*}\right] \tag{2.24}$$

and for the permittivity:

$$\varepsilon_{33}^{S*} = \varepsilon_{33}^{T} - \left[d^{*}\right]\left[e^{*}\right]^{T} \tag{2.25}$$

with

$$\left[d^{*}\right] = \left[\begin{array}{ccccc} d_{31} & d_{32} & 0 & 0 & 0 \end{array}\right] \tag{2.26}$$

and

$$\left[e^{*}\right] = \left[\begin{array}{ccccc} e_{31}^{*} & e_{32}^{*} & 0 & 0 & 0 \end{array}\right] \tag{2.27}$$

The values of e_{31}^{*}, e_{32}^{*} and ε_{33}^{S*} can therefore be computed knowing the elastic matrix $\left[c^{E*}\right]$ and the values of d_{31} and d_{32} and ε_{33}^{T}

2.3 Database of piezoelectric materials

`m_piezo` `Dbval` includes a number of material characteristics for piezoelectric materials. The properties are obtained from the datasheet of the material, but as we will illustrate, the data is not always sufficient to calculate all the material properties needed for the computations. Most of the information in the datasheet is generally related to the constitutive equations written in the form of (2.5). For PZT, PVDF, or piezoelectric composites based on PZT and PVDF, the general form of these matrices is:

$$\begin{Bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ D_1 \\ D_2 \\ D_3 \end{Bmatrix} = \begin{bmatrix} s_{11}^E & s_{12}^E & s_{13}^E & 0 & 0 & 0 & 0 & 0 & d_{31} \\ s_{12}^E & s_{22}^E & s_{23}^E & 0 & 0 & 0 & 0 & 0 & d_{32} \\ s_{13}^E & s_{23}^E & s_{33}^E & 0 & 0 & 0 & 0 & 0 & d_{33} \\ 0 & 0 & 0 & s_{44}^E & 0 & 0 & 0 & d_{24} & 0 \\ 0 & 0 & 0 & 0 & s_{55}^E & 0 & d_{15} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s_{66}^E & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{15} & 0 & \varepsilon_{11}^T & 0 & 0 \\ 0 & 0 & 0 & d_{24} & 0 & 0 & 0 & \varepsilon_{22}^T & 0 \\ d_{31} & d_{32} & d_{33} & & 0 & 0 & 0 & 0 & \varepsilon_{33}^T \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ E_1 \\ E_2 \\ E_3 \end{Bmatrix} \tag{2.28}$$

For an orthotropic material, the compliance matrix $\left[s^E\right]$ can be written as a function of the engineering constant E_i,ν_{ij} and G_{ij} as follows:

$$\left[s^E\right] = \begin{bmatrix} \frac{1}{E_x} & \frac{-\nu_{yx}}{E_y} & \frac{-\nu_{zx}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_{xy}}{E_x} & \frac{1}{E_y} & \frac{-\nu_{zy}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_{xz}}{E_x} & \frac{-\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{xz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix} \tag{2.29}$$

where z is aligned with the poling direction 3, and x, y with directions $1, 2$ respectively. Note that the matrix is symmetric so that:

$$\frac{\nu_{yx}}{E_y} = \frac{\nu_{xy}}{E_x}, \quad \frac{\nu_{zx}}{E_z} = \frac{\nu_{xz}}{E_x}, \quad \frac{\nu_{zy}}{E_z} = \frac{\nu_{yz}}{E_y} \tag{2.30}$$

A bulk piezoelectric ceramic exhibits transverse isotropic properties: the properties of the material are the same in the plane perpendicular to the poling direction. In this case, the compliance matrix reduces to:

$$\left[s^E\right] = \begin{bmatrix} \frac{1}{E_p} & \frac{-\nu_p}{E_p} & \frac{-\nu_{zp}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_p}{E_p} & \frac{1}{E_p} & \frac{-\nu_{zp}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_{pz}}{E_p} & \frac{-\nu_{pz}}{E_p} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{zp}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{zp}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2(1+\nu_p)}{E_p} \end{bmatrix} \tag{2.31}$$

and due to the symmetry we have:

$$\frac{\nu_{zp}}{E_z} = \frac{\nu_{pz}}{E_p} \tag{2.32}$$

where the subscript p refers to the in-plane properties. The matrix of piezoelectric coefficients is:

$$[d] = \begin{bmatrix} 0 & 0 & 0 & 0 & d_{15} & 0 \\ 0 & 0 & 0 & d_{15} & 0 & 0 \\ d_{31} & d_{31} & d_{33} & 0 & 0 & 0 \end{bmatrix} \tag{2.33}$$

and the matrix of dielectric permittivities:

$$\left[\varepsilon^T\right] = \begin{bmatrix} \varepsilon_{11}^T & 0 & 0 \\ 0 & \varepsilon_{11}^T & 0 \\ 0 & 0 & \varepsilon_{33}^T \end{bmatrix} \tag{2.34}$$

In order to use such a piezoelectric material in a 3D model, it is therefore necessary to have access to the 5 elastic constants E_p,E_z,ν_p,ν_{zp} and G_{zp}, 3 piezoelectric constants d_{31},d_{33}, and d_{15} and two dielectric constants ε_{11}^T,ε_{33}^T. Unfortunately, such constants are generally not given in that form, but can be calculated from the material properties found in the datasheet.

It is important to introduce the electromechanical coupling factors which are generally given in the datasheet and are a function of the elastic, piezoelectric and dielectric properties of the material. They measure the effectiveness of the conversion of mechanical energy into electrical energy (and vice-versa). There is one coupling factor for each piezoelectric mode:

$$\begin{aligned} k_{31}^2 &= \frac{d_{31}^2}{\varepsilon_{33}^T s_{11}^E} \\ k_{33}^2 &= \frac{d_{33}^2}{\varepsilon_{33}^T s_{33}^E} \\ k_{15}^2 &= \frac{d_{15}^2}{\varepsilon_{11}^T s_{55}^E} \end{aligned} \tag{2.35}$$

In addition, coupling factors k_p for radial modes of thin discs, and k_t for thickness modes of arbitrary shaped thin plates are also commonly given in datasheet. k_p is related to k_{31} through:

$$k_p^2 = \frac{2k_{31}^2}{1 + \frac{s_{12}^E}{s_{11}^2}} \tag{2.36}$$

k_t is always lower than k_{33} but there does not seem to be a simple explicit expression of k_t as a function of the material properties. The fact that k_t is lower than k_{33} means that electrical energy conversion in the d_{33}-mode is less effective for a thin plate than for a rod. The definition of the coupling factors k_{33} and k_{15} also allows to write alternative expressions:

$$\begin{aligned} k_{33}^2 &= 1 - \frac{s_{33}^D}{s_{33}^E} \\ k_{15}^2 &= 1 - \frac{s_{55}^D}{s_{55}^E} = 1 - \frac{\varepsilon_{11}^S}{\varepsilon_{11}^T} \end{aligned} \tag{2.37}$$

We illustrate the use of these different relationships to form the full set of mechanical, piezoelectric and dielectric properties for the material *SONOX P502* from Ceramtec (http://www.ceramtec.com/) which is a soft piezoceramic. The properties found in the datasheet on matweb.com are given in Table 2.2 .

Material property	value	unit
Piezoelectric properties		
d_{33}	440	$10^{-12} m/V$
d_{31}	-185	$10^{-12} m/V$
d_{15}	560	$10^{-12} m/V$
e_{33}	16.7	$C/m^2 = As/m^2$
g_{33}	$26.9\ 10^{-3}$	Vm/N
Permittivity		
ε_{33}^T	$1850\ \varepsilon_0$	F/m
ε_{33}^S	$875\ \varepsilon_0$	F/m
ε_{11}^T	$1950\ \varepsilon_0$	F/m
ε_{11}^S	$1260\ \varepsilon_0$	F/m
Elastic properties		
s_{11}^E	$18.5\ 10^{-12}$	m^2/N
s_{33}^E	$20.7\ 10^{-12}$	m^2/N
c_{33}^D	$15.7\ 10^{10}$	N/m^2
c_{55}^D	$6.5\ 10^{10}$	N/m^2
Coupling coefficients		
k_{33}	0.72	
k_{15}	0.74	
k_{31}	0.33	
k_p	0.62	
k_t	0.48	
Density		
ρ	7740	kg/m^3

Table 2.2: Properties of *SONOX P502* from the datasheet found on https://www.matweb.com (2013)

E_p and E_z are computed directly from the definitions of s_{11}^E and s_{33}^E:

$$E_p = \frac{1}{s_{11}^E} = 54.05GPa \tag{2.38}$$

$$E_z = \frac{1}{s_{33}^E} = 48.31GPa \tag{2.39}$$

Knowing the value of s_{11}^E, d_{31}, ε_{33}^T and k_p, s_{12}^E can be computed:

$$s_{12}^E = -s_{11}^E + 2\frac{d_{31}^2}{k_p^2 \varepsilon_{33}^T} = -7.6288\,10^{-12} m^2/N$$

allowing to compute the value of ν_p:

$$\nu_p = -E_p s_{12}^E = 0.4124$$

and the value of G_p

$$G_p = \frac{E_p}{2(1+\nu_p)} = 19.17GPa$$

From the value c_{55}^D and k_{15}, we compute

$$s_{55}^E = \frac{1}{c_{55}^D (1-k_{15}^2)} = 34\,10^{-12} m^2/N$$

from which the the value of G_{zp} is computed:

$$G_{zp} = \frac{1}{s_{55}^E} = 29.41GPa$$

The value of ν_{zp} cannot be calculated from the datasheet information. We therefore assume that, as for most PZT ceramics:

$$\nu_{zp} = 0.39$$

The value of ν_{pz} is calculated as:

$$\nu_{pz} = \frac{E_p}{E_z}\nu_{zp} = 0.44$$

The complete set of values is summarized in Table 2.3. These are the values used in `m_piezo`.

Material property	value	unit
Piezoelectric properties		
d_{33}	440	$10^{-12}m/V$
d_{31}	-185	$10^{-12}m/V$
d_{15}	560	$10^{-12}m/V$
Permittivity		
ε_{33}^T	1850 ε_0	F/m
ε_{11}^T	1950 ε_0	F/m
Mechanical properties		
E_p	54.05	GPa
E_z	48.31	GPa
G_{zp}	29.41	GPa
G_p	19.17	GPa
ν_p	0.4124	
ν_{zp}	0.39	
ν_{pz}	0.44	
ρ	7740	kg/m^3

Table 2.3: Properties of *SONOX P502* to be used in 3D finite element models

Note that there is some redundancy in the data from the datasheet, which allows to check for consistency. The two following coupling factors are computed from the data available and checked against the tabulated values.

$$k_{31} = \sqrt{\frac{d_{31}^2}{\varepsilon_{33}^T s_{11}^E}} = 0.3361$$

$$k_{33} = \sqrt{\frac{d_{33}^2}{\varepsilon_{33}^T s_{33}^E}} = 0.7556$$

The values are close to the values in Table 2.2. In addition, the value of g_{33} is given by:

$$g_{33} = \frac{d_{33}}{\varepsilon_{33}^T} = 0.0269Vm/N$$

and corresponds exactly to the value tabulated. The value of e_{33} can be computed using Equation (2.7), leading to:

$$e_{33} = 19.06C/m^2$$

where there is a difference of about 15% with the tabulated value of $e_{33} = 16.7C/m^2$. Note however that this last value was found on matweb.com and is not given in the more recent datasheet on

Ceramtec website (in 2023).

Using (2.37) to compute k_{15} with the values from the datasheet, one gets:

$$k_{15} = \sqrt{1 - \frac{\varepsilon_{11}^S}{\varepsilon_{11}^T}} = 0.5948$$

which shows the non-consistency of the value of ε_{11}^S in the datasheet. In fact, when computed using (2.8), one gets:

$$\varepsilon_{11}^S = 908\varepsilon_0$$

This illustrates the fact that it is difficult to obtain the full set of parameters needed for computation for piezoelectric materials, as there are often some inconsistencies amongst the data available from the manufacturers. What we believe is a ”best compromise” was used in the material properties available in SDT.

From the input values in `m_piezo` (Table 2.3), it is possible to compute the mechanical, piezoelectric and permittivity matrices used in the four different forms of the constitutive equations (2.4),(2.5),(2.9),(2.10) using the relationships (2.6)-(2.8)) and (2.11)-(2.13).

The command `p_piezo('TabDD',model)` gives access to all the matrices based in the input values in `m_piezo`. This will be illustrated in section section 3.5.1 .

As the mechanical properties of PZT are not strongly orthotropic, a simplification can be done by considering that the material is isotropic (for the mechanical and dielectric properties, not the piezoelectric properties). An isotropic version of *SONOX P502* is included in `m_piezo` under the name of *SONOX_P502_iso* whose properties are given in Table 2.4.

Material property	value	unit
Piezoelectric properties		
d_{33}	440	$10^{-12} m/V$
d_{31}	-185	$10^{-12} m/V$
d_{15}	560	$10^{-12} m/V$
Permittivity		
ε^T	1850 ε_0	F/m
Mechanical properties		
E	54	GPa
ν	0.41	
ρ	7740	kg/m^3

Table 2.4: Simplified material properties for *SONOX P502* considering mechanical isotropy

The second example is the *PIC 255* PZT, also a soft piezoceramic, from PI ceramics. The properties found in the datasheet in the year 2013 are given in Table 2.5 (https://www.piceramic.com). Note that C_{33}^D was not given, therefore we estimated it from the value of *PIC 155* given in the same datasheet, which is just slightly stiffer.

Material property	value	unit
Piezoelectric properties		
d_{33}	400	$10^{-12} m/V$
d_{31}	-180	$10^{-12} m/V$
d_{15}	550	$10^{-12} m/V$
g_{31}	-11.3 10^{-3}	Vm/N
g_{33}	25 10^{-3}	Vm/N
Permittivity		
ε_{33}^T	1750 ε_0	F/m
ε_{11}^T	1650 ε_0	F/m
Elastic properties		
s_{11}^E	16.1 10^{-12}	m^2/N
s_{33}^E	20.7 10^{-12}	m^2/N
c_{33}^D	11 10^{10}	N/m^2
Coupling coefficients		
k_{33}	0.69	
k_{15}	0.66	
k_{31}	0.35	
k_p	0.62	
k_t	0.47	
Density		
ρ	7800	kg/m^3

Table 2.5: Properties of *PIC 255* from the datasheet (2013)

E_p and E_z are computed directly from the definitions of s_{11}^E and s_{33}^E:

$$E_p = \frac{1}{s_{11}^E} = 62.11\, GPa$$

$$E_z = \frac{1}{s_{33}^E} = 48.31\, GPa$$

Knowing the value of s_{11}^E, d_{31}, ε_{33}^T and k_p, s_{12}^E can be computed:

$$s_{12}^E = -s_{11}^E + 2\frac{d_{31}^2}{k_p^2 \varepsilon_{33}^T} = -5.22\, 10^{-12}\, m^2/N$$

allowing to compute the value of ν_p:

$$\nu_p = -E_p s_{12}^E = 0.3242$$

and the value of G_p

$$G_p = \frac{E_p}{2(1+\nu_p)} = 23.53\,GPa$$

The value of s_{55}^E can be computed as:

$$s_{55}^E = \frac{d_{15}^2}{\varepsilon_{11}^T k_{15}^2} = 4.75\,10^{-11}\,m^2/N$$

which leads to:

$$G_{zp} = \frac{1}{s_{55}^E} = 21.03\,GPa$$

Again, the value of ν_{zp} cannot be calculated from the datasheet information. We cannot assume a value of 0.39 as previously, as it would lead to a non-physical value of ν_{pz}. As ν_p is in the range of 0.32 and ν_{zp} is typically slightly lower, we assume that :

$$\nu_{zp} = 0.30$$

The value of ν_{pz} is calculated as:

$$\nu_{pz} = \frac{E_p}{E_z}\nu_{zp} = 0.39$$

The complete set of values is summarized in Table 2.6. These are the values used in `m_piezo`. Note that there is some redundancy in the data from the datasheet, which allows to check for consistency. The two following coupling factors are computed from the data available and checked against the tabulated values.

$$k_{31} = \sqrt{\frac{d_{31}^2}{\varepsilon_{33}^T s_{11}^E}} = 0.36$$
$$k_{33} = \sqrt{\frac{d_{33}^2}{\varepsilon_{33}^T s_{33}^E}} = 0.70$$

The values are very close to the values in Table 2.5. In addition, the value of g_{33} and g_{31} are given by:

$$g_{31} = \frac{d_{31}}{\varepsilon_{33}^T} = -11.6\,10^{-3}\,Vm/N$$
$$g_{33} = \frac{d_{33}}{\varepsilon_{33}^T} = 25.8\,10^{-3}\,Vm/N$$

and are also very close to the values tabulated.

Material property	value	unit
Piezoelectric properties		
d_{33}	400	$10^{-12}m/V$
d_{31}	-180	$10^{-12}m/V$
d_{15}	550	$10^{-12}m/V$
Permittivity		
ε_{33}^T	1750 ε_0	F/m
ε_{11}^T	1650 ε_0	F/m
Mechanical properties		
E_p	62.11	GPa
E_z	48.31	GPa
G_{zp}	21.03	GPa
G_p	23.53	GPa
ν_p	0.3242	
ν_{zp}	0.30	
ν_{pz}	0.39	
ρ	7800	kg/m^3

Table 2.6: Properties of *PIC 255* to be used in 3D finite element models from datasheet in 2013

As shown in the derivations above, the datasheet for PZT material typically do not contain the full information to derive all the coefficients needed for computations, and some hypothesis need to be made. In addition, it is usual to have a variation of 10 % or more on these properties from batch to batch, and the datasheet are not updated for each batch. Note also that the properties are given at 20 °C and are temperature dependant. The variations with temperature are rarely given in the datasheet. This may also account for inaccuracies in the computations.

The more recent datasheet found on PI Ceramics website (2023) leads to slightly different properties, and includes the value of C_{33}^D, which gives a more precise value for E_z. The new datasheet information is given in Table 2.7 and the resulting `m_piezo` input parameters in Table 2.8. The updated properties are included in the *PIC255b* material in `m_piezo`. It is advised to use this updated material property, as the main difference is for the Young's modulus in the direction of poling. This parameter has an important impact on the longitudinal natural frequency of disks and rods.

Material property	value	unit
Piezoelectric properties		
d_{33}	400	$10^{-12} m/V$
d_{31}	-180	$10^{-12} m/V$
d_{15}	550	$10^{-12} m/V$
g_{31}	-11.8 10^{-3}	Vm/N
g_{33}	25 10^{-3}	Vm/N
Permittivity		
ε_{33}^T	1800 ε_0	F/m
ε_{11}^T	1750 ε_0	F/m
Elastic properties		
s_{11}^E	16 10^{-12}	m^2/N
s_{33}^E	19 10^{-12}	m^2/N
c_{33}^D	15.4 10^{10}	N/m^2
Coupling coefficients		
k_{33}	0.69	
k_{15}	0.65	
k_{31}	0.35	
k_p	0.62	
k_t	0.47	
Density		
ρ	7800	kg/m^3

Table 2.7: Properties of *PIC 255b* from the datasheet (2023)

Material property	value	unit
Piezoelectric properties		
d_{33}	400	$10^{-12}m/V$
d_{31}	-180	$10^{-12}m/V$
d_{15}	550	$10^{-12}m/V$
Permittivity		
ε_{33}^T	1800 ε_0	F/m
ε_{11}^T	1750 ε_0	F/m
Mechanical properties		
E_p	62.5	GPa
E_z	52.63	GPa
G_{zp}	21.64	GPa
G_p	23.39	GPa
ν_p	0.3389	
ν_{zp}	0.30	
ν_{pz}	0.3562	
ρ	7800	kg/m^3

Table 2.8: Properties of *PIC 255b* to be used in 3D finite element models from datasheet in 2023

In much the same way, the material properties of *PIC 181* which is a hard piezoceramic from the same manufacturer have been updated from the *PIC181* to the *PIC181b* properties in `m_piezo`.

Table 2.9 summarizes the different material properties available in SDT, and the year of the datasheet where the original data was found.

Manufacturer	type	year	SDT name
Ceramtec	Sonox P502	2023	SONOX_P502
Ceramtec	Sonox P502 - simplified	2023	SONOX_P502_iso
PI Ceramics	PIC181	2013	PIC181
PI Ceramics	PIC181	2023	PIC181b
PI Ceramics	PIC255	2013	PIC255
PI Ceramics	PIC255	2023	PIC255b
Ferroperm	Pz21	2018	FerropermPz21
Ferroperm	Pz23	2018	FerropermPz23
Ferroperm	Pz24	2018	FerropermPz24
Ferroperm	Pz26	2018	FerropermPz26
Ferroperm	Pz27	2018	FerropermPz27
Ferroperm	Pz28	2018	FerropermPz28
Ferroperm	Pz29	2018	FerropermPz29
Ferroperm	Pz34	2018	FerropermPz34
Ferroperm	Pz46	2018	FerropermPz46
Noliac	NCE51	2012	Noliac.NCE51

Table 2.9: Piezoelectric materials available in SDT: manufacturer references and year, and SDT name

2.4 Illustration of piezoelectricity in statics: patch example

2.4.1 Patch in extensional mode

Consider a thin piezoelectric patch of dimensions b x h x w. The poling direction, noted 3 in the IEEE Standards on piezoelectricity is perpendicular to the plane of the piezoelectric patch. Continuous electrodes are present on the top and bottom surfaces ($z = 0$, $z = h$) so that the electric potential is constant on these surfaces and denoted by V_1 and V_2 respectively. We assume that a difference of potential is applied between the electrodes, resulting in an electric field parallel to the poling direction and equal to (Figure 2.5)

$$E_3 = -\frac{dV}{dz} = \frac{-(V_2 - V_1)}{h} = \frac{V_1 - V_2}{h}$$

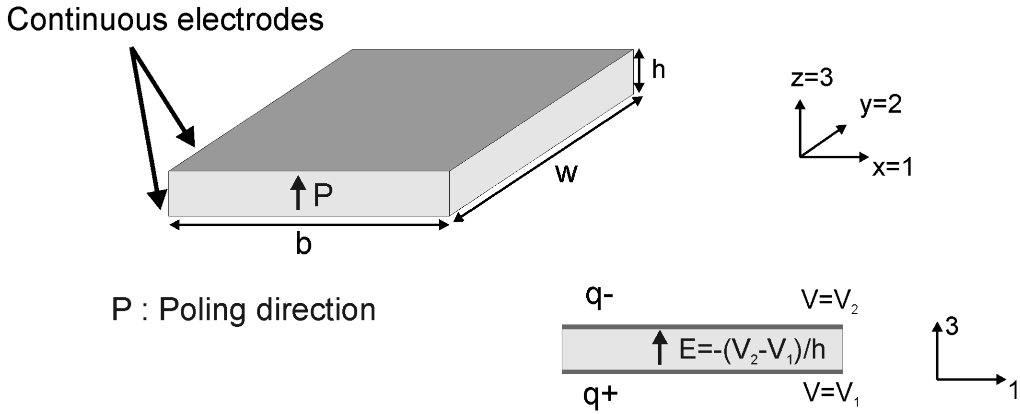


Figure 2.5: A piezoelectric patch poled through the thickness with continuous electrodes on the top and bottom surfaces

We adopt the following expression for the constitutive equations:

$$\begin{array}{l} \{S\} = \left[s^E\right]\{T\} + [d]^T\{E\} \\ \{D\} = [d]\{T\} + \left[\varepsilon^T\right]\{E\} \end{array} \tag{2.40}$$

The patch is assumed to be unconstrained so that it can expand freely, leading to $\{T\} = 0$, so that we have :

$$\{S\} = \left\{ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{array} \right\} = [d]^T\{E\} = \left\{ \begin{array}{c} d_{31}\frac{V_1-V_2}{h} \\ d_{32}\frac{V_1-V_2}{h} \\ d_{33}\frac{V_1-V_2}{h} \\ 0 \\ 0 \\ 0 \end{array} \right\} \tag{2.41}$$

We have taken into account the fact that the electric field is in the z-direction only. This shows that when applying a difference of potential across the thickness (in the poling direction), strains will be induced in the directions 1,2, and 3. The magnitude of these different strains is proportional to the d_{3i} coefficients of the piezoelectric material. For a ceramic PZT material, $d_{31} = d_{32} < 0$, and $d_{33} > 0$ and is generally between 2 and 3 times larger in magnitude than d_{31} and d_{32}.

The second equation can be used in order to assess the amount of charge that is accumulated on both electrodes. We have :

$$\{D\} = \left\{ \begin{array}{c} D_1 \\ D_2 \\ D_3 \end{array} \right\} = \left[\varepsilon^T\right]\{E\} \tag{2.42}$$

The only non-zero component of the D vector is D_3 given by :

$$D_3 = \varepsilon_{33}^T \frac{V_1 - V_2}{h} \tag{2.43}$$

The charge accumulated on the electrode is given by :

$$q = -\int_S \{D\} \{n\} \, dS$$

where $\{n\}$ is the normal to the electrode. For the top electrode, this leads to :

$$q = -\frac{\varepsilon_{33}^T A}{h}(V_1 - V_2)$$

where A is the surface of the electrode. For the bottom electrode

$$q = \frac{\varepsilon_{33}^T A}{h}(V_1 - V_2)$$

When $(V_1 - V_2)$ is positive, the electric field is in the direction of poling and the charge on the top electrode is negative, while the charge accumulated on the bottom electrode is positive (Figure 2.5). Note that this equation corresponds to the equation linking the charge to the difference of potential for a capacitor ($q = C\Delta V$). The value of the capacitance is therefore :

$$C^T = \frac{\varepsilon_{33}^T A}{h}$$

which corresponds to the capacitance of the free piezoelectric patch ($\{T\} = 0$).

If we now consider the case where the piezoelectric patch is fully mechanically constrained ($\{S\} = 0$), we have:

$$\begin{aligned} \{T\} &= -\left[e\right]^T \{E\} = -\left[e\right]^T \{E\} \\ \{D\} &= \left[\varepsilon^S\right] \{E\} \end{aligned} \tag{2.44}$$

leading to :

$$\{T\} = \left\{ \begin{array}{c} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{array} \right\} = \left\{ \begin{array}{c} -e_{31}\frac{V_1-V_2}{h} \\ -e_{32}\frac{V_1-V_2}{h} \\ -e_{33}\frac{V_1-V_2}{h} \\ 0 \\ 0 \\ 0 \end{array} \right\} \tag{2.45}$$

$$D_3 = \varepsilon_{33}^S \frac{V_1-V_2}{h}$$

The fact that the patch is not allowed to expand is responsible for the generation of internal stresses which are proportionnal to the e_{3i} coefficients. In this case, the capacitance is given by:

$$C^S = \frac{\varepsilon_{33}^S A}{h}$$

which corresponds to the capacitance of the constrained piezoelectric patch ($\{S\} = 0$). This illustrates the fact that the capacitance of a piezoelectric patch depends on the mechanical boundary conditions. This is not the case for other types of dielectric materials in which the piezoelectric effect is not present, and for which therefore the capacitance is independent on the mechanical strain or stress.

2.4.2 Patch in shear mode

We now consider the same patch but where the polarization is in the plane of the actuator, as represented in Figure 2.6. As in the previous example, continuous electrodes are present on the top and bottom surfaces ($z = 0$, $z = h$) so that the electric potential is constant on these surfaces and denoted by V_1 and V_2 respectively. We assume that a difference of potential is applied between the electrodes, resulting in an electric field perpendicular to the poling direction and equal to

$$E_2 = -\frac{dV}{dz} = \frac{-(V_2 - V_1)}{h} = \frac{V_1 - V_2}{h}$$

The electric field is now applied in direction 2, so that it will activate the shear $d_{24} = d_{15}$ mode of the piezoelectric material.

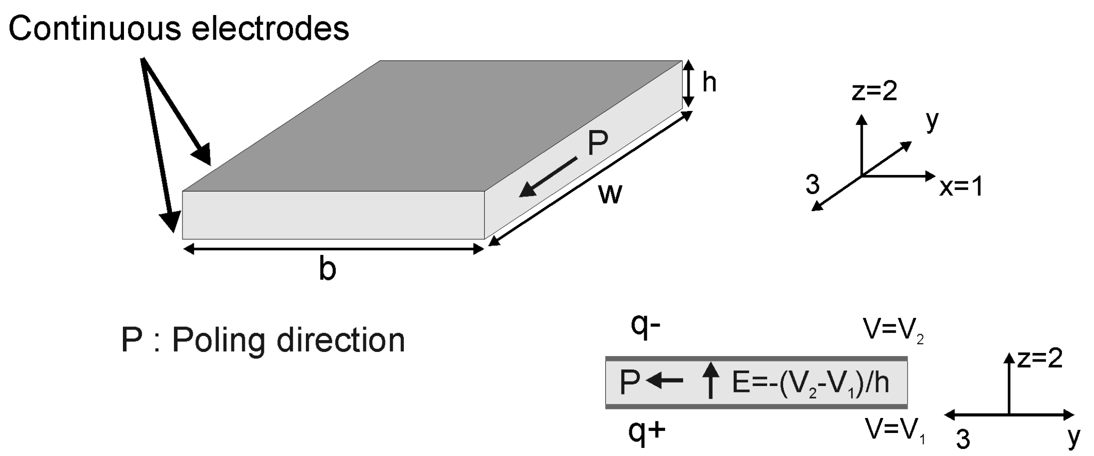


Figure 2.6: A piezoelectric patch poled in the plane with continuous electrodes on the top and bottom surfaces

The patch is assumed to be unconstrained so that it can expand freely, leading to $\{T\} = 0$, so that we have :

$$\{S\} = \left\{ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{array} \right\} = [d]^T \{E\} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ d_{24}\frac{V_1 - V_2}{h} \\ 0 \\ 0 \end{array} \right\} \tag{2.46}$$

We have taken into account the fact that the electric field is in the z-direction only, corresponding to direction 2 in the local axis of the piezoelectric material (direction 3 is the poling direction by convention). This shows that when the patch is poled in the plane, when applying a difference of potential across the thickness, a shear strain in the local 23 plane will be induced. The magnitude of this strain is proportional to the d_{24} coefficient of the piezoelectric material.

The second equation can be used in order to assess the amount of charge that is accumulated on both electrodes. We have :

$$\{D\} = \left\{ \begin{array}{c} D_1 \\ D_2 \\ D_3 \end{array} \right\} = \left[\varepsilon^T\right] \{E\} \tag{2.47}$$

The only non-zero component of the D vector is D_2 given by :

$$D_2 = \varepsilon_{22}^T \frac{V_1 - V_2}{h} \tag{2.48}$$

The charge accumulated on the electrode is given by :

$$q = -\int_S \{D\} \{n\} \, dS$$

where $\{n\}$ is the normal to the electrode. For the top electrode, this leads to :

$$q = -\frac{\varepsilon_{22}^T A}{h}(V_1 - V_2)$$

where A is the surface of the electrode. For the bottom electrode

$$q = \frac{\varepsilon_{22}^T A}{h}(V_1 - V_2)$$

When $(V_1 - V_2)$ is positive, the charge on the top electrode is negative, while the charge accumulated on the bottom electrode is positive (Figure 2.6). The value of the capacitance is therefore:

$$C^T = \frac{\varepsilon_{22}^T A}{h}$$

which corresponds to the capacitance of the free piezoelectric patch ($\{T\} = 0$) and is close to the capacitance when the poling is out of the plane of the transducer because $\varepsilon_{22}^T \simeq \varepsilon_{33}^T$ (in reality, there is typically a difference of 5% between these two values so that the capacitance will be slightly different).

If we now consider the case where the piezoelectric patch is fully mechanically constrained ($\{S\} = 0$), we have:

$$\begin{array}{c} \{T\} = -\left[e\right]^T \{E\} = -\left[e\right]^T \{E\} \\ \{D\} = \left[\varepsilon^S\right] \{E\} \end{array} \tag{2.49}$$

leading to :

$$\{T\} = \left\{ \begin{array}{c} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ -e_{24}\frac{V_1-V_2}{h} \\ 0 \\ 0 \end{array} \right\} \tag{2.50}$$

$$D_2 = \varepsilon_{22}^S \frac{V_1-V_2}{h}$$

In this case, the capacitance is given by:

$$C^S = \frac{\varepsilon_{22}^S A}{h}$$

which corresponds to the capacitance of the constrained piezoelectric patch activated in shear ($\{S\} = 0$). Note that this capacitance is clearly different from C^S when the poling is out of the plane, because the value of ε_{22}^S is very different from the value of ε_{33}^S, due to the different values of stiffness and piezoelectric coefficients in shear and extensional mode.

3

Finite element formulations for piezoelectric structures

Contents

Hamilton's principle is used to derive the dynamic variational principle [1]:

$$\begin{aligned}
&\int_{t_1}^{t_2} \left(\int_V \left[-\rho \{\ddot{u}\}^T \{\delta u\} - \{S\}^T \left[c^E\right] \{\delta S\} + \{E\}^T [e] \{\delta S\} \right.\right. \\
&+ \{S\}^T [e]^T \{\delta E\} + \{E\}^T \left[\varepsilon^S\right] \{\delta E\} + \{f\}^T \{\delta u\} - \{\rho_e\}^T \{\delta\phi\} \Big] \, dV \\
&\left. + \int_{\Omega_1} \{t\}^T \{\delta u\} \, d\Omega - \int_{\Omega_2} \{\sigma\}^T \{\delta\phi\} \, d\Omega \right) dt = 0
\end{aligned} \tag{3.1}$$

where V is the volume of the piezoelectric structure, ρ is the mass density, $\{u\}$ is the displacement field and $\{\delta u\}$ its variation, $\{\phi\}$ is the electric pontential and $\{\delta\phi\}$ its variation. $\{f\}$ is the volumic force, $\{\rho_e\}$ the volumic charge density, $\{t\}$ the vector of applied surface forces on Ω_1 and $\{\sigma\}$ the charge density applied on Ω_2. The variational principle is the starting point for all discrete finite element formulations. 3D and shell approximations are detailed below.

3.1 Piezoelectric solid finite elements

For 3D solids, the discretized strain and electric fields are linked to the discretized displacement vector (u, v, w) and electric potential ϕ by:

$$\left\{ \begin{array}{c} S \\ E \end{array} \right\} = \left\{ \begin{array}{c} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \\ E_x \\ E_y \\ E_z \end{array} \right\} = \left[\begin{array}{cccc} N,x & 0 & 0 & 0 \\ 0 & N,y & 0 & 0 \\ 0 & 0 & N,z & 0 \\ 0 & N,z & N,y & 0 \\ N,z & 0 & N,x & 0 \\ N,y & N,x & 0 & 0 \\ 0 & 0 & 0 & -N,x \\ 0 & 0 & 0 & -N,y \\ 0 & 0 & 0 & -N,z \end{array} \right] \left\{ \begin{array}{c} u \\ v \\ w \\ \phi \end{array} \right\} \tag{3.2}$$

where $N,x\ u$ is a short notation for

$$\sum_i \frac{\partial N_i}{\partial x} u_i$$

and $N_i(x, y, z)$ are the finite element shape functions. Plugging (3.2) in (3.1) leads to the discrete set of equations which are written in the matrix form

$$\left[\begin{array}{cc} M_{qq} & 0 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} \ddot{q}_{mech} \\ \ddot{V} \end{array} \right\} + \left[\begin{array}{cc} K_{qq} & K_{qV} \\ K_{Vq} & K_{VV} \end{array} \right] \left\{ \begin{array}{c} q_{mech} \\ V \end{array} \right\} = \left\{ \begin{array}{c} F_{mech} \\ Q \end{array} \right\} \tag{3.3}$$

where $\{q_{mech}\}$ contains the mechanical degrees of freedom (3 per node related to u, v, w), and $\{V\}$ contains the electrical degrees of freedom (1 per node, the electric potential ϕ). $\{F_{mech}\}$ is the vector of applied external mechanical forces, and $\{Q\}$ is the vector of applied external charges.

3.2 Piezoelectric shell finite elements

Shell strain is defined by the membrane, curvature and transverse shear as well as the electric field components. In the piezoelectric multi-layer shell elements implemented in SDT, it is assumed that in each piezoelectric layer $i = 1...n$, the electric field takes the form $\vec{E} = (0 \quad 0 \quad E_{zi})$. E_{zi} is assumed to be constant over the thickness h_i of the layer and is therefore given by $E_{zi} = -\frac{\Delta\phi_i}{h_i}$ where $\Delta\phi_i$ is the difference of potential between the electrodes at the top and bottom of the piezoelectric layer i. It is also assumed that the piezoelectric principal axes are parallel to the structural orthotropy axes.

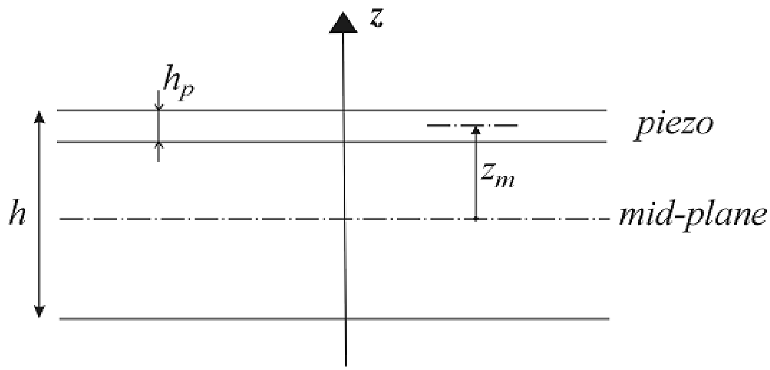


Figure 3.1: Multi-layer shell piezoelectric element

The discretized strain and electric fields of a piezoelectric shell take the form

$$\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \\ \kappa_{xx} \\ \kappa_{yy} \\ 2\kappa_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \\ -E_{z1} \\ ... \\ -E_{zn} \end{Bmatrix} = \begin{bmatrix} N,x & 0 & 0 & 0 & 0 & 0 & ... & 0 \\ 0 & N,y & 0 & 0 & 0 & 0 & ... & 0 \\ N,y & N,x & 0 & 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & -N,x & 0 & ... & 0 \\ 0 & 0 & 0 & N,y & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & N,x & -N,y & 0 & ... & 0 \\ 0 & 0 & N,x & 0 & N & 0 & ... & 0 \\ 0 & 0 & N,y & -N & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{h_1} & ... & 0 \\ ... & ... & ... & ... & ... & 0 & ... & -\frac{1}{h_n} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ ru \\ rv \\ \Delta\phi_1 \\ ... \\ \Delta\phi_n \end{Bmatrix} \tag{3.4}$$

There are thus n additional degrees of freedom $\Delta\phi_i$, n being the number of piezoelectric layers in the laminate shell. The constitutive laws are obtained by using the "piezoelectric plates" hypothesis (2.16) and the definitions of the generalized forces N, M, Q and strains $\varepsilon, \kappa, \gamma$ for shells:

$$\left\{\begin{array}{c} N \\ M \\ Q \\ D_{z1} \\ \dots \\ D_{zn} \end{array}\right\} = \left[\begin{array}{cccccc} A & B & 0 & G_1^T & \dots & G_n^T \\ B & D & 0 & z_{m1}G_1^T & \dots & z_{mn}G_n^T \\ 0 & 0 & F & 0 & \dots & 0 \\ G_1 & z_{m1}G_1 & 0 & -{\varepsilon_1}^S & \dots & 0 \\ \dots & \dots & \dots & 0 & \dots & 0 \\ G_n & z_{mn}G_n & 0 & 0 & \dots & -{\varepsilon_n}^S \end{array}\right] \left\{\begin{array}{c} \epsilon \\ \kappa \\ \gamma \\ -E_{z1} \\ \dots \\ -E_{zn} \end{array}\right\} \tag{3.5}$$

D_{zi} is the electric displacement in piezoelectric layer , z_{mi} is the distance between the midplane of the shell and the midplane of piezoelectric layer i (Figure 3.1), G_i is given by

$$G_i = \left\{\begin{array}{ccc} e_{31}^* & e_{32}^* & 0 \end{array}\right\}_i [R_s]_i \tag{3.6}$$

where * refers to the piezoelectric properties under the piezoelectric plate assumption as detailed in section 2.2 and $[R_s]_i$ are rotation matrices associated to the angle θ of the principal axes $1, 2$ of the piezoelectric layer given by:

$$[R_s] = \left[\begin{array}{ccc} \cos^2\theta & \sin^2\theta & \sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -\sin\theta\cos\theta \\ -2\sin\theta\cos\theta & 2\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{array}\right] \tag{3.7}$$

Plugging (3.4) into (3.1) leads again to:

$$\left[\begin{array}{cc} M_{qq} & 0 \\ 0 & 0 \end{array}\right] \left\{\begin{array}{c} \ddot{q}_{mech} \\ \ddot{V} \end{array}\right\} + \left[\begin{array}{cc} K_{qq} & K_{qV} \\ K_{Vq} & K_{VV} \end{array}\right] \left\{\begin{array}{c} q_{mech} \\ V \end{array}\right\} = \left\{\begin{array}{c} F_{mech} \\ Q \end{array}\right\} \tag{3.8}$$

where $\{q_{mech}\}$ contains the mechanical degrees of freedom (5 per node corresponding to the displacements u, v, w and rotations rx, ry), and $\{V\}$ contains the electrical degrees of freedom. The electrical DOFs are defined at the element level, and there are as many as there are active layers in the laminate. Note that the electrical degree of freedom is the difference of the electric potential between the top and bottom electrodes $\Delta\phi$.

3.3 Full order model

Piezoelectric models are described using both mechanical q_{mech} and electric potential DOF V. As detailed in sections section 3.1 and section 3.2 , one obtains models of the form

$$\left[\begin{array}{cc} Z_{qq}(s) & Z_{qV} \\ Z_{Vq} & Z_{VV} \end{array}\right] \left\{\begin{array}{c} q_{mech} \\ V \end{array}\right\} = \left\{\begin{array}{c} F_{mech} \\ Q \end{array}\right\} \tag{3.9}$$

for both piezoelectric solids and shells, where $Z_{qq}(s)$ is the dynamic (mechanical) stiffness expressed as a function of the Laplace variable s.

For piezoelectric shell elements, electric DOFs correspond to the difference of potential on the electrodes of one layer, while the corresponding load is the charge Q. In SDT, the electric DOFs for shells are unique for a single shell property and are thus giving an implicit definition of electrodes (see `p_piezo Shell`). Note that a common error is to fix all DOF when seeking to fix mechanical DOFs, calls of the form `'x==0 -DOF 1:6'` avoid this error.

For volume elements, each volume node is associated with an electric potential DOF and one defines multiple point constraints to enforce equal potential on nodes linked by a single electrode and sets one of the electrodes to zero potential (see `p_piezo ElectrodeMPC` and for example section 8.1.2 for a tutorial on how to set these constraints). During assembly the constraints are eliminated and the resulting model has electrical DOFs that correspond to potential, or differences of potential (if the other electrode's potential is set to 0) and loads to charge.

Short circuit (SC), charge sensors configurations correspond to cases where the potential is forced to zero (the electrical circuit is shorted). In (3.9), this corresponds to a case where the potential (electrical DOF) is fixed and the charge corresponds to the resulting force associated with this boundary condition.

A **voltage actuator** corresponds to the same problem with $V = V_{In}$ (built in SDT using `fe_load ('DofSet')` entries). The closed circuit charge is associated with the constraint on the enforced voltage and can be computed by extracting the second row of (3.9)

$$\{Q\} = [Z_{qC}]\{q_{mech}\} + [Z_{VV}]\{V_{In}\} \tag{3.10}$$

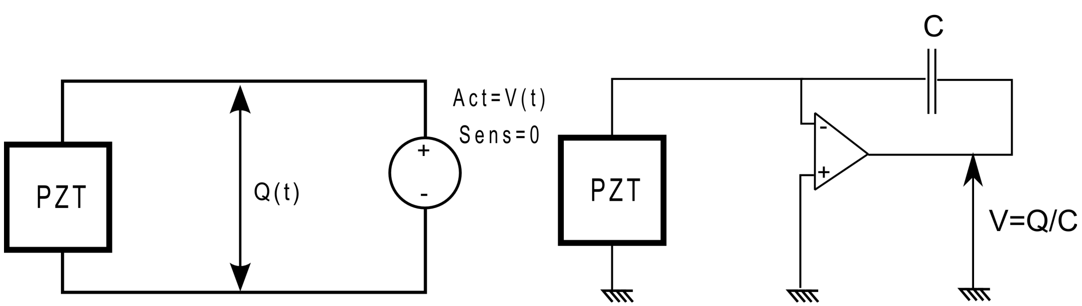


Figure 3.2: Short circuit: voltage actuator, charge sensor

`p_piezo ElectrodeSensQ` provides utilities to build the charge sensors, including sensor combinations.

SC is the only possibly boundary condition to impose in a FEM model where voltage is the unknown. The alternative is to leave the potential free which corresponds to not specifying any boundary condition.

Open circuit (OC), voltage sensor, configurations correspond to cases where the charge remains zero and a potential is created on the electrodes due to mechanical deformations.

A **piezoelectric actuator driven using a charge source** also would correspond to this configuration (but the usual is voltage driving).

The voltage DOFs $\{V\}$ associated to open-circuits are left free in (3.9). Since electrostatics are normally considered, Z_{VV} is actually frequency independent and the voltage DOFs could be condensed exactly

$$\{V\} = [Z_{VV}]^{-1} (Q_{in} - [Z_{Vq}] \{q_{mech}\}) \tag{3.11}$$

This configuration is to be used for a voltage sensor, for example when the piezoelectric transducer is attached directly to the data acquisition card or a voltage amplifier (with very large impedance for sensing). In both cases the impedance is very large leading to a configuration close to an open circuit (OC, infinite impedance). Another example of OC boundary conditions is the use of current (charge) amplifiers for actuation, which is rarely used in practice but possible.

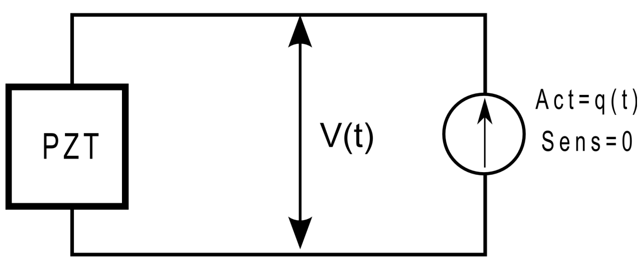


Figure 3.3: Open circuit (voltage sensor, charge actuator)

Since voltage is an explicit DOF, it can be observed using `fe_case('SensDof')` sensor entries. Similarly charge is dual to the voltage, so a charge input would be a simple point load on the active DOF associated to an electrode. Note that specifying a charge distribution does not make sense since you cannot both enforce the equipotential condition and specify a charge distribution that results from this constraint.

It is possible to observe charge in an OC condition, but this is of little interest since this charge will remain at 0.

In summary, when computing modes under voltage actuation, the proper boundary condition is a SC, while for current (charge) actuation, it would be OC. For sensing, a voltage sensor corresponds to OC, while a charge sensor requires SC.

3.4 Using the Electrode stack entry

SDT 6.6 underwent significant revisions to get rid of solver strategies that were specific to piezo applications. The `info,Electrodes` of earlier releases is thus no longer necessary. To avoid disruption of user procedures, you can still use the old format with a `.ver=0` field.

`p_piezo ElectrodeInit` is used to build/verify a data structure describing master electric DOFs associated with electrodes defined in your model. The `info,Electrode` stack entry is a structure with fields

`.data` rows `NodeId IsOpen` gives the electrode nodes and for each one `1` if the circuit is open (voltage free), and `0` if it is closed (voltage enforced or fixed, actuator).

`.ver=1` is used to specify that the more general piezoelectric strategies of SDT >= 6.6 are used. This is the combined with the `p_piezo Electrode2Case` command which builds piezo loads and sensors. For SDT 6.5 strategies, use `.ver=0`.

`.def` `.DOF`, `.lab_in` **only needed** when combining multiple electrodes into a single input. The `.lab_in` is a cell array of strings, you should end the string with `V` so that it shows Q for associated charge sensors.

Each column gives the weighting coefficients associated with each electrode. Thus `def=[1;0;1]` corresponds to a single equal input on electrodes 1 and 3. Note that it does not make sense to combine electrical DOFs that are of mixed nature (actuator/sensor).

The `.DOF` field should contain `NodeId+.21` since the potential corresponds to DOF `.21`.

The `.lab_in` field can be used to provide labels associated with each actuator/sensor defined as a column of def. You should end the label with `V` so that the collocated sensor ends with a `Q` label.

`.cta` `.lab` (optional) can be used to combine electrodes into sensors / actuators. Each row of `.cta` defines a sensor (with matching `.lab`). Each column corresponds to an electrode declared in the `.data` field. You cannot combine open and closed circuit electrodes. It is possible to use both a `.cta` and a `.def` field.

`[model,data]=p_piezo('ElectrodeInit',model);` generates a default value for the electrode stack entry. Combination of actuators and sensors (both charge and voltage) is illustrated in section section 6.3.4 .

3.5 Example 1 : Static response of a piezoelectric patch

3.5.1 Static response of a patch in extension mode

In this very simple example, the electric field and the strains are all constant, so that the electric potential and the displacement field are linear. The example is treated analytically in section section 2.4 . It is therefore possible to obtain an exact solution using a single volumic 8-node finite element (with linear shape functions, the nodal unknowns being the displacements in x,y and z and the electric potential ϕ). Consider a piezoelectric patch whose dimensions and material properties are given in Table 3.1. The material properties correspond to the material *SONOX_P502_iso* in `m_piezo`.

Property	Value
b	10 mm
w	10 mm
h	2 mm
E	54 GPa
ν	0.41
$d_{31} = d_{32}$	-185 $10^{-12} pC/N$ (or m/V)
d_{33}	440 $10^{-12} pC/N$ (or m/V)
$d_{15} = d_{24}$	560 $10^{-12} pC/N$ (or m/V)
$\varepsilon_{33}^T = \varepsilon_{22}^T = \varepsilon_{11}^T$	1850 ε_0
ε_0	8.854 $10^{-12} Fm^{-1}$

Table 3.1: Geometrical and material properties of the piezoelectric patch

We first produce the mesh, associate the material properties and define the electrodes with `d_piezo('TutoPzPatchExt-s1')` . The default material is *SONOX_P502_iso*. The number of elements in the x, y and z directions are given by n_x,n_y and n_z.

The information about the nodes associated to each electrode can be obtained through the following call (Figure 3.4):

```
p_piezo('TabInfo',model)
```

I/O name	NodeId
Top	5.0
Bottom	1.0

Figure 3.4: Tabinfo gives information about nodes associated to electrodes

In d_piezo('TutoPzPatchExt-s2') , the material is changed for example to PIC_255 with the following call, and the full set of mechanical, piezoelectric and permittivity matrices can be obtained in order to check consistency with the datasheet (Figure 3.5).

```
%% Step 2 Define material properties
model.pl=m_piezo('dbval 1 -elas 2 PIC_255');
p_piezo('TabDD',model) % creates the table with full set of matrices
```

MatId 1	cE	elastic stiffness 0 E ...	Pa		
9,36076E10	4,684125E10	4,229144E10	0E00	0E00	0E00
4,684125E10	9,388326E10	4,253161E10	0E00	0E00	0E00
4,229144E10	4,253161E10	7,389928E10	0E00	0E00	0E00
0E00	0E00	0E00	2,103E10	0E00	0E00
0E00	0E00	0E00	0E00	2,103E10	0E00
0E00	0E00	0E00	0E00	0E00	2,353E10
MatId 1	sE	elastic flexibility 0 ...	1/Pa		
1,610047E-11	-5,219771E-12	-6,209894E-12	0E00	0E00	-0E00
-5,219771E-12	1,610047E-11	-6,279182E-12	0E00	0E00	-0E00
-6,209894E-12	-6,279182E-12	2,069965E-11	0E00	0E00	-0E00
0E00	0E00	0E00	4,755112E-11	0E00	-0E00
0E00	0E00	0E00	0E00	4,755112E-11	-0E00
0E00	0E00	0E00	0E00	0E00	4,249894E-11
MatId 1	cD	elastic stiffness 0 El...	Pa		
1,039336E11	5,710994E10	2,464781E10	0E00	0E00	0E00
5,710994E10	1,040949E11	2,498594E10	0E00	0E00	0E00
2,464781E10	2,498594E10	1,040462E11	0E00	0E00	0E00
0E00	0E00	0E00	3,72511E10	0E00	0E00

Figure 3.5: Example subset of table with the full set of mechanical, dielectric and piezoelectric coefficients in the 4 different forms of the constitutive equations

The next step (d_piezo('TutoPzPatchExt-s3')) defines the boundary conditions and load case. We consider here two cases, the first one where the patch is free to expand, and the second one where it is mechanically constrained (all mechanical degrees of freedom are equal to 0).

In `d_piezo('TutoPzPatchExt-s4')` , we can look at the deformed shape, and plot the electric field for both cases.

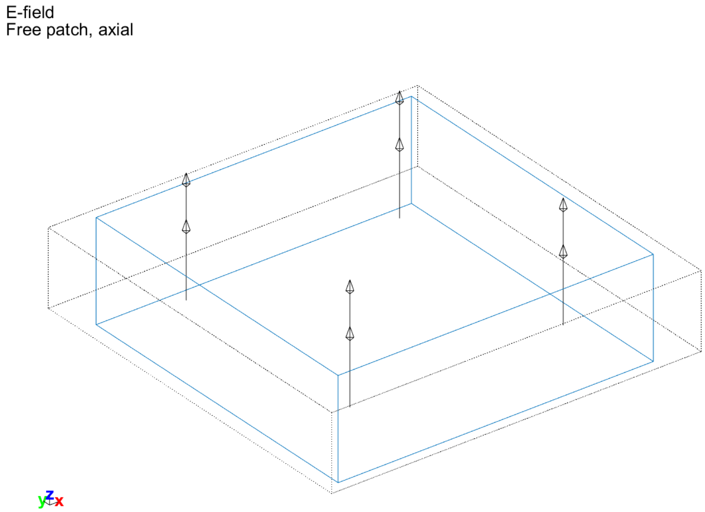


Figure 3.6: Visualization of the electric field and deformed shape for the free patch under unit voltage excitation

For the free patch deformed shape, we compute the mean strains from which d_{31}, d_{32} and d_{33} are deduced. The values are found to be equal to the analytical values used in the model.

```
Relation between mean strain on free structure and d_3i
    {'E3 mean'}    {[500.0000]}    {[500]}    {'E3 analytic'}

    {'Sx'}    {[-1.8000e-10]}    {[-1.8000e-10]}    {'d_31'}
    {'Sy'}    {[-1.8000e-10]}    {[-1.8000e-10]}    {'d_32'}
    {'Sz'}    {[ 4.0000e-10]}    {[ 4.0000e-10]}    {'d_33'}
```

Note that the parameters of the constitutive equations can be recovered using `d_piezo('TutoPzPatchExt-s5')` .

```
%% Step 5 : check constitutive law
% Decompose constitutive law
CC=p_piezo('viewdd -struct',cf); %
```

where the fields of CC are self-explanatory. The parameters which are not directly defined are computed from the equations presented in section 2.1 .
For the constrained patch, we compute the mean stress from which we can compute the e_{31}, e_{32} and e_{33} values which are found to be equal to the analytical values used in the model.

```
Relation between mean stress on pure electric and e_3i
    {'Tx'}    {[-8.3642]}    {[-8.3642]}    {'e_31'}
    {'Ty'}    {[-8.3178]}    {[-8.3178]}    {'e_32'}
    {'Tz'}    {[14.2916]}    {[14.2916]}    {'e_33'}
```

In `d_piezo('TutoPzPatch-s6')` , we can also compute the charge and the charge density (in pC/m^2) accumulated on the electrodes, and compare with the analytical values

```
    {0x0 double                 }    {'Top'       }    {'Bottom'     }
    {'Free patch, axial'        }    {[7.7473e-10]}    {[-7.7473e-10]}
    {'Constrained patch, axial'}     {[3.3876e-10]}    {[-3.3876e-10]}
```

```
Theoretical values of capacitance
    {'CT'}    {[7.7473e-10]}
    {'CS'}    {[3.3876e-10]}
```

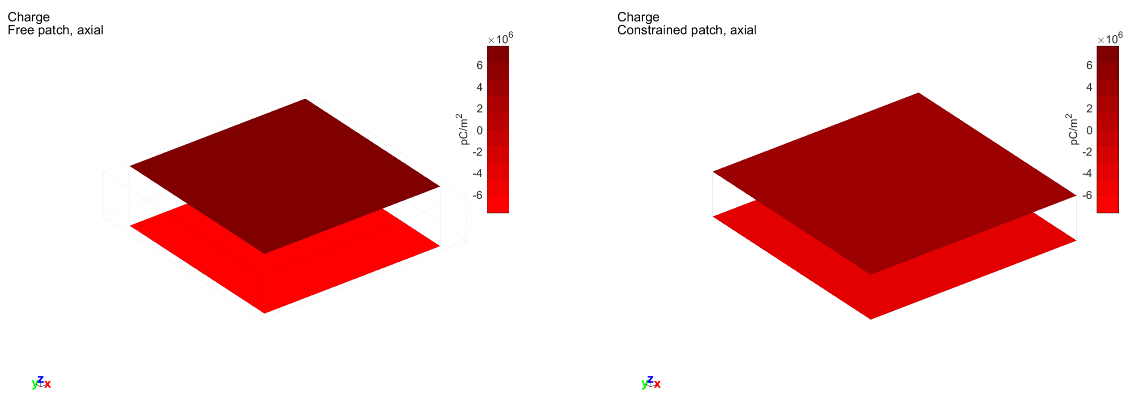


Figure 3.7: Vizualisation of the total charge on the electrodes for the unconstrained and constrained patch under unit voltage excitation

The results clearly show the very large difference of charge density between the two cases (free patch or constrained patch).

For this simple static example, a finer mesh can be used, but it does not lead to more accurate results (this can be done by changing the values in the call to `d_piezo('mesh')` for example:

```
% Build mesh with refinement
model=d_piezo('MeshPatch lx=1e-2 ly=1e-2 h=2e-3 nx=5 ny=5 nz=2');
% Now a model with quadratic elements
model=d_piezo('MeshPatch lx=1e-2 ly=1e-2 h=2e-3 Quad');
```

3.5.2 Static response of a patch in shear mode

As for the patch in extension, as the fields are also uniform (see section 2.4.2), the problem can be modelled with a single 8-node element. In `d_piezo('TutoPzPatchShear-s1')` , the patch is meshed and then the poling is aligned with the $-y$ axis by performing a rotation of 90^o around the x-axis .

In `d_piezo('TutoPzPatchShear-s2')` and `d_piezo('TutoPzPatchShear-s3')` , the response is computed both for the free case and the fully constrained case. The deformed shape for the free case is shown in Figure 3.8 together with the applied electric field:

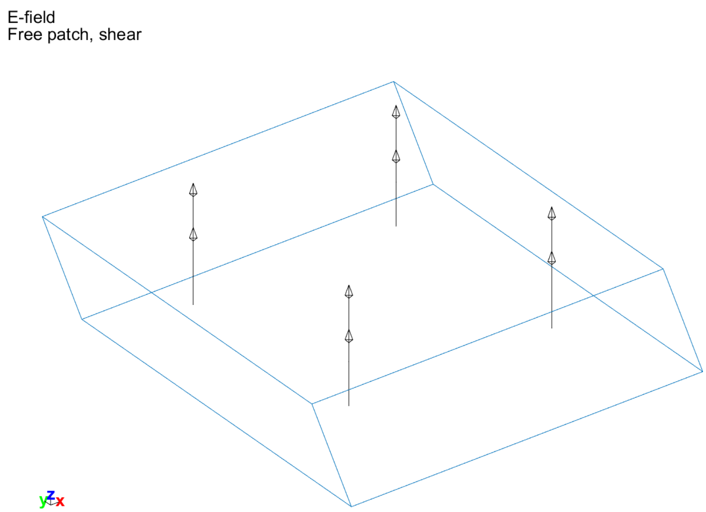


Figure 3.8: Deformed shape of a piezoelectric patch poled in the plane with an electric field applied in the out-of-plane direction

In `d_piezo('TutoPzPatchShear-s4')` , the mean of shear strain and stress is evaluated and compared to the d_{24} and e_{24} piezo coefficients. Note that the mean values are computed in the global

yz axis for which a negative strain corresponds to a positive strain in the local 23 axis.

```
Relation between mean strain on free structure and d_24
    {'E3 mean'}    {[500]}    {[500]}    {'E3 analytic'}
    {'Syz'}    {[5.6000e-10]}    {[5.6000e-10]}    {'d_24'}

    Relation between mean stress on pure electric and e_24
    {'Tyz'}    {[10.7234]}    {[10.7234]}    {'e_24'}
```

Finally in `d_piezo('TutoPzPatchShear-s5')` , the capacitance is evaluated and compared to the theoretical values, showing a perfect agreement, and demonstrating the difference with the extension case for C^S.

```
{0x0 double              }      {'Top'      }    {'Bottom'    }
    {'Free patch, shear'     }      {[8.1900e-10]}    {[-8.1900e-10]}
    {'Constrained patch, shear'}    {[5.1874e-10]}    {[-5.1874e-10]}

Theoretical values of capacitance
    {'CT'}    {[8.1900e-10]}
    {'CS'}    {[5.1874e-10]}
```

3.6 Example 2: Dynamic response of a piezoelectric disk

In this next example, we consider a piezoelectric disk of thickness=$2mm$ and radius=$8mm$ which has electrodes on the top and bottom surfaces. The material used is *PIC181* from *PI Ceramics*. In `d_piezo('TutoPzDiskImpedance-s1')` , the mesh and corresponding electrodes are generated and represented in Figure 3.9.

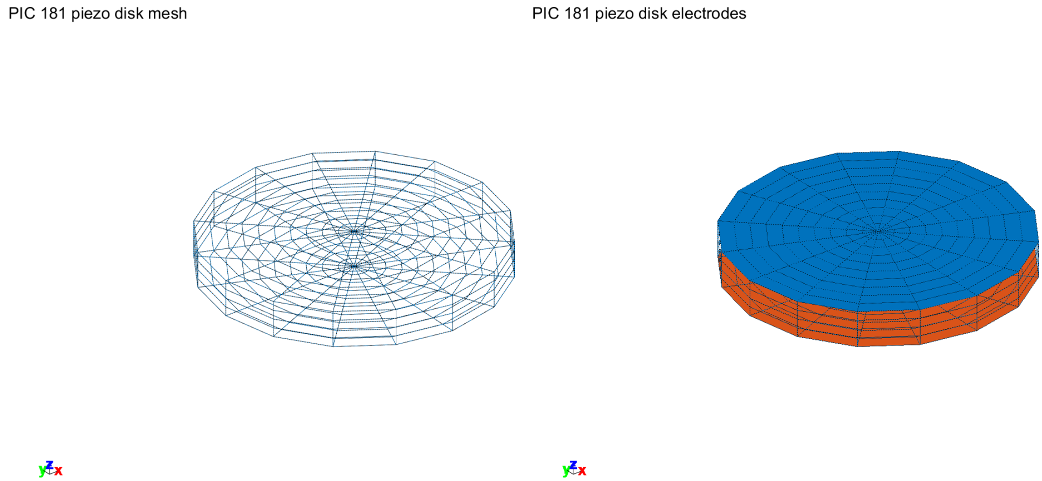


Figure 3.9: Piezo electric disk made of bulk PIC181 material (radius=8mm, thickness=2mm): mesh(left) and electrodes(right)

In `d_piezo('TutoPzDiskImpedance-s2')` , we compute the dynamic response of the disk subjected to an imposed voltage on the top electrode (the bottom electrode is grounded in the model), the frequency range is from 20 to 200 kHz. We represent the voltage distribution on the disk, as well as the electric field at 20 kHz in Figure 3.10.

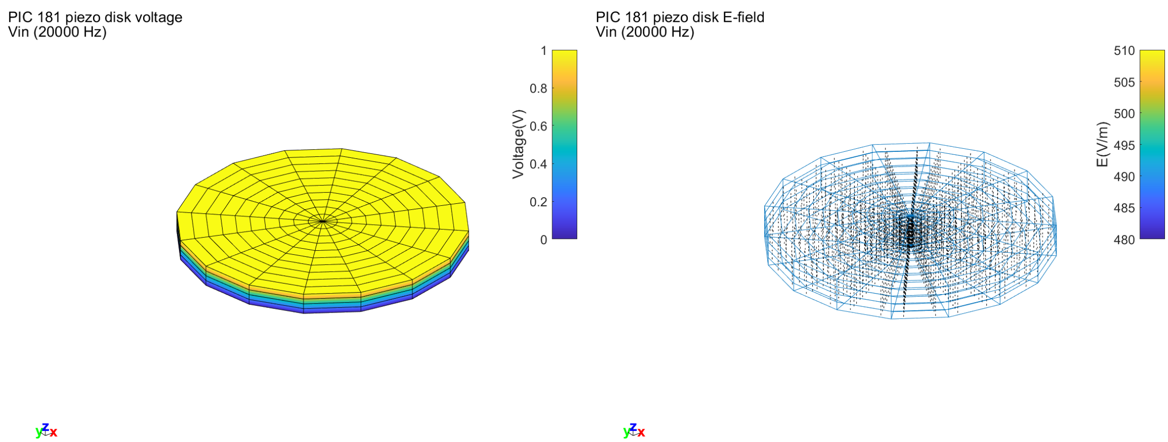


Figure 3.10: Response of a piezoelectric disk made of bulk PIC181 material (radius=8mm, thickness=2mm) at 20kHz, voltage distribution(left) and electric field (right)

In d_piezo('TutoPzDiskImpedance-s3') , a charge sensor is defined on the piezoelectric disk, allowing to compute the charge accumulated at each frequency for an accumulated input voltage. It is represented in Figure 3.11(left).

d_piezo('TutoDiskImpedance-s4') allows to compute the electrical impedance of the disk. It is defined as $Z = V/I = 1/(j\omega)\, V/Q$ and represented in Figure 3.11(right). The anti-resonance in the impedance curve around 140kHz corresponds to the short-circuited radial resonance frequency of the disk, while the resonance around 160kHz corresponds to the open-circuit resonance frequency of the disk. The spacing between these two frequencies can be used to compute the electromechanical coefficient of the disk for the first radial resonance.

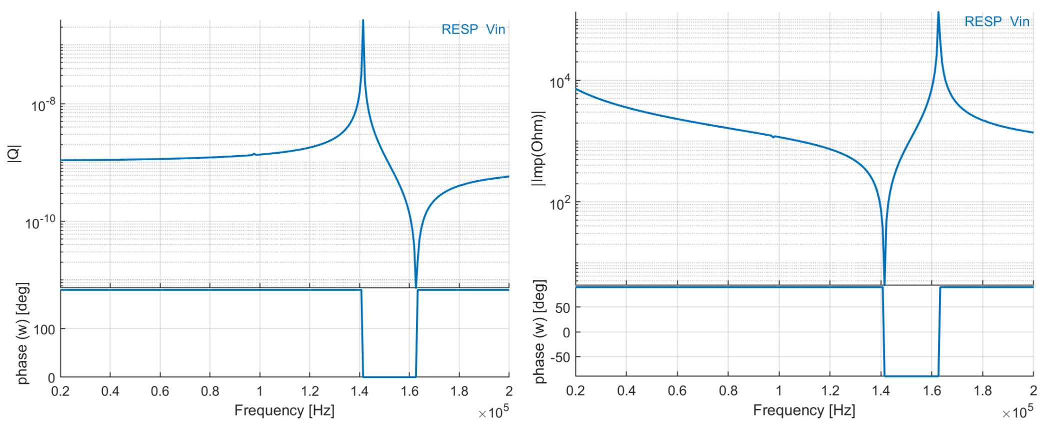


Figure 3.11: Response of a piezoelectric disk made of bulk PIC181 material (radius=$8mm$, thickness=$2mm$) from 20kHz to 200kHz, Q/V(left) and electric impedance (Ohm) (right)

4

Sensors and Actuators definition

Contents

4.1 Input/Output shape matrices

Dynamic loads applied to a discretized mechanical model can be decomposed into a product $\{F\}_q = [b]\{u(t)\}$ where

- the **input shape matrix** $[b]$ is time invariant and characterizes spatial properties of the applied forces

- the vector of inputs $\{u\}$ allows the description of the time/frequency properties.

Similarly it is assumed that the outputs $\{y\}$ (displacements but also strains, stresses, etc.) are linearly related to the model coordinates $\{q\}$ through the sensor **output shape matrix** ($\{y\} = [c]\{q\}$).

Input and output shape matrices are typically generated with `fe_c`, `fe_case` or `fe_load`. For sensors see also `scell`. Understanding what they represent and how they are transformed when model DOFs/states are changed is essential.

Linear mechanical models take the general forms

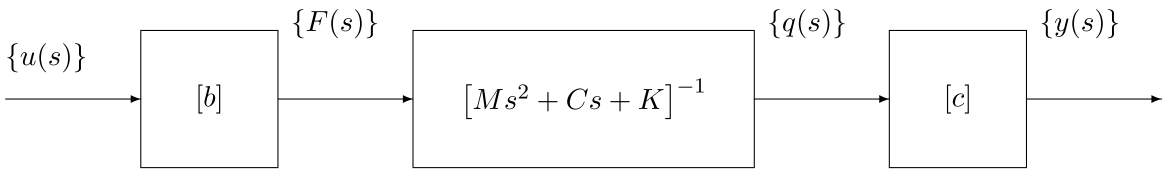
$$\begin{array}{c}\left[Ms^2+Cs+K\right]_{N\times N}\{q(s)\} = [b]_{N\times NA}\{u(s)\}_{NA\times 1}\\ \{y(s)\}_{NS\times 1} = [c]_{NS\times N}\{q(s)\}_{N\times 1}\end{array} \tag{4.1}$$

in the frequency domain, and

$$\begin{array}{c}[M]\{q''\}+[C]\{q'\}+[K]\{q\} = [b]\{u(t)\}\\ \{y(t)\} = [c]\{q(t)\}\end{array} \tag{4.2}$$

in the time domain.

N is the number of degrees of freedom in the model, NA is the number of independent actuators, and NS is the number of sensors.
In the model form (4.1), the first set of equations describes the evolution of $\{q\}$. The components of q are called Degrees Of Freedom (DOFs) by mechanical engineers and states in control theory. The second *observation* equation is rarely considered by mechanical engineers (hopefully the *SDT* may change this). The purpose of this distinction is to lead to the block diagram representation of the structural dynamics



which is very useful for applications in both control and mechanics.

In the simplest case of a point force input at a DOF q_l, the input shape matrix is equal to zero except for DOF l where it takes the value 1

$$[b_l] = \begin{array}{cc} \left[\begin{array}{c} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{array}\right] & \begin{array}{c} \\ \\ \leftarrow l \\ \\ \\ \end{array} \end{array} \tag{4.3}$$

Since $\{q_l\} = [b_l]^T \{q\}$, the transpose of this Boolean input shape matrix is often called a *localization matrix*. Boolean input/output shape matrices are easily generated by `fe_c` Input/output shape matrices become really useful when not Boolean. For applications considered in the *SDT* they are key to

- distributed FEM loads, see `fe_load`.
- the description of piezoelectric loads and sensors, see `p_piezo`.
- the description of sensors that do not directly correspond to DOFs (accelerations in non global directions at positions that do not correspond to finite element nodes).
- model reduction. To allow the changes to the DOFs q while retaining the physical meaning of the I/O relation between $\{u\}$ and $\{y\}$.

4.2 Collocated force-displacement pairs

Collocated force-displacement pairs are commonly used in active vibration control, as they result in an alternance of poles and zeros in the open-loop transfer functions, leading to unconditionally stable control schemes (when actuators and sensors dynamics are neglected).

The `fe_case('DofLoadSensDof')` command provides a generic way to build collocated pairs. On first defines the sensors (related to DOFs in the model) and then creates the corresponding collocated forces. This is illustrated In the `d_piezo('TutoPzBeamCol')` demo below on a 3D model of a U-shaped cantilever beam.

The first step `d_piezo('TutoPzBeamCol-s1')` consists in the meshing, application of boundary conditions and point sensor definition.

The sensor is defined on node 104 in direction x with the following call.

```
% Introduce a point displacement sensor and visualize
% sdtweb sensor#slab % URN based definition of sensors
model = fe_case(model,'SensDOF','Point Sensors',{'104:x'});
```

Figure 4.1 shows the finite element mesh of the U-beam with a sensor added on node 104 in the x-direction. In (`d_piezo('TutoPzBeamCol-s2')`), the load is then defined as being collocated to the sensor, i.e, in the x-direction and on node 104 with the following call:

```
%% Step 2 : Introduce collocated force actuator and compute response
model=fe_case(model,'DofLoad SensDof','Collocated Force','Point Sensors:1')
% 1 for first sensor if there are multiple
```

The static response of the U-beam to this load is computed with `fe_simul` and shown in Figure 4.2.

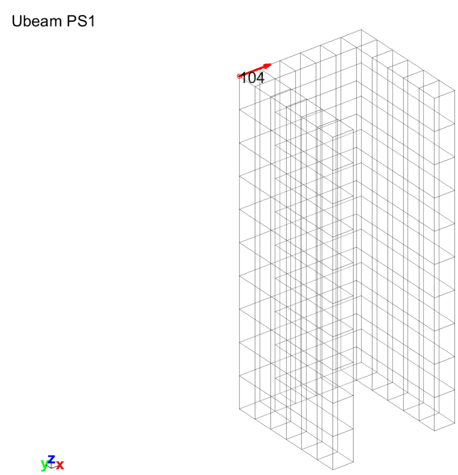


Figure 4.1: U-beam with displacement sensor added on node 104 in direction x

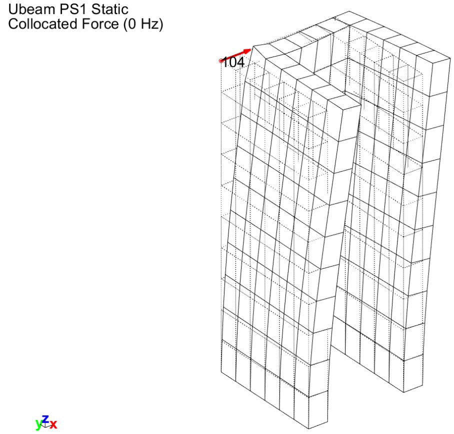


Figure 4.2: Static response of U-beam to load on node 104 in direction x

In `d_piezo('TutoPzBeamCol-s3')` , the collocated transfer function is then computed using `fe_simul` and plotted in the `iiplot` environment (Figure 4.3).

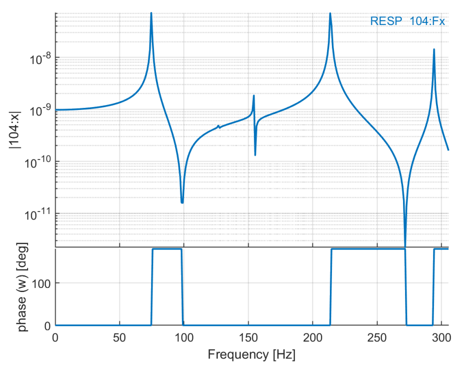


Figure 4.3: Collocated transfer function for a force applied on node 104 in direction x, and a displacement at the same node and in the same direction

Multiple sensors and actuators can also be generated. In `d_piezo('TutoPzBeamCol-s4')` , the single sensor is changed to two sensors (adding a sensor on node 344 in y-direction). Note that if the same name is used in the definition of the sensors, the previous definition is replaced (here 'Point sensors' is the name of the sensing case). The two sensors are shown on Figure 4.4.

```
%% Step 4 : multiple collocated sensors and actuators
% Introduce two sensors and visualize
```

```
model = fe_case(model,'SensDOF','Point sensors',{'104:x';'344:y'});
cf=feplot(model);
```

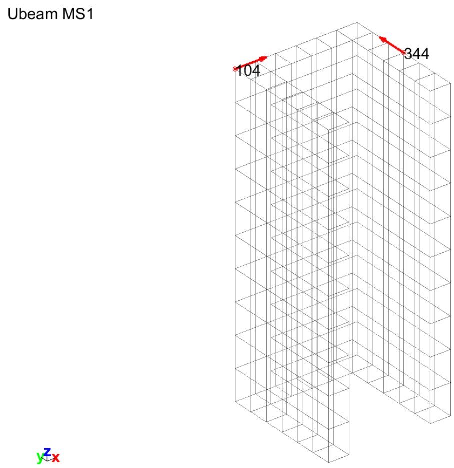


Figure 4.4: U-beam with displacement sensor added on node 104 in direction x and 344 in direction y

Two forces collocated to these sensors are then defined in `d_piezo('TutoPzBeamCol-s5')` , and the static response is computed and shown in Figure 4.5.

The four transfer functions resulting from the definition of two sensors and actuators are then computed and plotted in `d_piezo('TutoPzBeamCol-s6')` . Two of them are collocated resulting in alternance of poles and zeros (Figure 4.6), and the two others are not collocated and do not show this alternance (Figure 4.7). Note that the two transfer functions are identical due to the reciprocity in linear systems.

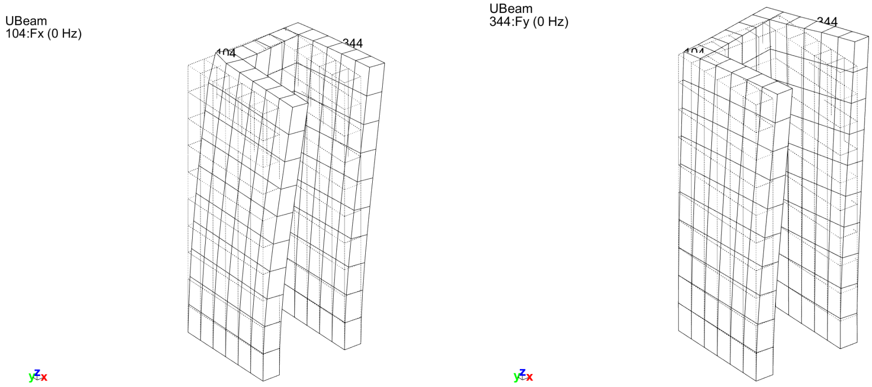


Figure 4.5: Static response of the U-beam to two loads (104-x, 344-y)

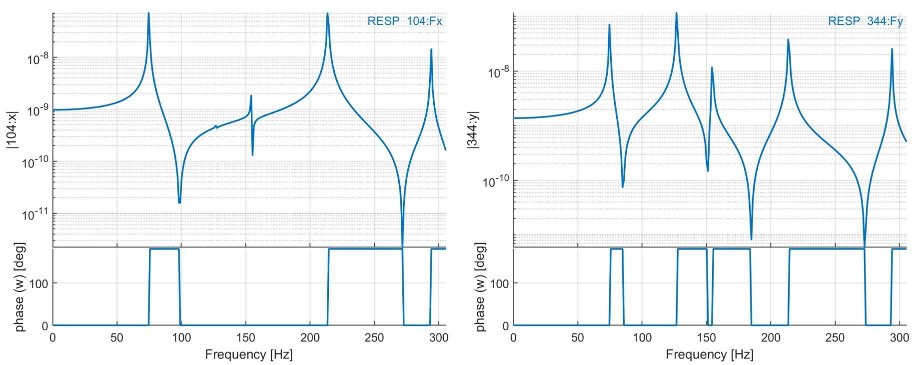


Figure 4.6: Collocated transfer functions for two loads (104-x, 344-y)

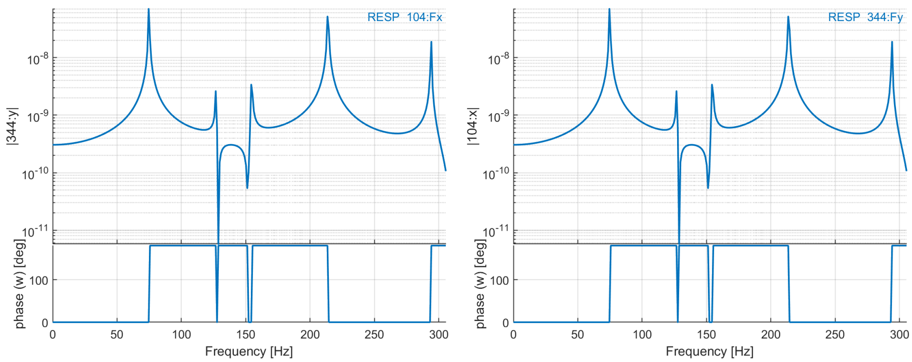


Figure 4.7: The 4 transfer functions for two loads (104-x, 344-y), and two collocated point sensors

4.3 Non-collocated force-displacement pairs and combinations

The example below illustrates the general definition of actuators and sensors which are not collocated.

In `d_piezo('TutoPzBeamNCol-s1')`), the mesh is created and two point sensors are defined as 104-x and 207-y (Figure 4.8)

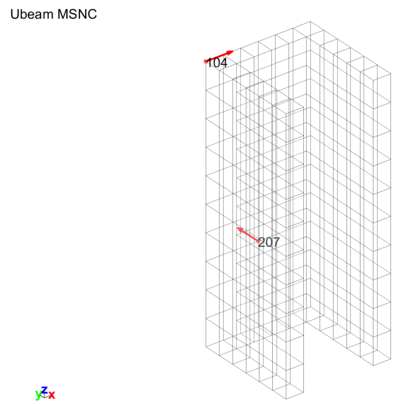


Figure 4.8: Point sensors at node 104 in direction x and 207 in direction y

Then in `d_piezo('TutoPzBeamNCol-s2')` , two actuators are defined by combination of three forces (207-x, 241-x and 207-y). The first actuator consists in two forces on nodes 207 and 241 in opposite direction along x acting together, and the third force is on 207-y.

```
%% Step 2 : Define point actuators
% relative force between DOFs 207x and 241x and one point loads at DOFs 207y
data  = struct('DOF',[207.01;241.01;207.02],'def',[1 0;-1 0;0 1]);
model=fe_case(model,'DofLoad','Actuators',data); %
```

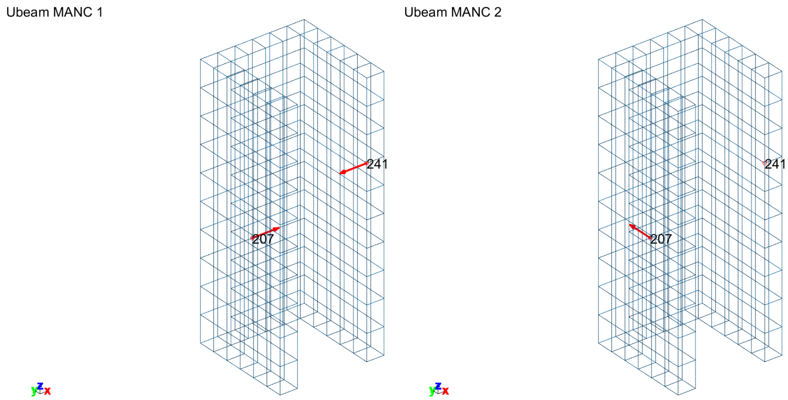


Figure 4.9: Load combinations for Ubeam

In `d_piezo('TutoPzBeamNCol-s3')` and (`d_piezo('TutoPzBeamNCol-s4')`), the static and dynamic responses to these loads are computed and represented in Figure 4.10 and Figure 4.11.

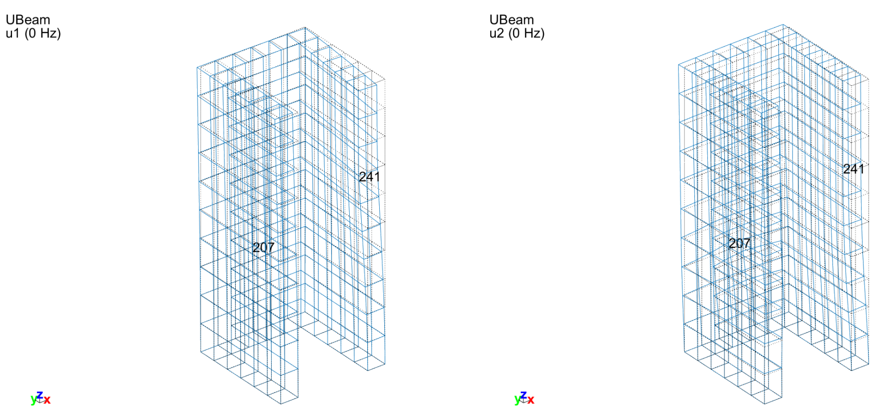


Figure 4.10: Static responses of Ubeam to load combinations

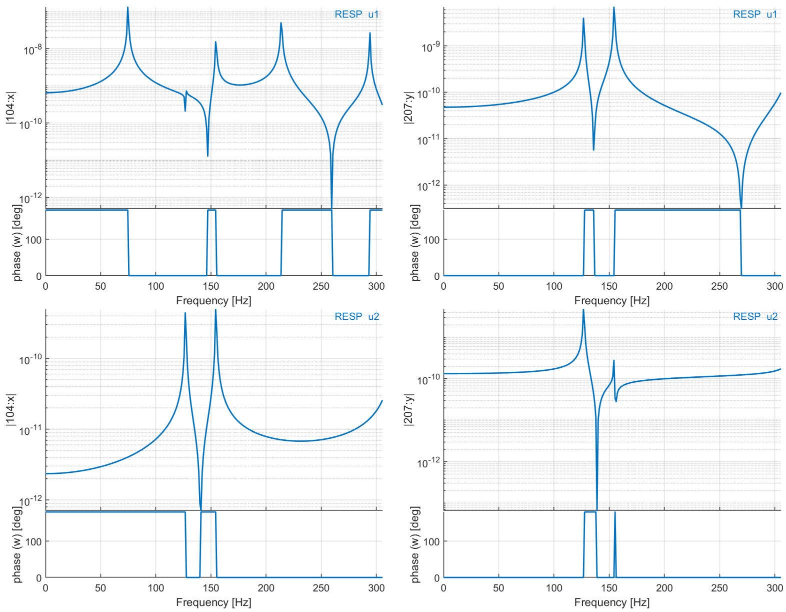


Figure 4.11: Dynamic response of Ubeam to load combinations u1 and u2, sensors on node 104 in direction x, and 207 in direction y

4.4 Other types of actuators

Other types of loads, such as surface or volumic loads are handled by the `fe_load` command (`sdtweb('fe_load')` for more details) in SDT. The case of imposed displacement is handled with `fe_case('DofSet')` calls. The following script illustrates the use of volume and surface loads on the U-beam.

In `d_piezo('TutoPzBeamSurfVol-s1')` , the first load is defined by 'data' and is a constant volumic load in the y-direction. The second load, defined by 'data2' is in the z-direction and its magnitude depends both on the x and z position in the U-beam ($f(x, z) = x * (z - 1)^3$). The two loads are represented in Figures 4.12 and 4.13 using arrows and color codes (for the variable force only).

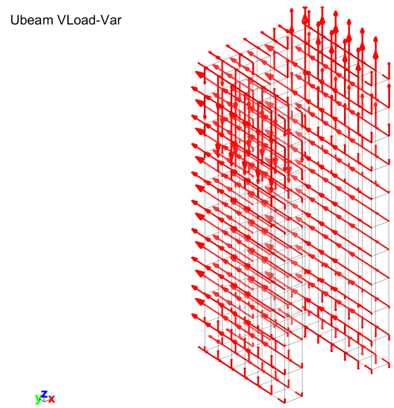


Figure 4.12: Constant volumic force applied in the y-direction

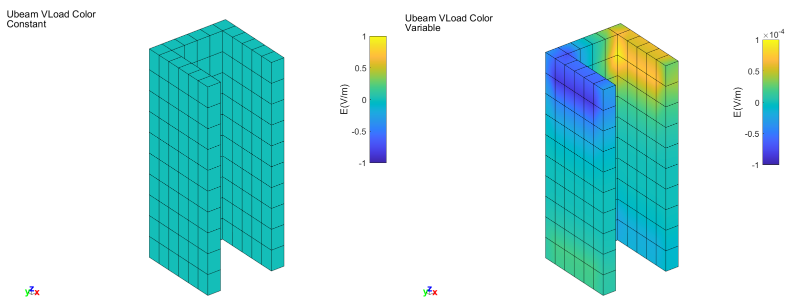


Figure 4.13: Variable volumic force applied in the z-direction represented by arrows (left) and color code proportionnal to amplitude in z-direction (right)

```
%% Step 1 Apply  a volumic load and represent
model = femesh('testubeam');
data=struct('sel','groupall','dir',[0 32 0]);
data2=struct('sel','groupall','dir',{{0,0,'(z-1).^3.*x'}});
model=fe_case(model,'FVol','Constant',data, ...
                     'FVol','Variable',data2);

% Visualize loads
cf=feplot(model); iimouse('resetview');

% Make mesh transparent :
fecom('showfialpha') %
```

```
% Visualize Load
fecom proviewon

% Improve figure
sdth.urn('Tab(Cases,Constant){deflen,.5,arProp,"linewidth,2"}',cf)
fecom curtabcases 'Constant' % Shows the case 'Constant'
```

In `d_piezo('TutoPzBeamSurfVol-s2')` , the surface load is defined using selectors to define the area of the surface where the load is applied. In this case, the load is on the surface corresponding to $x = -0.5$ and $z > 1.25$. The resulting surface load is represented in Figure 4.14.

```
%% Step 2 :  Apply a surface load case in a model using selectors
data=struct('sel','x==-.5', ...
            'eltsel','withnode {z>1.25}','def',1,'DOF',.19);
model=fe_case(model,'Fsurf','Surface load',data); cf=feplot(model);
```

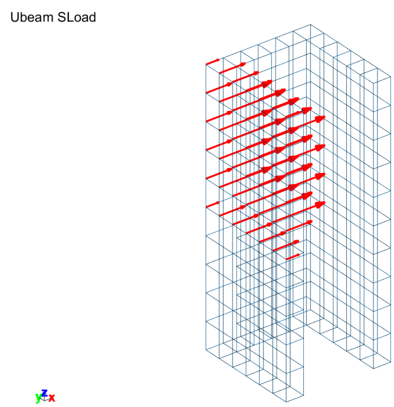


Figure 4.14: Surface load applied on a specific surface of the U-beam

In `d_piezo('TutoPzBeamSurfVol-s3')` , the same is done using node lists.

```
%% Step 3 : Applying a surfacing load case in a model using node lists
data=struct('eltsel','withnode {z>1.25}','def',1,'DOF',.19);
NodeList=feutil('findnode x==-.5',model);
data.sel={'','NodeId','==',NodeList};
model=fe_case(model,'Fsurf','Surface load 2',data); cf=feplot(model);
```

And lastly in d_piezo('TutoPzBeamSurfVol-s4') , the use of sets to define surface loads (Figure 4.15) is illustrated.

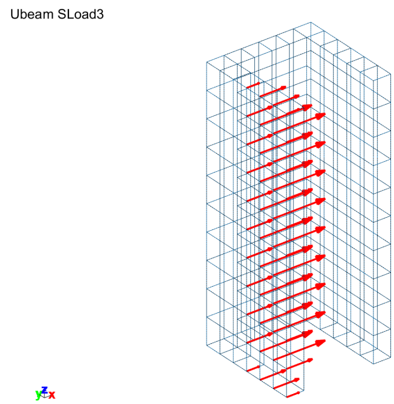


Figure 4.15: Surface load applied on a specific surface of the U-beam

```
%% Step 4 : Applying a surfacing load case in a model using sets

% Define a face set
[eltid,model.Elt]=feutil('eltidfix;',model);
i1=feutil('findelt withnode {x==-.5 & y<0}',model);i1=eltid(i1);
i1(:,2)=2; % fourth face is loaded
data=struct('ID',1,'data',i1,'type','FaceId');
model=stack_set(model,'set','Face 1',data);

% define a load on face 1
data=struct('set','Face 1','def',1,'DOF',.19);
model=fe_case(model,'Fsurf','Surface load 3',data); cf=feplot(model);
```

Another type of load is the case when degrees of freedom are imposed (imposed displacement, velocity, acceleration, or electric potential in the case of piezoelectric actuators). The script below illustrates the case where the U-beam has an imposed displacement on the cantilever side, in the y-direction (Figure 4.16).

In d_piezo('TutoPzBeamUimp-s1') , the mesh is created and the boundary conditions are applied. One has to be careful not to block the degrees of freedom in the direction of the imposed motion (y in this example):

```
%% Step 1 Meshing and BC
model = femesh('test ubeam');
% BC : Impose displacement -  Fix all other dofs for Base
model=fe_case(model,'FixDof','Clamping','z==0 -DOF 1 3');
```

In d_piezo('TutoPzBeamUimp-s2') , the displacement at the base is imposed using a 'DofSet' entry.

```
%% Step 2 Apply base displacement in y-direction
% find node z==0
nd=feutil('find node z==0',model);
data.DOF=nd+.02; data.def=ones(length(nd),1);
model=fe_case(model,'DofSet','Uimp',data);
```

Then in In d_piezo('TutoPzBeamUimp-s3') , a point sensor is introduced on node 104 in direction y (Figure 4.16).

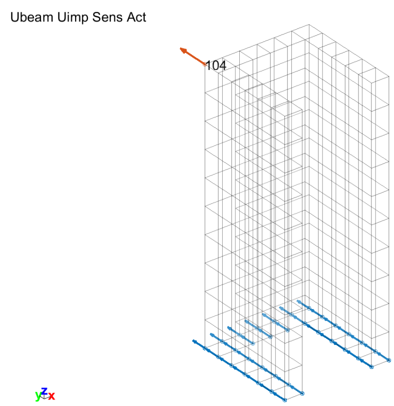


Figure 4.16: Imposed displacement of the base of the U-beam in the y-direction, and sensor at node 104 in the y-direction

Finally in (d_piezo('TutoPzBeamUimp-s4')), the transfer function is computed using fe_simul and plotted in the iiplot environment (Figure 4.17).

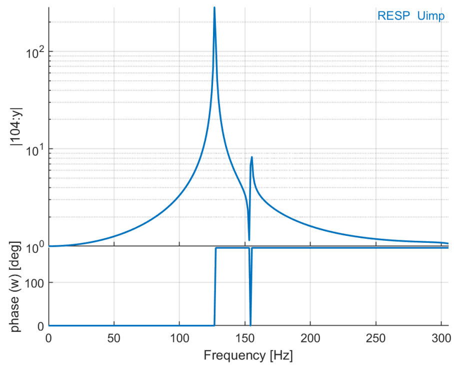


Figure 4.17: Transfer function for an imposed displacement of the base of the U-beam in the y-direction, and a displacement at node 104 in the y-direction

As expected, the static response is unitary, as the U-beam does not deform when a uniform unitary displacement of the base is imposed, so that node 104 moves in the same direction and with the same magnitude as the base.

Note that it is also possible to impose an acceleration when building state-space models, which will be detailed in section 6.3 .

4.5 Other types of sensors

For point sensors, velocity and acceleration sensors can also be defined. The short script below introduces a collocated sensor actuator pair on node 104 in direction x for the U-beam. The sensor is either a displacement, velocity or acceleration sensor, and the resulting collocated FRFs are plotted in Figure 4.18.

`d_piezo('TutoPzBeamDispVelAcc-s1')` allows to build the mesh, apply the boundary conditions, and compute the mode shapes.

Then in `d_piezo('TutoPzBeamDispVelAcc-s2')` , a displacement, velocity and acceleration sensor is defined at not 104 in direction x, and a collocated force is introduced.

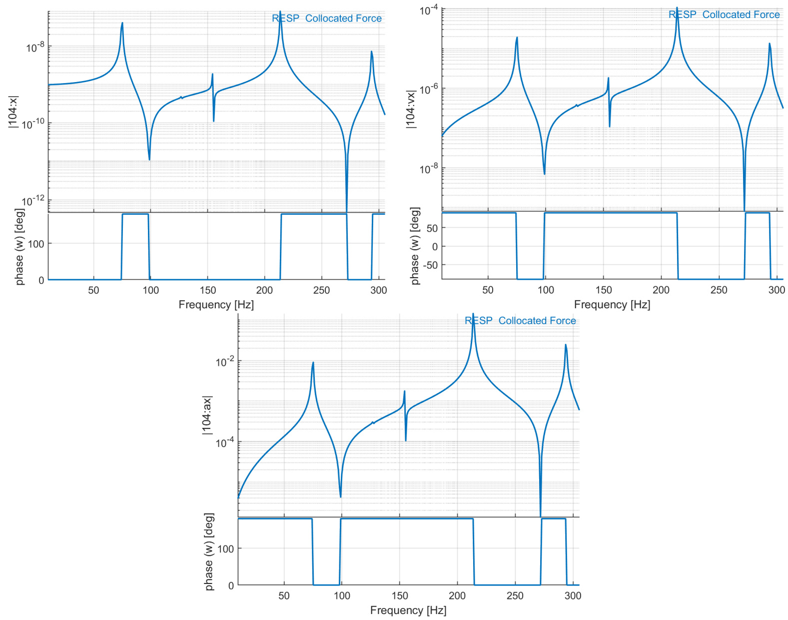


Figure 4.18: Collocated transfer functions with displacement, velocity and acceleration at the same point

```
%% Step 2 : Introduce a point displ/vel/acc sensor and collocated force
model = fe_case(model,'SensDOF','Sensors',{'104:x';'104:vx';'104:ax'});
model=fe_case(model,'DofLoad SensDof','Collocated Force','Sensors:1');
% 1 for first sensor if there are multiple
```

In `d_piezo('TutoPzBeamDispVelAcc-s3')`), the transfer function is computed for the three sensors and represented in Figure 4.18.
Other types of sensors are also defined in SDT using `fe_case('Sens...')` calls. They are detailed in the general SDT manual and are of the following type:

- general sensor (low level) `base/sstruct#general`
- relative displacement sensor `base/sstruct#rel`
- translation sensor `base/sstruct#trans`
- triaxial sensors `base/sstruct#triax`
- for laser vibrometers, defining line of sight of laser `base/sstruct#laser`

- `resultant` resultant force sensor.
- strain or stress sensor `base/sstruct#strain`

Refer to SDT general documentation for examples of definition of such transducers. SDT also provides advanced tools to define sensors which are not at the location of nodes in the finite element models. For more details, see the documentation of the `base/fe_sens` function.

In addition, piezoelectric sensors are discussed in the next section (section 4.6).

4.6 Piezoelectric sensors and actuators

4.6.1 General theory

Essentially, when using piezoelectric materials in finite element models of structures, additional electrical degrees of freedom are added to the model, generally in the form of voltage degrees of freedom. The general form of the equations of motion was already given (see (3.8)) and is recalled below:

$$\begin{bmatrix} M_{qq} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} q'' \\ V'' \end{Bmatrix} + \begin{bmatrix} C_{qq} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} q' \\ V' \end{Bmatrix} + \begin{bmatrix} K_{qq} & K_{qV} \\ K_{Vq} & K_{VV} \end{bmatrix} \begin{Bmatrix} q \\ V \end{Bmatrix} = \begin{Bmatrix} F_{mech} \\ Q \end{Bmatrix} \tag{4.4}$$

The equations of motion couple the electrical DOFs to the mechanical ones through the matrices K_{Vq} and K_{qV}.

When piezoelectric materials are used for sensing and actuating, they have associated electrodes which enforce equipotentiality on the surfaces. In the finite element model, this is taken into account by adding equality constraints between all the electrical DOFs associated to an electrode. One single electrical DOF is then linked to each electrode, and these DOFs are used for actuation and sensing.

Note that in the shell formulation of SDT, the electrical DOFs are not associated to nodes, but rather to elements, as detailed in section 3.2 . Each element can have several layers among which some can be defined with piezoelectric material. Each piezoelectric layer of the shell element has then one associated DOF which corresponds to the difference of potential between the top and the bottom electrode of that layer. This is equivalent to considering that the bottom electrode of each layer is set to 0 volts and that the electrical DOF represents the voltage on the top electrode.

For 3D elements, each node has an additional associated electrical DOF, so that it is necessary to define boundary conditions in terms of voltage on the two electrodes. In SDT, electrical DOFs are identified with .21.

As explained in section 3.3 , actuation can be of two types: an electrical charge Q can be imposed on one of the electrodes, which is equivalent to defining a mechanical force, as this term is on the right-hand-side of the equations. This case should be treated as the addition of point loads using `fe_case('DofLoad')` calls. It is however more common to use piezoelectric actuators in a voltage-driven mode, which corresponds to imposing the voltage, and can be treated using a `fe_case('DofSet')`.

Assume that the voltage DOFs can be divided in two parts :

$$\{V\} = \left\{ \begin{array}{c} V_0 \\ V_{in} \end{array} \right\} \tag{4.5}$$

where V_{in} corresponds to the voltages imposed on the actuators in the model, and the other voltage DOFs are left free. We assume that the constraints related to the electrodes have already been taken into account to reduce the different matrices. The first line of the equations of motion becomes (assuming K_{VV} is a diagonal matrix, which is usually the case as it corresponds to the matrix of the individual capacitances) :

$$\left[\begin{array}{cc} M_{qq} & 0 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} q'' \\ V_0'' \end{array} \right\} + \left[\begin{array}{cc} C_{qq} & 0 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} q' \\ V_0' \end{array} \right\} + \left[\begin{array}{cc} K_{qq} & K_{qV_0} \\ K_{V_0q} & K_{V_0V_0} \end{array} \right] \left\{ \begin{array}{c} q \\ V_0 \end{array} \right\} = \left\{ \begin{array}{c} F_{mech} - K_{qV_{in}} V_{in} \\ Q_0 \end{array} \right\} \tag{4.6}$$

The fact that there is no coupling term in the mass and damping matrices leads to an equivalent problem where an imposed voltage corresponds to a mechanical load $-\left[K_{qV_{in}}\right]\{V_{in}\}$.

Piezoelectric sensors are also of two types. Voltage sensing corresponds to measuring the voltage DOF directly, and can thus be defined by a simple `fe_case('SensDOF')` call. If a charge amplifier is used to measure the signal generated by the piezoelectrc sensor, this is equivalent to measuring a reaction force at the associated electrical DOF. In order to do that, one first needs to fix the associated electrical DOF using a `fe_case('FixDOF')` (if set to 0) or a `fe_case('DofSet')` (if used both as voltage actuator and charge sensor as in the case of self-sensing and piezo-shunt applications) call, and then define the charge sensor with a `p_piezo ElectrodeSensQ` call. Assume again that the electrical DOFs are divided in two parts

$$\{V\} = \left\{ \begin{array}{c} V_0 \\ V_q \end{array} \right\} \tag{4.7}$$

where V_q are fixed electrical DOFs where the charge needs to be measured. The charge on these electrodes is then given by:

$$Q = \left[K_{V_q q}\right]\{q\} + \left[K_{V_q V_q}\right]\{V_q\} \tag{4.8}$$

The definition of actuators and sensors of these different types are illustrated below for a plate modelled with shell elements, and an example of an accelerometer modelled with 3D elements.

4.6.2 Aluminum plate with 4 PZT patches (Shell model)

The first example deals with an aluminum plate with 4 piezoceramic patches. The geometry and the material properties are given in Figure 4.19 and Table 4.1

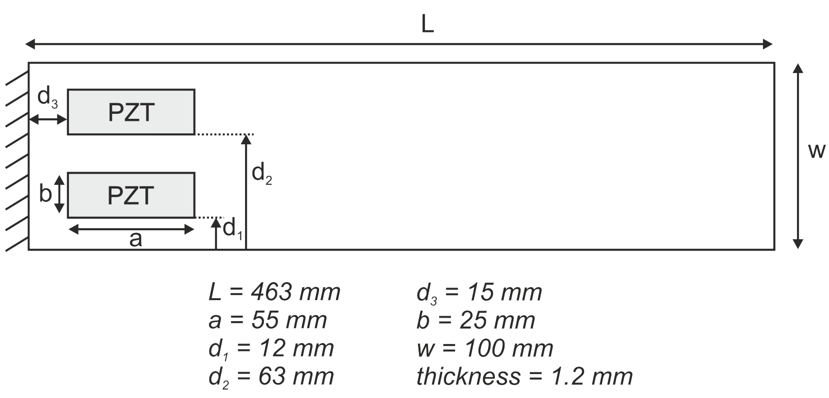


Figure 4.19: Geometric details of the aluminum plate with 4 piezoceramic patches

Property	Value
Aluminum plate	
E	$72GPa$
ν	0.3
ρ	$2700kg/m^3$
Piezoceramic patches	
E	$54.05GPa$
ν	0.41
ρ	$7740kg/m^3$
thickness	$0.25mm$
d_{31}	-185 $10^{-12}pC/N$ (or m/V)
d_{32}	-185 $10^{-12}pC/N$ (or m/V)
ε_{33}^T	1850 ε_0
ε_0	8.854 $10^{-12}Fm^{-1}$

Table 4.1: Material properties of the plate and the piezoceramic patches

It corresponds to a cantilevered plate with 4 piezoelectric patches modeled using the `p_piezo Shell` formulation.

The first step `d_piezo('TutoPzPlate4pzt-s1')` consists in the creation of the model, the definition of the boundary conditions, and the definition of the default damping coefficient. The different meshing procedures are detailed further in section 5 . The resulting mesh is shown in Figure 4.20

One can have access to the piezoelectric material properties (Figure 4.21) and the list of nodes associated to each pair of electrodes (Figure 4.22) using `p_piezo Tab` calls. Here nodes 1055 to 1058 are associated to the four pairs of electrodes defined in the model. The corresponding degree of freedom is the difference of potential between the electrodes in each pair corresponding to a specific piezoelectric layer. In this models, layers 1 and 3 are piezoelectric in groups 2 and 3 (the internal layer correspond to supporting aluminum plate). Therefore only .21 (electrical) DOF is associated to nodes 1055-1058.

```
p_piezo('TabDD',model);      % List piezo constitutive laws
```

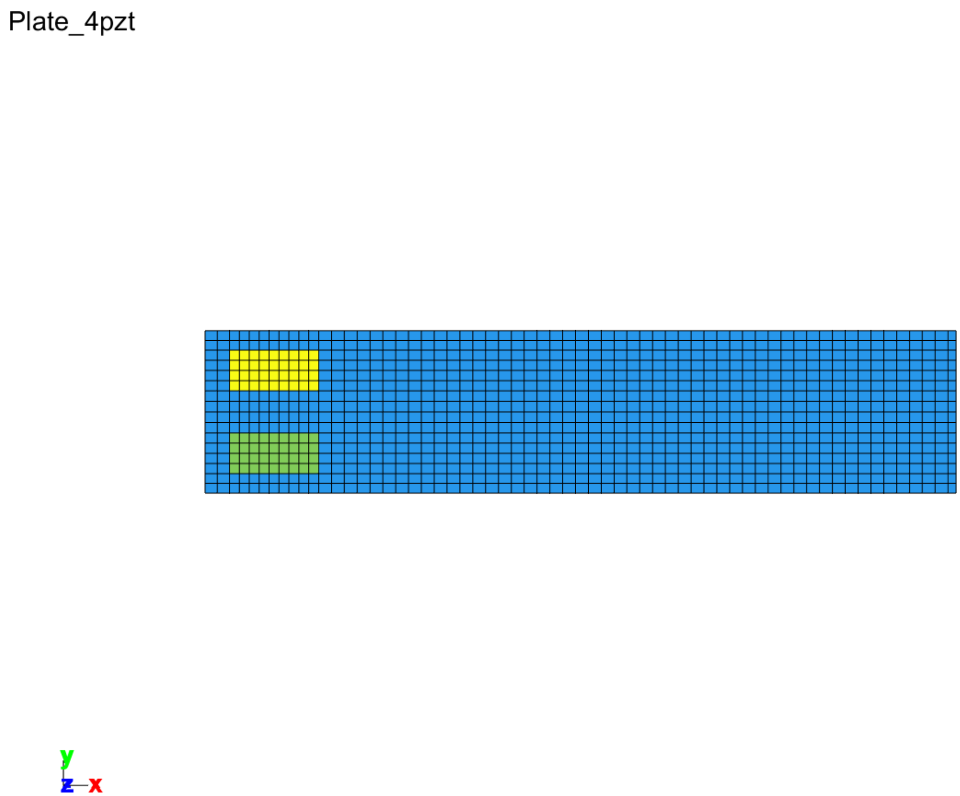


Figure 4.20: Mesh of the composite plate. The different colors represent the different groups

MatId 101	cE	elastic stiffness 0 E field	GPa	
6.491164803461954E7	2.661377569419401E7	0.0	0.0	-0.0
2.661377569419401E7	6.491164803461955E7	0.0	0.0	-0.0
0.0	0.0	1.9148936170212768E7	0.0	-0.0
0.0	0.0	0.0	1.9148936170212768E7	-0.0
-0.0	-0.0	-0.0	-0.0	1.9148936170212768E7
MatId 101	sE	elastic flexibility 0 Elec ...	1/GPa	
1.8518518518518518E-8	-7.592592592592592E-9	0.0	-0.0	-0.0
-7.592592592592592E-9	1.8518518518518518E-8	0.0	0.0	-0.0
0.0	0.0	5.222222222222222E-8	0.0	-0.0
-0.0	0.0	0.0	5.222222222222222E-8	0.0
-0.0	-0.0	-0.0	0.0	5.222222222222222E-8
MatId 101	cD	elastic stiffness 0 Elec d...	GPa	
9.325568599816561E7	5.495781365774009E7	0.0	0.0	0.0
5.495781365774009E7	9.325568599816564E7	0.0	0.0	0.0
0.0	0.0	1.9148936170212768E7	0.0	0.0
0.0	0.0	0.0	1.9148936170212768E7	0.0
0.0	0.0	0.0	0.0	1.9148936170212768E7

Figure 4.21: Example subset of table with the full set of mechanical, dielectric and piezoelectric coefficients in the 4 different forms of the constitutive equations

```
r1=p_piezo('TabInfo',model); % List piezo related properties
```

ProId-Layer	ElectricNodeId
ProId104-I3	1055.0
ProId104-I1	1056.0
ProId109-I3	1057.0
ProId109-I1	1058.0
I/O name	NodeId
ProId104-I3	1055.0
ProId104-I1	1056.0
ProId109-I3	1057.0
ProId109-I1	1058.0

Figure 4.22: Information about electrical master nodes related to each piezoelectric layer. Here layers 1 and 3 for two zones with ProID 104 and 109 define the 4 piezoelectric patches

The next step `d_piezo('TutoPzPlate4pzt-s2')` consists in the definition of the actuators and sensors in the model. Here, we consider one actuator on node 1055 (layer 3 of group 1), the four piezoelectric patches are used as charge sensors, and the tip displacement of the cantilever beam is measured at the right-upward corner of the beam (corresponding to node 1054 here). Note that in order for Q-S1, Q-S2 and Q-S3 to measure resultant charge, the corresponding electrical difference of potential needs to be set to zero. If this is not done, then the charge sensors will measure a charge close to zero (round-off errors) as there is no charge when the difference of potential across the electrodes is free. For Q-Act, the electrical DOF is already fixed due to the fact that the patch is used as a voltage actuator.

```
%% Step 2 - Define actuators and sensors and visualize
nd=feutil('find node x==463 & y==100',model);
model=fe_case(model,'SensDof','Tip',{[num2str(nd) ':z']}); % Displ sensor
i1=p_piezo('TabInfo',model);i1=i1.Electrodes(:,1);
model=fe_case(model,'DofSet','V-Act',struct('def',1,'DOF',i1(1)+.21, ...%Act
    'Elt',feutil('selelt proid 104',model))); % Elt defined for display
model=p_piezo(sprintf('ElectrodeSensQ  %i Q-Act',i1(1)),model); % Charge sensors
model=p_piezo(sprintf('ElectrodeSensQ  %i Q-S1',i1(2)),model);
model=p_piezo(sprintf('ElectrodeSensQ  %i Q-S2',i1(3)),model);
model=p_piezo(sprintf('ElectrodeSensQ  %i Q-S3',i1(4)),model);
% Fix ElectrodeSensQ dofs to measure resultant (charge)
model=fe_case(model,'FixDof','SC*S1-S3',i1(2:end)+.21);
cf=feplot(model); fecom('view3')
cf.mdl.name='Plate_4pzt'; d_piezo('SetStyle',cf); feplot(cf);
```

We can now visualize the voltage actuator and the tip displacement sensor. Note that to visualize the piezoelectric actuator, it is necessary to associated a group of elements when defining it (see `fe_case('DofSet')` command above).

```
fecom('showfialpha')
fecom('proviewon')

% Arrow length and thickness
sdth.urn('Tab(Cases,Tip){Proview,on,deflen,20}',cf)
sdth.urn('Tab(Cases,Tip){arProp,"linewidth,2"}',cf)
fecom('curtabCase',{'Tip';'V-Act'})
```

Figure 4.23 shows the visualization of the tip sensor and voltage actuator.

We can also visualize the charge sensors (Figure 4.24). Note however that in the case of plates, it is not possible to distinguish on the visualization if the patch is on the top or the bottom of the plate, so that Q-S2 and Q-S3 would give the same visualisation.

```
fecom('curtabCase',{'Q-S1';'Q-S2'})
cf.mdl.name='Plate_4pzt_QS1-2';d_piezo('SetStyle',cf); feplot(cf);
```

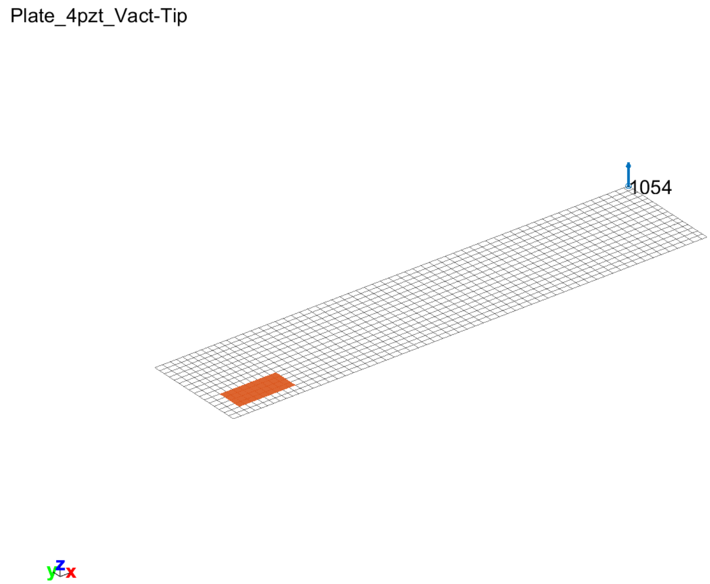


Figure 4.23: Visualization of the tip sensor and voltage actuator

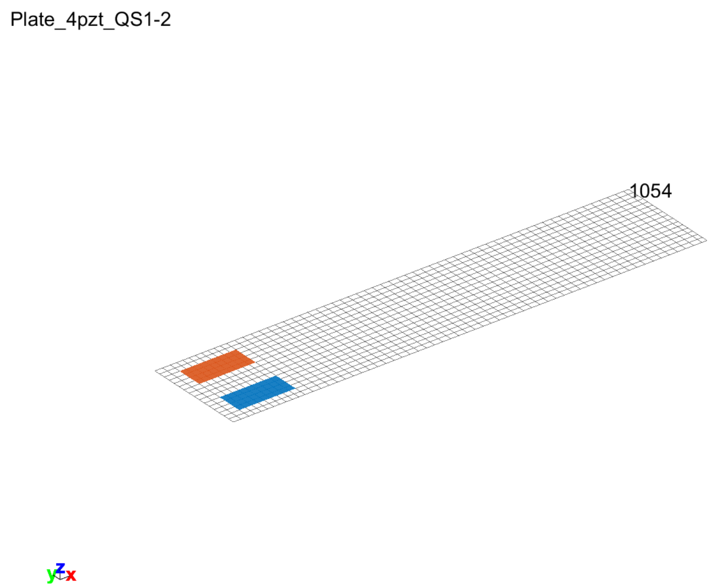


Figure 4.24: Visualization of the 2 charge sensors Q-S1 and Q-S2

In order to check the effect of the actuator, we compute the static response using the full model and represent the deformed shape (Figure 4.25) in `d_piezo('TutoPzPlate4pzt-s3')` .

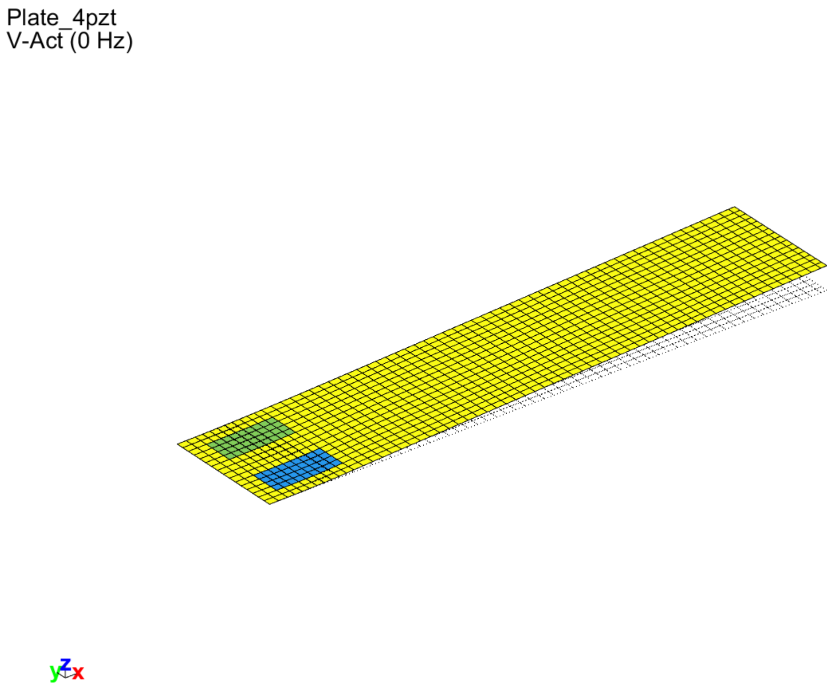


Figure 4.25: Deformed shape under voltage actuation on one of the bottom piezoelectric patches

We also compute the transfer function between the actuator and the four charge sensors, as well as the tip sensor using the full model.

Figure 4.26 shows the different transfer functions from the voltage actuator to the tip displacement sensor (left) and to all charge sensors (right). The figure shows that Q-SAct shows an alternance of poles and zeros, but the smallest distance between the poles and zeros. This configuration corresponds to transfer functions used for shunting applications. For QS1 to QS3, the pole-zero alternance is lost due to the fact that the sensor and the actuator are not strictly collocated. The pole-zero distance is also very different, although if we are looking at the structure with the beam theory, the three FRFs should be identical. This demonstrates clearly the need for shell models, and the impact of the location of the sensor on the pole-zero pattern, and pole-zero distances.

4.6.3 Piezoelectric shaker with an accelerometer mounted on top (3D model)

The second example deals with an accelerometer (sensor) mounted on a piezoelectric shaker (actua-

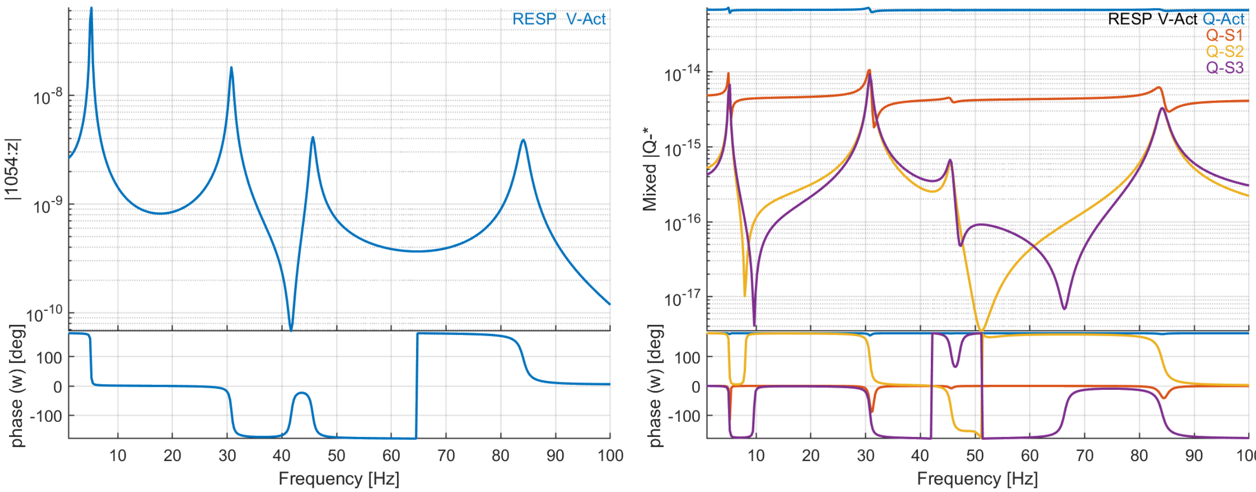


Figure 4.26: Open-loop transfer function between V-Act and tip displacement (left), the 4 charge sensors (right)

tor). The piezoelectric shaker consists of two steel cylindrical parts with a piezoelectric disc inserted in between. The base of the shaker is fixed and the piezoelectric element is used as an actuator: imposing a voltage difference between the electrodes results in the motion of the top surface of the shaker to which the accelerometer is attached (Figure 4.27).

The piezoelectric properties for the sensing element in the piezoelectric shaker are given in Table 4.3. The actuating element has the same properties as the sensing element and is poled through the thickness, as the actuator.

Part	Material	E (GPa)	ρ (kg/m^3)	ν
Wear plate	Al_2O_3	400	3965	0.22
Sensing element	Piezo	54	7740	0.44
Proof mass	Steel	210	7800	0.3

Table 4.2: Mechanical properties of the wear plate, sensing element and proof mass

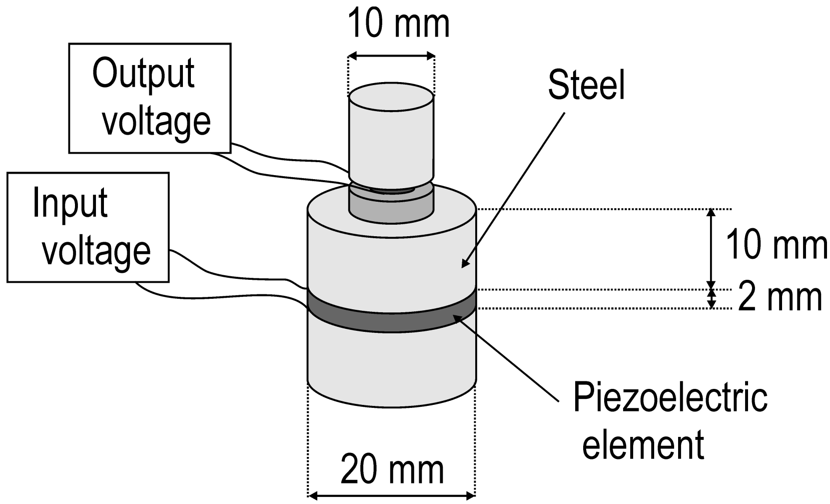


Figure 4.27: Piezoelectric accelerometer attached to a piezoelectric shaker for sensor calibration

Property	Value
$d_{31} = d_{32}$	-185 $10^{-12} pC/N$ (or m/V)
d_{33}	440 $10^{-12} pC/N$ (or m/V)
$d_{15} = d_{24}$	560 $10^{-12} pC/N$ (or m/V)
$\varepsilon^T_{33} = \varepsilon^T_{22} = \varepsilon^T_{11}$	1850 ε_0
ε_0	8.854 $10^{-12} Fm^{-1}$

Table 4.3: Piezoelectric properties of the sensing element

The mesh is generated with `d_piezo('TutoPzAccShaker-s1')` and represented in Figure 4.28. In the meshing script, both a voltage and a charge sensor are defined for the piezoelectric disk in the accelerometer for the top electrode, and the the bottom electrode potential is set to zero.

In `d_piezo('TutoPzAccShaker-s2')` , a voltage actuator is then added to the shaker top electrode, while the bottom electrode potential is set to 0. The shaker is mechanically fixed at the bottom.

```
%% Step 2 - Define actuators and sensors
  % -input "In" says it will be used as a voltage actuator
model=p_piezo('ElectrodeMPC Top Actuator -input "Vin-Shaker"',model,'z==-0.01');
  % -ground generates a v=0 FixDof case entry
model=p_piezo('ElectrodeMPC Bottom Actuator -ground',model,'z==-0.012');
% Voltage sensor will be used - remove charge sensor
```

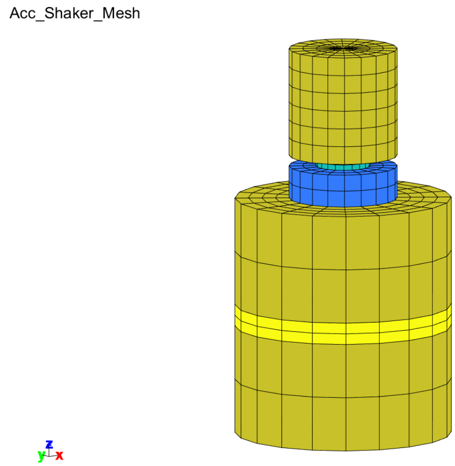


Figure 4.28: Mesh of the piezoelectric accelerometer attached to a piezoelectric shaker

```
model=fe_case(model,'remove','Q-Top sensor');
```

The different electrodes in the model can be visualized (Figure 4.29) and each actuator/sensor can be visualized separately (Figure 4.30)

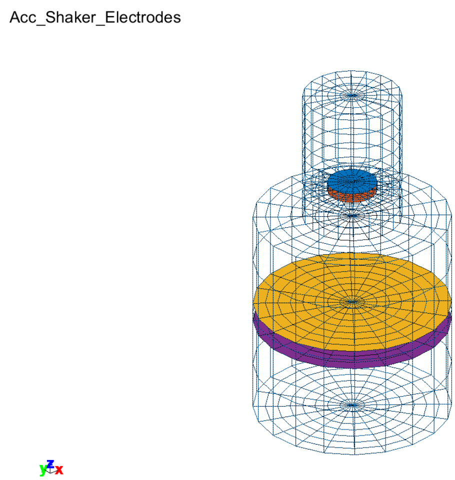


Figure 4.29: Top and bottom electrodes for shaker voltage actuator and accelerometer voltage sensor

```
% Visualize Vin electrode
```

```
fecom curtabcases Vin-Shaker %

% Visualize VSens electrode
fecom curtabcases 'V-Top sensor'
```

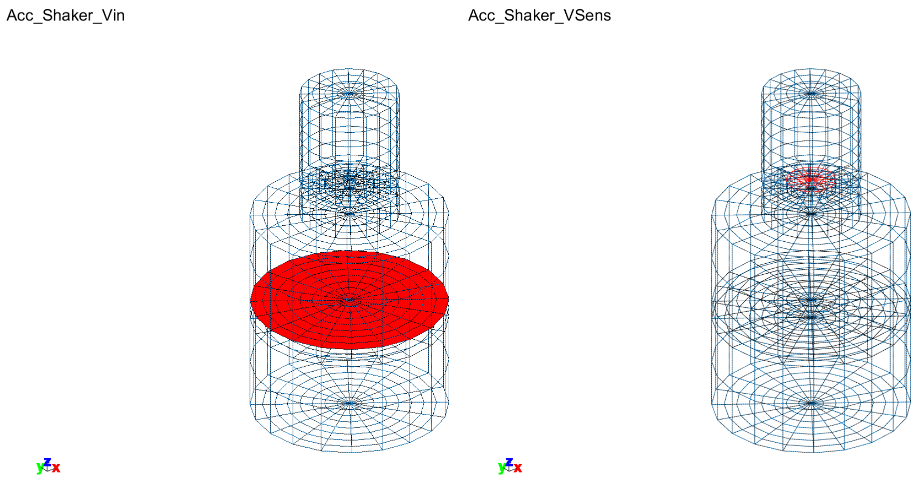


Figure 4.30: Shaker input voltage(left) and accelerometer output voltage(right)

We can have an overview of the electrodes and associated electrical DOFs (Figure 4.31) :

```
r1=p_piezo('TabInfo',model); % List piezo related properties
```

I/O name	NodeId
Top sensor	17.0
Bottom sensor	11.0
Top Actuator	856.0
Bottom Actuator	854.0

Figure 4.31: Electrodes and associated nodes/DOFs

In `d_piezo('TutoPzAccShaker-s3')` , the FRF between the voltage applied on the piezoelectric shaker and accelerometer response (voltage mode) is computed and represented in Figure 4.32, together with the transfer function for the displacement of the base of the accelerometer.

```
% End of script
```

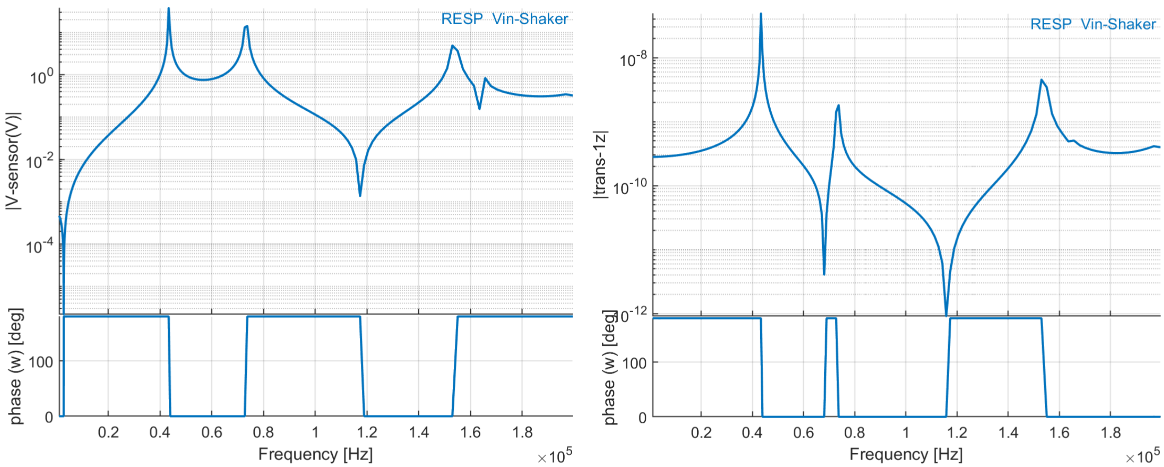


Figure 4.32: FRF between the voltage applied on the piezoelectric shaker and accelerometer response (voltage mode), and the displacement of the base of the accelerometer

5

Methods for meshing plates with piezoelectric patches

Contents

This section explains how to mesh plates with piezoelectric patches of different shapes. In all cases, multi-layer plate elements are used to represent the host plate and the additional piezoelectric layers. The first (basic) strategy consists in assigning additional layers with given piezoelectric properties to regions of the mesh. The second approach uses a script to remesh locally the plate and add piezoelectric layers according to a predefined set of patch shapes and piezoelectric properties.

5.1 Manual meshing

We consider again the example treated in section 4.6.2 , represented in Figure 4.19. The material properties of the aluminum plate and the piezoceramic patches are given in Table 4.1. The thickness of the plate is $1.2mm$, and the piezoelectric material corresponds to the *SONOX_P502_iso* material in `m_piezo Database`.

In the first step `d_piezo('TutoPzMeshingBasics-s1')` , the mesh is built with simple SDT tools:

```
% See full example in d_piezo('ScriptTutoPzMeshingBasics')
d_piezo('DefineStyles');

%% Step 1 - Mesh the plate
 model=struct('Node',[1 0 0 0 0 0 0],'Elt',[]);
 model=feutil('addelt',model,'mass1',1);
 % Note that the extrusion values are chosen to include the patch edges
 dx=[linspace(0,15,3) linspace(15,15+55,10) linspace(15+55,463-5,50) 463];
 model=feutil('extrude 0 1 0 0',model,.001*unique(dx));
 dy=[linspace(0,12,3) linspace(12,12+25,5) linspace(12+25,63,5) ...
    linspace(63,63+25,5) linspace(63+25,100,3)];
 model=feutil('extrude 0 0 1 0',model,.001*unique(dy));
```

Note that the meshing is such that the patch edges corresponds to limits of elements in the mesh. It is then possible to divided the mesh in different groups, related to the two areas where the patches are added, and the host structure with no piezoelectric properties. Then a different ProID is set for the regions with piezoelectric patches.

This is done on the second step `d_piezo('TutoPzMeshingBasics-s2')`

```
 %% Step 2 - Set patch areas and set different properties
 model.Elt=feutil('divide group 1 withnode{x>.015 & x<.07 & y>.013 & y<.037 }',model);
 model.Elt=feutil('divide group 2 withnode{x>.015 & x<.07 & y>.064 & y<.088 }',model);
 model.Elt=feutil('set group 1 proId3',model);
 model.Elt=feutil('set group 2 proId4',model);
```

Visualizing the mesh and using `fe_com('colordatapro')` (see Figure 5.1) allows to check that different properties have been assigned to the two regions where piezoelectric patches need to be added.

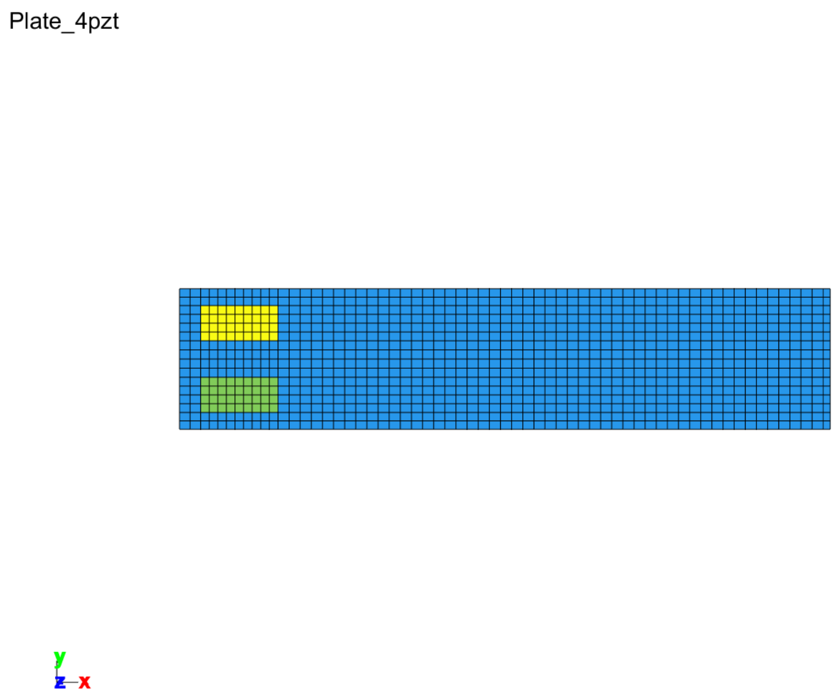


Figure 5.1: Mesh of the composite plate. The different colours represent the different groups

The next step `d_piezo('TutoPzMeshingBasics-s3')` is to define the material properties for the host structure and the patches. Here, *Sonox_P502_iso* is used:

```
%% Step 3 - Material Properties
model.pl=m_elastic('dbval 1 Aluminum');
model.pl=m_piezo(model.pl,'dbval 3 -elas2 SONOX_P502_iso');
% To avoid warning due to the use of simplified piezo properties.
model=p_piezo('DToSimple',model);
```

Piezoelectric material properties are divided in two parts. The first one (ProId 3 here) contains the piezoelectric data, and is linked to elastic properties with ProID 2 in this example through the `-elas2` command. If the piezoelectric properties do not exist in the database, it is always possible to introduce them by hand with the following commands:

```
  model.pl=m_elastic('dbval 1 Aluminum');
  d=zeros(1,18); d([11 13])=560e-12; d([3 6])=-185e-12; d(9)=440e-12;
  eps=zeros(1,9); eps([1 5 9])= 8.854e-12*1850;
%                                      elasid |dij coeff  | dielectric coeffs
  model.pl=[ model.pl zeros(1,24);
            3 fe_mat('m_piezo','SI',2) 2       d          eps;
            2 fe_mat('m_elastic','SI',1) 54e9 0.41 7740 zeros(1,25)];
```

The piezoelectric and dielectric coefficients are stored in a vector of 18 (3x6 matrix) and 9 (3x3 matrix) values respectively. For more information see `m_piezo 2:General 3D piezo`.

In `d_piezo('TutoPzMeshingBasics-s4')` , now that the material properties have been defined, it is necessary to create laminates with three layers, the top and bottom layers being made of the piezoelectric material (ProID 3), of thickness $0.25mm$ and the middle layer being the host structure made of aluminum with thickness $1.2mm$.

```
%% Step 4 - Laminate properties and piezo electrodes
model.il=p_shell('dbval 1 laminate 1 1.2e-3 0', ...
  'dbval 2 laminate 3 2.5e-4 0 1 1.2e-3 0 3 2.5e-4 0');
```

It is also necessary to assign electrodes to each patch, and an electrical DOF id. This single DOF represents the difference of voltage between the top and bottom electrode and is the same for all elements in the piezoelectric patch, as the electrodes impose equipotentiality. Here for example, DOF 1682.21 is assigned to the first layer of group 1, and DOF 1683.21 to the third layer of the same group.

```
%%% Piezo electrodes
%%%                                          NdNb LayerID   NdNb  LayerID
model.il=p_piezo(model.il,'dbval 3 shell 2 1682    1   0    1683  3 0');
model.il=p_piezo(model.il,'dbval 4 shell 2 1684    1   0    1685  3 0');
```

We can now check that the patches are implemented properly by computing the static response to an applied voltage to the patch in group 1 (bottom). The layers are numbered from bottom to top, so it corresponds to DOF 1682.21. Note that in order to find the top and bottom of an element for more complex (curved) meshes, it is possible to use the following command to show the orientation of the normal of the elements (Figure 5.2):

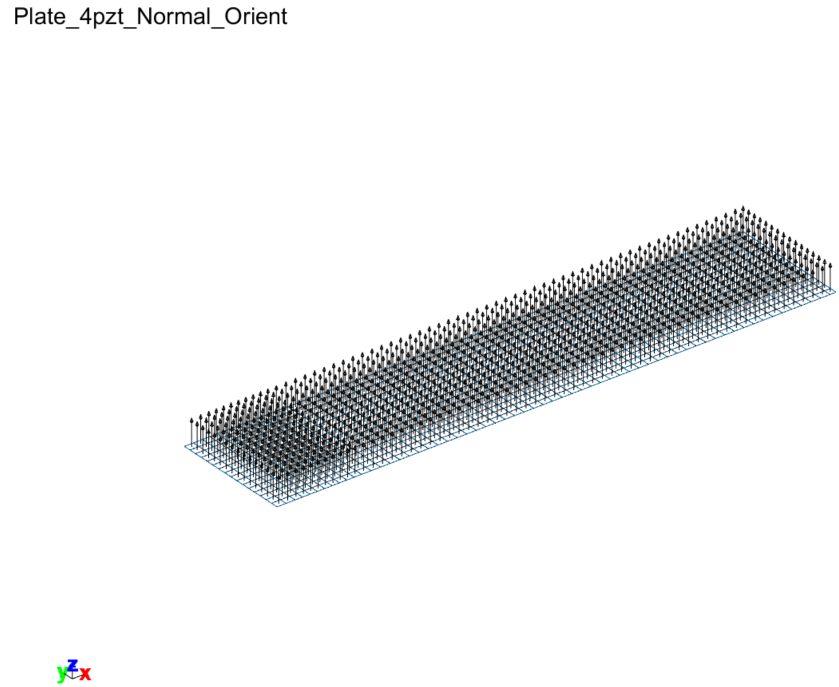


Figure 5.2: Mesh of the plate with arrows showing the orientation of the normals to the elements

```
cf=feplot(model); fecom('showmap'); fecom('view3');
% scale properly
fecom('scalecoeff 1e-10'); fecom('showmap')
```

The electrical DOF corresponds to the difference of electrical potential between the top and bottom electrodes, so if one applies a difference of potential of $1V$, it results in a negative electric field which is in the opposite direction of the poling direction (always in the positive z (normal) direction for a plate element). Because the d_{31} and d_{32} are negative coefficient, the resulting strain is positive (negative electric field multiplied by negative constant). A positive strain at the bottom of the plate should result in an upward motion of the tip of the beam, which is verified in `d_piezo('TutoPzMeshingBasics-s5')` (Figure 5.3).

The tip-displacement is:

```
d =  2.5244e-06
```

The script above also defines three charge sensors on DOFs 1683.21, 1684.21 and 1685.21. If the electrodes are not short-circuited, it would result in a zero-charge measured, so the electrical DOFs are set to zero for these three patches so that a resultant charge can be measured.

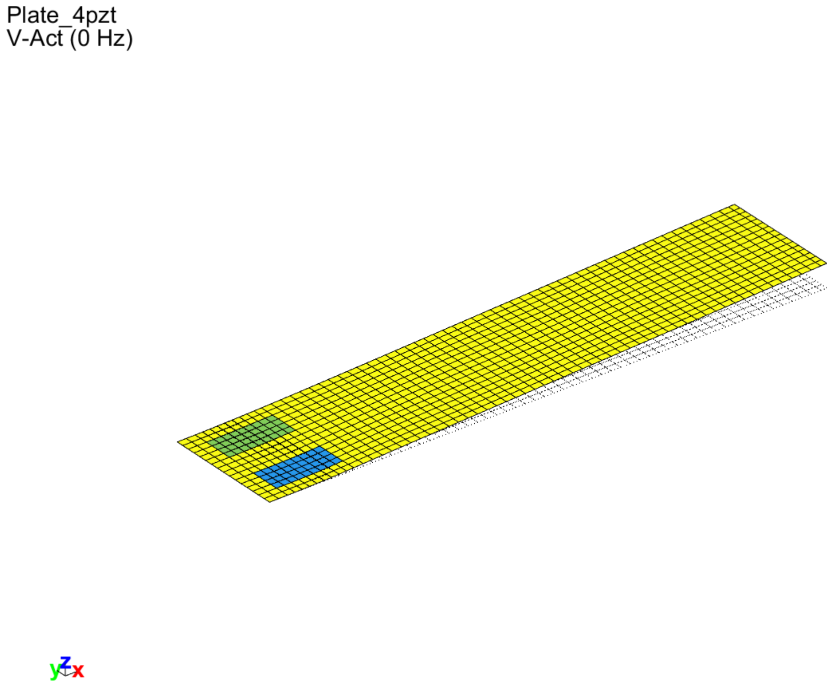


Figure 5.3: Static response to a unit voltage application on one of the bottom piezoelectric patches

To implement a piezoelectric patch on a single side of the plate, it is important to treat properly the offset, as the layer sequence is not symmetric with respect to the neutral plane of the plate. This is done by using the z_0 parameter when defining the laminates. z_0 corresponds to the distance from the mid-plane to the bottom of the plate (see `sdtweb('p_shell')`). It is illustrated in the last step `d_piezo('TutoPzMeshingBasics-s6')` which is the same as the previous one but with piezoelectric patches only on the bottom of the plate, in which case the offset should be $z_0 = -0.6mm - 0.25mm = -0.85mm$ where $0.6mm$ is half the thickness of the support plate and $0.25mm$ is the thickness of the piezo.

```
%%  Laminate properties and piezo electrodes
% This is where an offset must be specified
%(else z0=-7.25e-4 (half of total thickness which is not correct).
model.il=p_shell('dbval 1 laminate 1 1.2e-3 0', ...
   'dbval 2 laminate z0=-8.5e-4 3 2.5e-4 0 1 1.2e-3 0 ');
```

The difference of static response is:

```
d =  2.5244e-06
d2 = 3.1700e-06
difference of static response 25.5743%
```

The results show that the plate with only one piezoelectric element on the bottom is less stiff then the case with piezos on both the top and bottom: the tip displacement is 25% higher (3.17 $\mu m/V$ vs 2.52 $\mu m/V$) .

When the piezoelectric element is at the top of the plate, then the offset should be equal to the thickness of the main plate divided by two (negative value).

5.2 Automated inclusion of piezo patches

The automated procedure requires the plate model to be defined in mm. In order to add the patches, one needs to make a structure which contains the initial plate model and a description of the different patches to be added.

In this example, the patches are assigned with `+Rect.Sonox_P502_iso.5525TH0_25` where the sign +/- specifies if the patch is on the top (+) or the bottom (-) of the plate, the first argument gives the geometry (Rect for Rectangular), the second argument is the piezoelectric material type from the database, the next argument is the size (55mm x 25mm here) and the last argument is the thickness (0.25mm).

So the first step `d_piezo('TutoPzMeshingAuto-s1')` is to make the host plate. Here, we define a mesh so that the edges of the piezoelectric patches will correspond to the mesh (as in the previous example of meshing manually).

Then in `d_piezo('TutoPzMeshingAuto-s2')` we add four rectangular piezoelectric patches whose size and shape comply with the mesh.

```
%% Step 2: Add patches
RG.list={'Name','Lam','shape'
   'Main_plate', model,''  % Base structure
   'Act1', ... % name of patch
   'BaseId1 -Rect.Sonox_P502_iso.5525TH0_25 +Rect.Sonox_P502_iso.5525TH0_25', ...
     struct('shape','LsRect', ... % Remeshing strategy (lsutil rect here)
       'xc',15+55/2,'yc',12+25/2,'alpha',0,'tolE',.1)
    'Act2', ... % name of patch
```

```
   'BaseId1 -Rect.Sonox_P502_iso.5525TH0_25 +Rect.Sonox_P502_iso.5525TH0_25', ...
     struct('shape','LsRect', ... % Remeshing strategy (lsutil rect here)
       'xc',15+55/2,'yc',63+25/2,'alpha',0,'tolE',.1)
       };
mo1=d_piezo('MeshPlate',RG);
mo1=stack_rm(mo1,'info','Electrodes'); % Obsolete stack field to be removed
% To avoid warning due to the use of simplified piezo properties.
mo1=p_piezo('DToSimple',mo1);
```

The command `+Rect.Sonox_P502_iso.5525TH0_25` refers to a patch which is added on the top of the plate (+), is rectangular (Rect), is made of Sonox_P502_iso, is of size 55mm x 25mm (5525) and thickness 0.25 mm (TH0_25). Then we need to specify the position of the patch with 'xc' and 'yc' which correspond to the coordinates of its center.

The model obtained is here the same as the one previously obtained with manual meshing. The static response when actuating the first piezo patch is computed in `d_piezo('TutoPzMeshingAuto-s3')` . The value of the tip displacement is

```
d1t = 2.5244e-09
```

Note that the automated meshing introduces the patches in the order of the description, so that here the first piezoelectric patch is on the bottom (-), and the static response is downwards, as in the case of manual meshing. This allows to compared the displacement at the tip which has the same value (2.52 $\mu m/V$).

Automated meshing can also be done when the patch edges do not coincide with the mesh of the host plate. In this case, automated local remeshing is applied around the patches inserted. In `d_piezo('TutoPzMeshingAuto-s4')` we take the example of an initial mesh of the plate with a uniform mesh of prescribed element size (here around 7 mm). The script to add the four patches is strictly identical, but the result is different due to the local remeshing.

The uniform mesh is generated with

```
%% Step 4 : use local remeshing element size is 7 mm
 model=feutil('objectquad 1 1',[0 0 0;1 0 0;0 1 0], ...
    feutil('refineline 7',[0 463]), ...
    feutil('refineline 7',[0 100]));
     model.unit='mm';
```

The values of $d1t$ and $d2t$ correspond to the tip displacement when actuating the first piezo patch, for the first model where the edges of the piezos correspond to the element sides in the mesh, and when there is a local remeshing. Due to the effect of the local remeshing (Figure 5.4), there is a slight difference ($\approx 1\%$). The values of tip displacement with the two different meshes are:

```
d1t = 2.5244e-09
d2t = 2.4901e-09
```

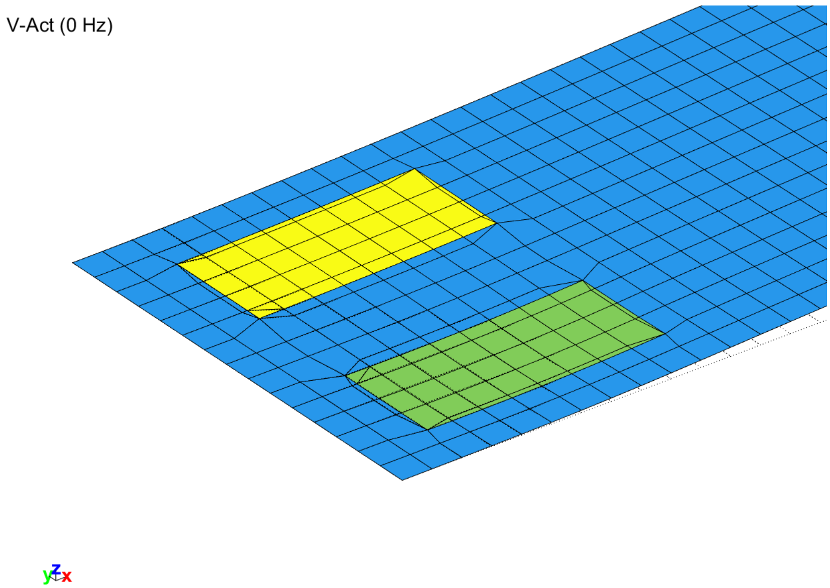


Figure 5.4: Static response to a unit voltage application on one of the bottom piezoelectric patches

In `d_piezo('TutoPzMeshingAuto-s5')` , in order to check this effect, we can use a finer mesh of the host structure and check for the convergence.

We see that when the plate mesh is finer (Figure 5.5), the local effect of remeshing is less and we converge to the tip displacement when the piezoelectric patch edges coincide with the mesh.

A last example in `d_piezo('TutoPzMeshingAuto-s6')` is the addition of circular patches, where we also check for convergence. The introduction of circular patches is done with the arguments `-Disk.Sonox_P502_iso.RC10TH0_25` and `+Disk.Sonox_P502_iso.RC10TH0_25` which specifies a disk made of *Sonox_P502_iso* material with a radius RC of 10mm and a thickness TH of 0.25 mm (same thickness as the rectangular patches).

The circular patches are introduced with the following syntax:

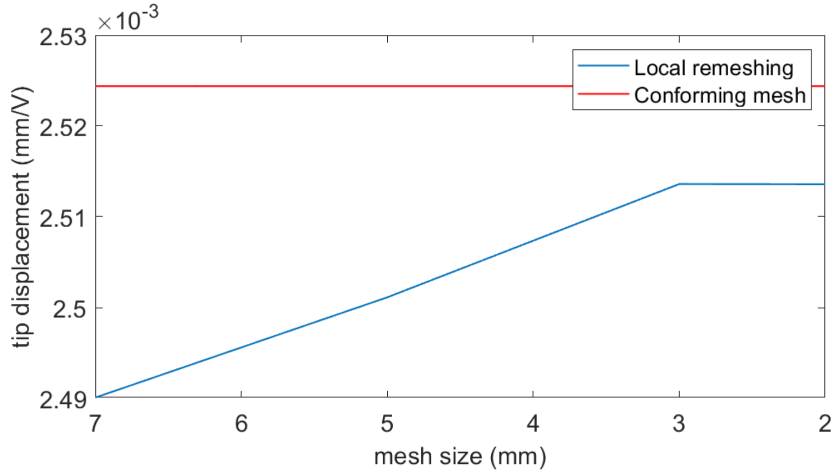


Figure 5.5: Convergence of the tip displacement when the size of the elements of the main plate is varied, using local remeshing. The red line corresponds to the tip displacement with the mesh conforming with the patch edges

```
RG.list={'Name','Lam','shape'
   'Main_plate', model,''  % Base structure
   'Act1', ... % name of patch
   'BaseId1 -Disk.Sonox_P502_iso.RC10TH0_25 +Disk.Sonox_P502_iso.RC10TH0_25', ...
   struct('shape','lscirc','xc',15+55/2,'yc',12+25/2),
   'Act2', ... % name of patch
   'BaseId1 -Disk.Sonox_P502_iso.RC10TH0_25 +Disk.Sonox_P502_iso.RC10TH0_25', ...
   struct('shape','lscirc','xc',15+55/2,'yc',63+25/2)};
```

The local remeshing can be see in Figure 5.6.
The last step `d_piezo('TutoPzMeshingAuto-s7')` allows to check convergence as for the rectangular patches with automated meshing.

The convergence of the tip displacement can be see in Figure 5.7. Note that the tip displacement is about 5 times smaller than with the rectangular patches, which is mainly due to the circular shape (less efficient than the rectangular shape for plate bending).

5.3 Using predefined patches

The different types of patches that can be integrated in plate meshes can be seen by typing the command `m_piezo Patch`. At the moment of writing this documentation, the output is

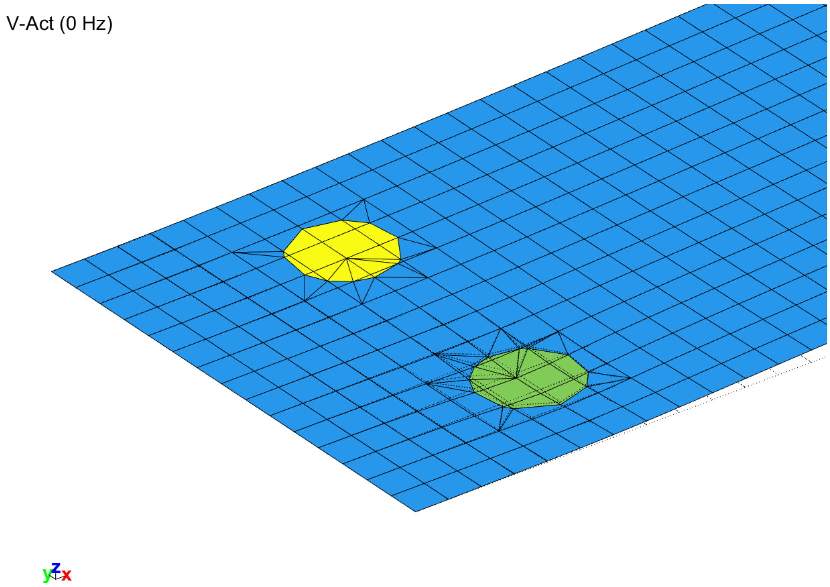


Figure 5.6: Static response to a unit voltage application on one of the bottom circular piezoelectric patches

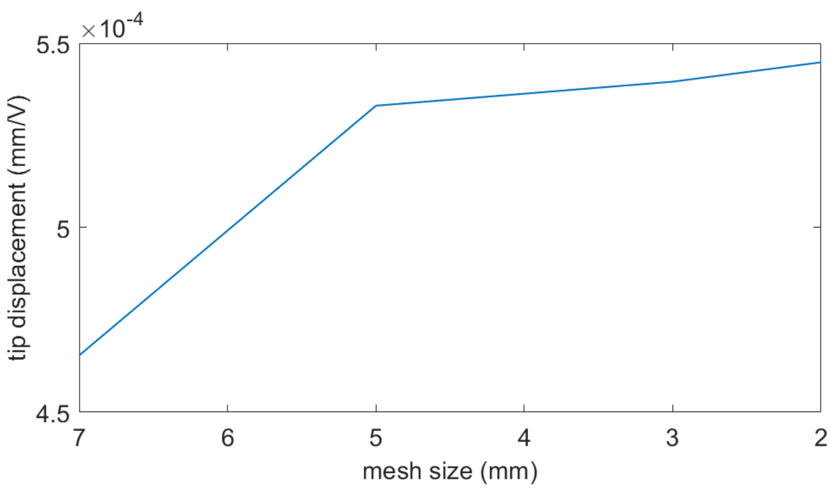


Figure 5.7: Convergence of the tip displacement when the size of the elements of the main plate is varied, for a circular piezoelectric patch

```
{'Noliac.NCE51.RC12TH1'  }    {'Noliac, material, disk ODiameter and THickness'}
{'SmartM.MFC-P1.2814'    }    {'Smart materials MFC d33, .WiLe (dims mm)'     }
{'SmartM.MFC-P2.2814'    }    {'Smart materials MFC d31, .WiLe (dims mm)'     }
{'Disk.Material.RC12TH1' }    {'Circular patch, material, geometry'           }
{'Rect.Material.2814TH_5'}    {'Rectangular patch, material, geometry'        }
```

The last two types of patches are the ones used in the previous scripts, which correspond to rectangular and circular patches, for which the material can be chosen, as well as the geometrical properties. The Noliac patch is a disk, and could be described with the generic Disk.Material.RCXXTHX command.

There also exist on the market packaged piezoelectric patches which are made of several layers, such as the Macro Fiber Composite (https://smart-material.com). The properties of the different layers are described in more details in section 7 . Two types of MFCs exist (P1, and P2).

The following example is the integration of two $P1$-type MFC patches on a beam, modeled with multi-layer shell elements. The example is further discussed in section 7.4 .

The mesh resulting from `d_piezo('TutoPzMeshingMFC-s1')` is represented in Figure 5.8.

The syntax for the MFC patches is:

```
RG.list={'Name','Lam','shape'
   'Main_plate', model,''  % Base structure
   'Act1', ... % name of patch
   'BaseId1 +SmartM.MFC-P1.8528 -SmartM.MFC-P1.8528', ... % Layout definition
     struct('shape','LsRect', ... % Remeshing strategy (lsutil rect here)
       'xc',RO.c+RO.a/2,'yc',RO.d+RO.b/2,'alpha',0,'tolE',.1)
       };
```

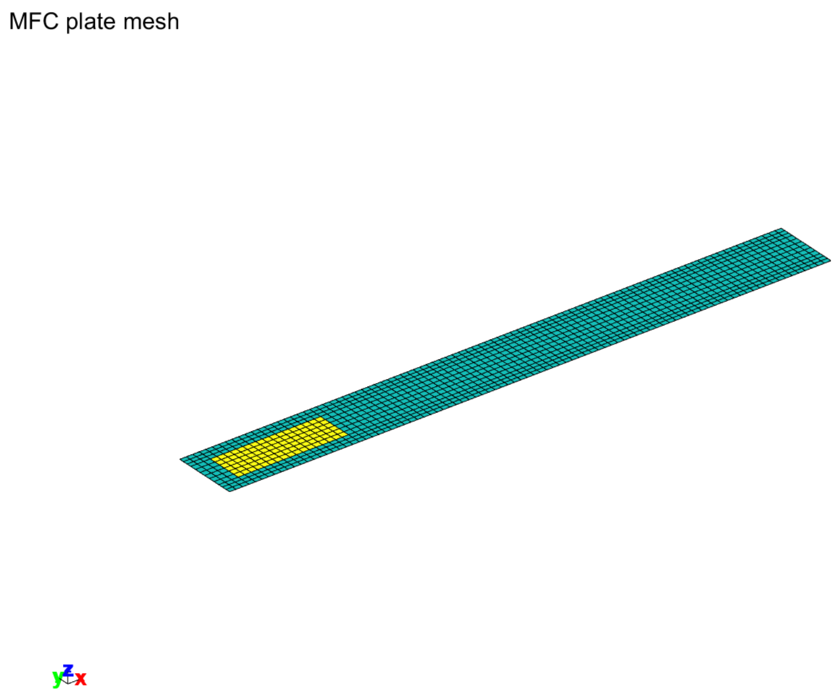


Figure 5.8: Mesh of the plate with MFC transducers on top and bottom. The different colours represent the different groups

6

Model reduction and I/O state-space models

Contents

6.1 Model reduction theory

Finite element models of structures need to have many degrees of freedom to represent the geometrical details of complex structures. For models of structural dynamics, one is however interested in

- a restricted frequency range ($s = i\omega \in [\omega_1 \;\; \omega_2]$)
- a small number of inputs and outputs (b, c)
- a limited parameter space α (updated physical parameters, design changes, non-linearities, etc.)

These restrictions on the expected predictions allow the creation of low order models that accurately represent the dynamics of the full order model in all the considered loading/parameter conditions. Model reduction procedures are discrete versions of Ritz/Galerkin analyzes: they seek solutions in the subspace generated by a reduction matrix T. Assuming $\{q\} = [T]\{q_R\}$, the second order finite element model (4.1) is projected as follows

$$\begin{gathered}\left[T^T M T s^2 + T^T C T s + T^T K T\right]_{NR\times NR} \{q_R(s)\} = \left[T^T b\right]_{NR\times NA} \{u(s)\}_{NA\times 1} \\ \{y(s)\}_{NS\times 1} = [cT]_{NS\times NR} \{q_R(s)\}_{NR\times 1}\end{gathered} \tag{6.1}$$

Modal analysis, model reduction, component mode synthesis, and related methods all deal with an appropriate selection of singular projection bases ($[T]_{N\times NR}$ with $NR \ll N$). This section summarizes the theory behind these methods with references to other works that give more details.

The solutions provided by *SDT* make two further assumptions which are not hard limitations but allow more consistent treatments while covering all but the most exotic problems. The projection is chosen to preserve reciprocity (left multiplication by T^T and not another matrix). The projection bases are assumed to be real.

An accurate model is defined by the fact that the input/output relation is preserved for a given frequency and parameter range

$$[c]\,[Z(s,\alpha)]^{-1}\,[b] \approx [cT]\left[T^T Z(s,\alpha) T\right]^{-1}\left[T^T b\right] \tag{6.2}$$

where $Z(s,\alpha)$ is the dynamic flexibility $Z(s) = \left[K + Cs + Ms^2\right]$ for a given set of parameters α

Traditional modal analysis, combines normal modes and static responses. *Component mode synthesis* (CMS) methods extend the selection of boundary conditions used to compute the normal modes. The *SDT* further extends the use of reduction bases to parameterized problems.

A key property for model reduction methods is that the input/output behavior of a model only depends on the vector space generated by the projection matrix T. Thus range(T) = range$(\tilde{T})$ implies that

$$[cT]\left[T^T Z T\right]^{-1}\left[T^T b\right] = \left[c\tilde{T}\right]\left[\tilde{T}^T Z \tilde{T}\right]^{-1}\left[\tilde{T}^T b\right] \tag{6.3}$$

This **equivalence property** is central to the flexibility provided by the *SDT* in CMS applications (it allows the decoupling of the reduction and coupled prediction phases) and modeshape expansion methods (it allows the definition of a static/dynamic expansion on sensors that do not correspond to DOFs).

6.1.1 Normal mode models

Normal modes are defined by the eigenvalue problem

$$\left(-\omega_j^2 [M] + [K]\right)_{N\times N} \{\phi_j\}_{N\times 1} = \{0\}_{N\times 1} \tag{6.4}$$

based on inertia properties (represented by the positive definite mass matrix M) and underlying elastic properties (represented by a positive semi-definite stiffness K). The matrices being positive, there are N independent eigenvectors $\{\phi_j\}$ (forming a matrix noted $[\phi]$) and eigenvalues ω_j^2 (forming a diagonal matrix noted $\left[\backslash \omega_j^2 \backslash\right]$).

As solutions of the eigenvalue problem (6.4), the full set of N normal modes verify two **orthogonality conditions** with respect to the mass and the stiffness

$$[\phi]^T [M] [\phi] = \left[\backslash \mu_j \backslash\right]_{N\times N} \quad \text{and} \quad [\phi]^T [K] [\phi] = \left[\backslash \mu_j \omega_j^2 \backslash\right] \tag{6.5}$$

where μ is a diagonal matrix of modal masses (which are quantities depending uniquely on the way the eigenvectors ϕ are scaled).

In the *SDT*, the normal modeshapes are assumed to be mass normalized so that $[\mu] = [I]$ (implying $[\phi]^T [M] [\phi] = [I]$ and $[\phi]^T [K] [\phi] = \left[\backslash \omega_j^2 \backslash\right]$). The **mass normalization** of modeshapes is independent from a particular choice of sensors or actuators.

Another traditional normalization is to set a particular component of $\tilde{\phi}_j$ to 1. Using an output shape matrix this is equivalent to $c_l \tilde{\phi}_j = 1$ (the observed motion at sensor c_l is unity). $\tilde{\phi}_j$, the modeshape with a component scaled to 1, is related to the mass normalized modeshape by $\tilde{\phi}_j = \phi_j/(c_l \phi_j)$.

$$m_j(c_l) = (c_l \phi_j)^{-2} \tag{6.6}$$

is called the **modal or generalized mass** at sensor c_l.

Modal stiffnesses are are equal to

$$k_j(c_l) = (c_l\phi_j)^{-2}\,\omega_j^2 \tag{6.7}$$

The use of mass-normalized modes, simplifies the normal mode form (identity mass matrix) and allows the direct comparison of the contributions of different modes at similar sensors. From the orthogonality conditions, one can show that, for an undamped model and mass normalized modes, the dynamic response is described by a sum of modal contributions

$$[\alpha(s)] = \sum_{j=1}^{N} \frac{\{c\phi_j\}\left\{\phi_j^T b\right\}}{s^2+\omega_j^2} \tag{6.8}$$

which correspond to pairs of complex conjugate poles $\lambda_j = \pm i\omega_j$.

In practice, only the first few low frequency modes are determined, the series in (6.8) is truncated, and a correction for the truncated terms is introduced (see section 6.1.2).

6.1.2 Static correction to normal mode models

Normal modes are computed to obtain the spectral decomposition (6.8). In practice, one distinguishes modes that have a resonance in the model bandwidth and need to be kept and higher frequency modes for which one assumes $\omega \ll \omega_j$. This assumption leads to

$$[c]\left[Ms^2+K\right]^{-1}[b] \approx \sum_{j=1}^{N_R} \frac{[c]\{\phi_j\}\{\phi_j\}^T[b]}{s^2+\omega_j^2} + \sum_{j=N_R+1}^{N} \frac{[c]\{\phi_j\}\{\phi_j\}^T[b]}{\omega_j^2} \tag{6.9}$$

For applications in reduction of finite element models, from the orthogonality conditions (6.5), one can easily show that for a structure with no rigid body modes (modes with $\omega_j = 0$)

$$[T_A] = [K]^{-1}[b] = \sum_{j=1}^{N} \frac{\{\phi_j\}\left\{\phi_j^T b\right\}}{\omega_j^2} \tag{6.10}$$

The static responses $[K]^{-1}[b]$ are called **attachment modes** in Component Mode Synthesis applications [4]. The inputs $[b]$ then correspond to unit loads at all interface nodes of a coupled problem.

One has historically often considered **residual attachment modes** defined by

$$[T_{AR}] = [K]^{-1}[b] - \sum_{j=1}^{NR} \frac{\{\phi_j\}\left\{\phi_j^T b\right\}}{\omega_j^2} \qquad (6.11)$$

where NR is the number of normal modes retained in the reduced model.

The vector spaces spanned by $[\phi_1 \ldots \phi_{NR} \;\; T_A]$ and $[\phi_1 \ldots \phi_{NR} \;\; T_{AR}]$ are clearly the same, so that reduced models obtained with either are dynamically equivalent. For use in the *SDT*, you are encouraged to find a basis of the vector space that diagonalizes the mass and stiffness matrices (normal mode form which can be easily obtained with `fe_norm`).

Reduction on modeshapes is sometimes called the **mode displacement method**, while the addition of the **static correction** leads to the **mode acceleration method**.

When reducing on these bases, the selection of retained normal modes guarantees model validity over the desired frequency band, while adding the static responses guarantees validity for the spatial content of the considered inputs. The reduction is only valid for this restricted spatial/spectral content but very accurate for solicitation that verify these restrictions.

Defining the bandwidth of interest is a standard difficulty with no definite answer. The standard, but conservative, criterion (attributed to Rubin) is to keep modes with frequencies below 1.5 times the highest input frequency of interest.

6.1.3 Illustration of modal truncation error and static correction

The concepts of model reduction using normal modes and static corrections are illustrated on a cantilever beam example. Figure 6.1 shows a tower of height $L = 140m$ excited at the tip and for which the displacement in the horizontal direction is computed at the same point where the excitation is applied (leading to a collocated transfer function). The full model solution is compared to a truncated solution where $NR = 3$. The beam is made of concrete with a Young's modulus $E = 30GPa$, a Poisson's ratio $\nu = 0.2$ and a mass density $\rho = 2200kg/m^3$. The section of the beam is a square hollow section with outer dimensions $20m$ x $20m$ and a thickness of $30cm$.

Figure 6.1: Cantilever beam excited at the tip in the horizontal direction

The script below details explicitely the different steps to compute the response using the reduced model. It uses low-level calls which are not computationally efficient in SDT but illustrate the strategy. The first step `d_piezo('TutoPzTowerRed-s1')` consists in building the mesh, preassembling the matrices $[K]$ and $[M]$ and defining the tip-load and tip-sensor.

The frequency band of interest is [0 10] Hz, in `d_piezo('TutoPzTowerRed-s2')` , we only keep the mode shapes which have a frequency below 15 Hz (1.5 x 10 Hz), i.e. in this case the first three bending modes, and compute the transfer function between a force acting on the tip of the beam in the horizontal direction, and the displacement at the tip in the same direction. We use first explicit computations after building matrices, which is not the most efficient in terms of computational time, but is aimed at illustrating the methodology with low-level calls.

```
%% Step 2 : Build Reduced basis (3 modes) and compute TF

% M and K matrices have been built with fe_mknl in d_piezo('MeshTower')
M=model.K{1}; K=model.K{2};
% Excitation
Load=fe_load(model); b=Load.def;

% Projection matrix for output
sens=fe_case(model,'sens');
[Case,model.DOF]=fe_mknl('init',model); % Build Case.T for active dofs
cta=sens.cta*Case.T;
```

```
% Compute Modeshapes
def=fe_eig(model);
% Rearrange with active dofs only
def.def=Case.T'*def.def; def.DOF=Case.T'*def.DOF;

% Build reduced basis on three modes
T=def.def(:,1:3);

% Reduce matrices and compute response - explicit computation
Kr=T'*K*T; Mr=T'*M*T; br=T'*b;

% Define frequency vector for computations
w=linspace(0,30*2*pi,512);% Extended frequency range

% Solution with full and reduced matrices - explicit computation
% Use loss factor =0.02 = default for SDT (1% modal damping)
for i1=1:length(w)
    U(:,i1)=(K*(1+0.02*1i)-w(i1)^2*M)\b;
    %Default loss factor is 0.02 in SDT
    Ur(:,i1)=(Kr*(1+0.02*1i)-w(i1)^2*Mr)\br; %Reduced basis
end

% Extract response on sensor and visualize in iicom
out1=cta*U; out2=cta*(T*Ur);

% Change output format to be compatible with iicom
C0m=d_piezo('BuildC1',w'/(2*pi),out1.','d-top','F-top'); C0m.name='Full';
C1m=d_piezo('BuildC1',w'/(2*pi),out2.','d-top','F-top'); C1m.name='3md';
ci=iiplot;iicom(ci,'curveinit',{'curve',C0m.name,C0m;'curve',C1m.name,C1m});
iicom('submagpha')
d_piezo('setstyle',ci); set(gca,'XLim',[0 10])
```

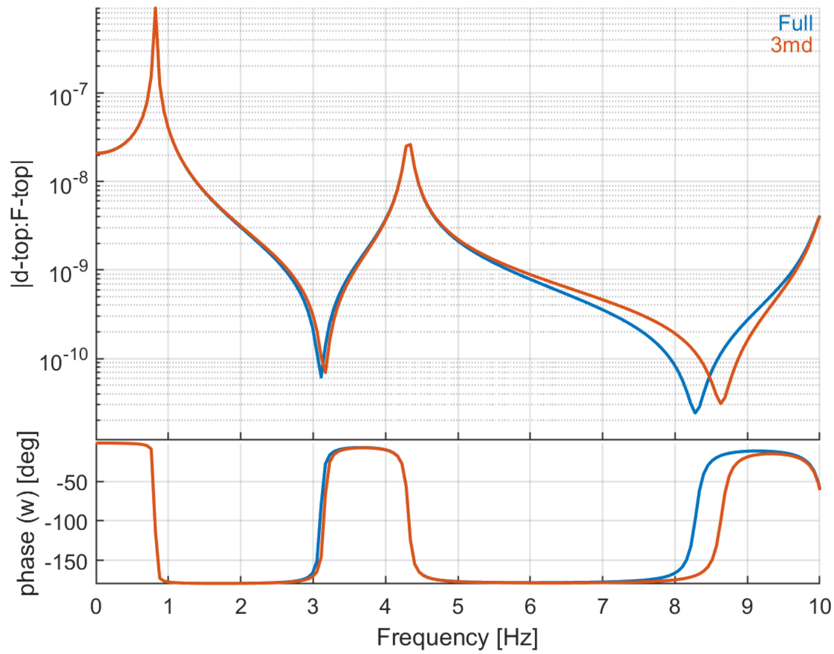


Figure 6.2: Transfer function between the tip displacement and the tip force: Comparison between the exact solution and a solution obtained by truncating the expansion in the modal basis to the first three bending modes

The transfer function computed with the full model is compared to the reduced model using only the first three modes of the beam in Figure 6.2. The agreement is very good at the resonances (poles) but there are some discrepancies around $\omega = 0$ (not clearly visible due to the logarithmic scale) and close to the anti-resonances (zeros). This is due to the contribution of the modes at higher frequencies which are not taken into account.

Frequency response functions are more efficiently computed using `fe_simul` in SDT. The general strategy to follow if one wants to use this function to compute the response of any given reduced model is to call `fe_reduc`. Several model reduction methods exist in the native function, but if one wants to implement another strategy, it is possible to call an external function. As using only mode shapes for model reduction is not accurate enough, this strategy is not implemented in `fe_reduc`.

In `d_piezo('TutoPzTowerRed-s3')` , we show how to call an external function to build a reduced basis consisting only of the first N modes of a structure (here $N = 3$). The results obtained are identical to the direct explicit approach (Figure 6.2).

```
%% Step 3 - Compute with fe_simul Full/3md
```

```
% Full model response
C0=fe_simul('dfrf -sens',stack_set(model,'info','Freq',w/(2*pi)));
% -sens option projects result on sensors only
C0.name='Full'; C0.X{2}={'d-top'}; C1.X{3}={'F-top'};

% With reduced basis 3 modes
model = stack_set(model,'info','EigOpt',[5 3 0]); % To keep 3 modes
% Build super-element with 3 modes
SE1=fe_reduc('call d_piezo@modal -matdes 2 1 3 4',model);

% Make model with a single super-element
SE0 = struct('Node',[],'Elt',[]);
mo1 = fesuper('SEAdd 1 -1 -unique -initcoef -newID se1',SE0,SE1) ;

% Define input/output
mo1=fe_case(mo1,'SensDOF','Output',21.01);
mo1=fe_case(mo1,'DofLoad','Input',21+.01,1);

% Compute response with fe_simul
C1=fe_simul('dfrf -sens',stack_set(mo1,'info','Freq',w/(2*pi)));
C1.name='3md+SE'; C1.X{2}={'d-top'};
```

It requires to build a function *modal.m* which is called by fe_reduc, the script of this function is given below:

```
function SE=modal()
%% just modal

% Transfers model to function
eval(iigui({'model','Case'},'MoveFromCaller'))
eval(iigui({'RunOpt'},'GetInCaller'))

% clean up model to remove info about matid/proid
SE=feutil('rmfield',model,'pl','il','unit','Stack');

% Compute modeshapes
def=fe_eig({model.K{1},model.K{2},Case.T, ...
   Case.mDOF},stack_get(model,'info','Eigopt','g'));
```

```
% reduce matrics based on N modes
SE.K = feutilb('tkt',def.def,model.K);
SE.DOF=[1:size(SE.K{1},1)]'+.99; % .99 DOFS
SE.Klab={'m' 'k' 'c' '4'};
SE.TR=def; SE.TR.adof=SE.DOF;

% assign to output
assignin('caller','T',SE)
assignin('caller','Case','');
```

The truncation error can be drastically reduced using static correction.
In d_piezo('TutoPzTowerRed-s4') , a new reduced basis is therefore computed using fe_norm to orthogonalize the basis with respect to $[K]$ and $[M]$. This is illustrated using low level calls.

```
%% Step 4 : Reduced basis 2 : 3 modes + static corr - explicit

Tstat=K\b;
T=fe_norm([def.def(:,1:3) Tstat],M,K); % Orthonormalize vectors

% Reduced matrices
Kr=T'*K*T; Mr=T'*M*T; br=T'*b;

% Ref solution
for i1=1:length(w)
     Ur2(:,i1)=(Kr*(1+0.02*1i)-w(i1)^2*Mr)\br;
end

out3=cta*(T*Ur2);

C2m=d_piezo('BuildC1',w'/(2*pi),out3.','d-top','F-top'); C2m.name='3md+Stat';
iicom(ci,'curveinit',{'curve',C0m.name,C0m;'curve',C2m.name,C2m});
iicom('submagpha'); d_piezo('setstyle',ci);
```

The transfer function computed using the static correction to the applied load is represented in Figure 6.3 in the extended frequency range from 0 to 30 Hz. The figure shows that the match is now excellent from 0 to 10 Hz. The static response and the anti-resonances (zeros) are predicted accurately. This is very important for active vibration control applications where the frequencies of the zero play a major role for predicting the performance of the active control scheme implemented.

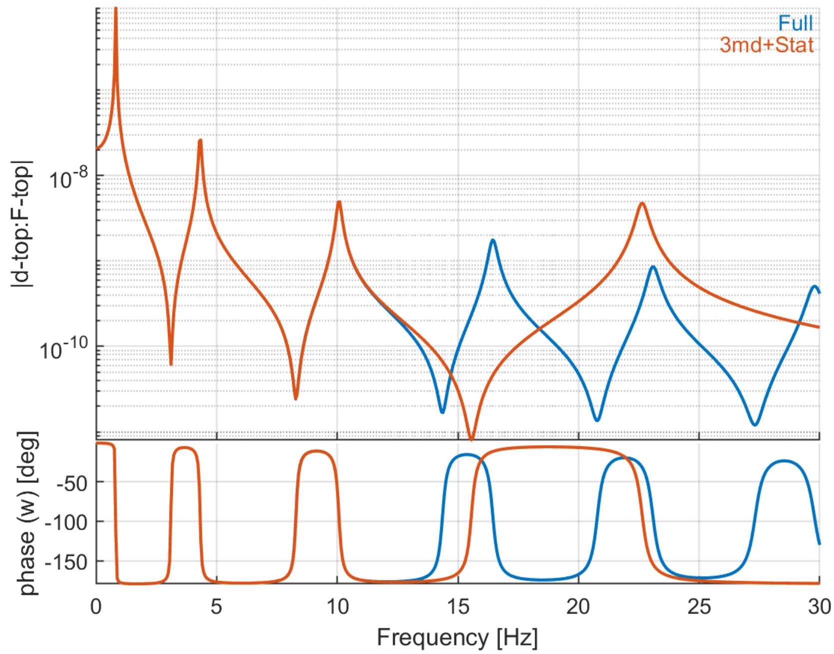


Figure 6.3: Transfer function between the tip displacement and the tip force: Comparison between the exact solution and a solution obtained with a reduced basis containing the first three modes and the static correction to the applied tip-load [0-30 Hz]

Extending the frequency range for computations allows to show the behavior of the model at higher frequencies, and illustrates the fact that an additional non-physical mode is present when using normal modes and static correction, whose contribution in the frequency band of interest is the same as the combination of all the contributions of the higher frequency modes for the full model. Note that the first zero after the three modes in the frequency band of interest is not correct, even with the static correction.

The model reduction with N modes + static correction is natively implemented in the `fe_reduc` as illustrated in `d_piezo('TutoPzTowerRed-s5')`

```
%% Step 5 with fe_simul
model=d_piezo('meshtower');
model = stack_set(model,'info','EigOpt',[5 3 0]); % To keep 3 modes

% Reduce model with 3 modes and static correction
mo1=fe_reduc('free -matdes -SE 2 1 3 4 -bset',model);
% Compute response with fe_simul and represent
```

```
C2=fe_simul('dfrf -sens',stack_set(mo1,'info','Freq',w/(2*pi))); %
C2.name='3md+Stat-SE'; C2.X{2}={'d-top'};
ci=iiplot;iicom(ci,'curveinit',{'curve',C0.name,C0;'curve',C2.name,C2});
iicom('submagpha'); d_piezo('setstyle',ci);
```

It gives identical results as the direct explicit computation (Figure 6.3).

If the number of modes in the frequency band of interest becomes large, the technique becomes much more costly: the eigenvalue solver will need to compute more mode shapes, which increases strongly the computational time, and the number of equations to solve is also high. Usually, as the frequency increases, the modal density increases as well, so that the modal projection technique is not efficient anymore.

6.2 State space models

6.2.1 General theory

While normal mode models are appropriate for structures, **state-space models** allow the representation of more general linear dynamic systems and are commonly used in the *Control Toolbox* or SIMULINK. The standard form for state space-models is

$$\begin{aligned}\{\dot{x}\} &= [A]\,\{x(t)\} + [B]\,\{u(t)\}\\ \{y\} &= [C]\,\{x(t)\} + [D]\,\{u(t)\}\end{aligned} \tag{6.12}$$

with inputs $\{u\}$, states $\{x\}$ and outputs $\{y\}$. State-space models are represented in the *SDT*, as generally done in other Toolboxes for use with MATLAB, using four independent matrix variables `a`, `b`, `c`, and `d` (you should also take a look at the LTI state-space object of the *Control Toolbox*).

The transfer functions from inputs to outputs are described in the frequency domain by

$$\{y(s)\} = \Big([C]\,[s\ I - A]^{-1}\,[B] + [D]\Big)\,\{u(s)\} \tag{6.13}$$

A state-space representation of the nominal structural model (4.1) is given by

$$\begin{aligned}\left\{\begin{array}{c} q' \\ q'' \end{array}\right\} &= \left[\begin{array}{cc} 0 & I \\ -M^{-1}K & -M^{-1}C \end{array}\right]\left\{\begin{array}{c} q \\ q' \end{array}\right\} + \left[\begin{array}{c} 0 \\ M^{-1}b \end{array}\right]\{u(t)\}\\ \{y(t)\} &= [c\ \ 0]\left\{\begin{array}{c} q \\ q' \end{array}\right\}\end{aligned} \tag{6.14}$$

The interest of this representation is mostly academic because it does not preserve symmetry (a useful feature of models of structures associated to the assumption of reciprocity) and because $[M]^{-1}[K]$ is usually a full matrix (so that the associated memory requirements for a realistic finite element model would be prohibitive). The *SDT* thus always starts by transforming a model to the normal mode form and the associated state-space model .

The natural state-space representation of normal mode models (6.4) is given by

$$\begin{aligned} \left\{ \begin{array}{c} p' \\ p'' \end{array} \right\} &= \left[\begin{array}{cc} 0 & I \\ -\Omega^2 & -\Gamma \end{array} \right] \left\{ \begin{array}{c} p \\ p' \end{array} \right\} + \left[\begin{array}{c} 0 \\ \phi^T b \end{array} \right] \{u(t)\} \\ \{y(t)\} &= [c\phi \;\; 0] \left\{ \begin{array}{c} p \\ p' \end{array} \right\} \end{aligned} \tag{6.15}$$

Transformations to this form are provided by `nor2ss` and `fe2ss`. Note however that, as demonstrated in section 6.1.3 , using strictly the modeshapes in the reduced state-space model is generally not sufficient to obtain a proper representation of the zeros, which are very important in control applications. `fe2ss` uses two different representations for reduced state-space models based on normal modes and static corrections, which are detailed in the next section.

6.2.2 State-space formulations with static correction

As shown in section 6.1.2 , it is possible to augment the basis of modeshapes with static corrections to applied loads to form an accurate reduction basis. Once the basis is orthonormalized, the additional vectors due to static corrections appear as additional modes in the basis. The transformation to a state-space model is straightforward using (6.15), but has the disadvantage to increase considerably the size of the model if a large number of input forces are considered. An alternative is presented below.

We recall that the modal decomposition of the flexibility matrix given by:

$$\left[Ms^2+K\right]^{-1}[b] \approx \sum_{j=1}^{NR} \frac{\{\phi_j\}\{\phi_j\}^T[b]}{s^2+\omega_j^2} + \sum_{j=NR+1}^{N} \frac{\{\phi_j\}\{\phi_j\}^T[b]}{\omega_j^2} \tag{6.16}$$

can be rewritten using (6.10) as:

$$\left[Ms^2+K\right]^{-1}[b] \approx \sum_{j=1}^{NR} \frac{\{\phi_j\}\{\phi_j\}^T[b]}{s^2+\omega_j^2} + [K]^{-1}[b] - \sum_{j=1}^{NR} \frac{\{\phi_j\}\{\phi_j\}^T[b]}{\omega_j^2} \tag{6.17}$$

which shows that the last two terms are proportional to the input $[b]$ and constant (do not depend on ω). They can therefore be introduced in matrix `d` of the state-space model, with matrices `a`,`b` and `c` corresponding to the case where only the normal modes are retained (6.15). The state-space model then becomes :

$$\begin{gathered}\left\{\begin{array}{c} p' \\ p'' \end{array}\right\} = \left[\begin{array}{cc} 0 & I \\ -\Omega^2 & -\Gamma \end{array}\right] \left\{\begin{array}{c} p \\ p' \end{array}\right\} + \left[\begin{array}{c} 0 \\ \phi^T b \end{array}\right] \{u(t)\} \\ \{y(t)\} = [c\phi \;\; 0] \left\{\begin{array}{c} p \\ p' \end{array}\right\} + [d]\,[u]\end{gathered} \tag{6.18}$$

where

$$[d] = [c]\,[K]^{-1}\,[b] - \sum_{j=1}^{NR} \frac{[c]\,\{\phi_j\}\,\{\phi_j\}^T\,[b]}{\omega_j^2} \tag{6.19}$$

The advantage of building the state-space model with such an approach is that the number of degrees of freedom is not increased, and that the $[d]$ matrix introduces a static contribution which cannot be destabilized by a controller, if active damping is applied for example. It introduces however a non-physical feed-through term which is directly proportional to the input force. This violates the principles of wave propagation: if the response to a rectangular impulse is computed for example, the wave should arrive to the sensors after a time related to the distance and wave velocity in the medium, and here, due to the non-zero $[d]$ matrix, a part of it arrives instantaneously.

Building state-space models with static corrections but no additional high-frequency modes can be done with the option `-dterm` in `fe2ss`.

6.2.3 State-space models with static correction: illustration on the tower example

The cantilever beam (tower) example presented in section 6.1.3 is considered again, and the two approaches (additional mode and $[d]$ matrix approach) to build a state-space model are illustrated below.

The first step `d_piezo('TutoPzTowerSS-s1')` consists in building the mesh and defining actuators and sensors as before.

Then in `d_piezo('TutoPzTowerSS-s2')` , the state-space model is created either with a 'dterm' or with an additional mode, and the transfer function between 'F-top' and 'd-top' is computed.

```
% Step 2 : State-space models
[sys,TR] = fe2ss('free 5 3 0 -dterm',model);
[sys2,TR2] = fe2ss('free 5 3 0 ',model);
```

```
w=linspace(0,30*2*pi,512);% Frequency range
% Convert to curve object and relabel X (fe2ss uses dofs and not sens/act names)
C1=qbode(sys,w,'struct-lab');C1.name='SS-dterm'; C1.X{2}={'d-top'};
C2=qbode(sys2,w,'struct-lab');C2.name='SS-mode'; C2.X{2}={'d-top'};
ci=iiplot;
iicom(ci,'curveinit',{'curve',C1.name,C1;'curve',C2.name,C2}); iicom('submagpha')
d_piezo('setstyle',ci);
%
```

Figure 6.4 shows that with the two approaches, the state-space model gives a correct representation of the transfer function in the frequency band of interest [0 10] Hz, but that the behavior is different at higher frequencies. The `-dterm` option does not introduce a high frequency mode, but the transfer function does not have a high-frequency roll-off due to the added constant term.

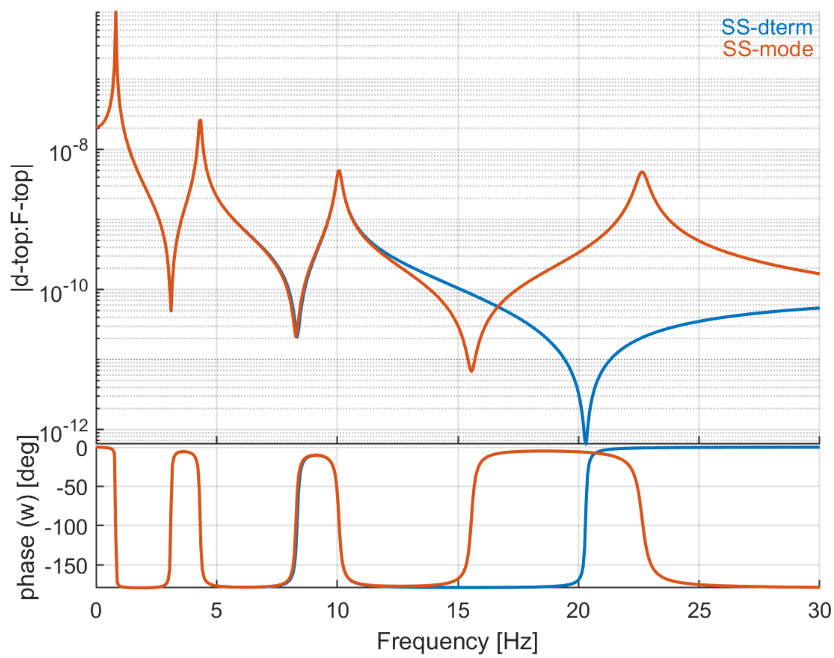


Figure 6.4: Transfer function between the tip displacement and the tip force: Comparison between state-space models using the d-term approach or the additional high-frequency mode for static correction

6.3 State-space models with imposed displacement and acceleration

In order to be able to impose a displacement or acceleration instead of a mechanical load, the displacement vector $\{q\}$ is divided in two contributions:

$$\{q\} = [T_C]\{q_C\} + [T_I]\{q_I\} \tag{6.20}$$

where $\{q_I\}$ is the part of $\{q\}$ on which either a displacement or an acceleration need to be imposed (I is used as a reference to "interface"), and $\{q_C\}$ represents the remaining dofs where no displacement or acceleration is imposed. As will be discussed later, different choices can be used for $[T_I]$ and $[T_C]$.

We assume a general form of these matrices where for the degrees of freedom related to imposed displacement or acceleration in $\{q\}$, the associated lines in $[T_C]$ must be all zeros, and for the $[T_I]$ matrix, all zeros except one on the column related to this degree of freedom. In doing so, the degrees of freedom related to $\{q_I\}$ are the physical displacements of the structure where the displacement or acceleration needs to be imposed.

If the full finite element model of the structure is used, matrix $[T]_C$ is a $N \times N$ identity matrix from which the columns corresponding to the imposed dofs have been removed.

The strategy to choose matrix $[T_I]$ will differ for the case of imposed displacement and acceleration, as will be detailed below. Using (6.20), the equations of motion (4.2) become ($u(t) = 0$)

$$[M]\left([T_C]\left\{q_C''\right\} + [T_I]\left\{q_I''\right\}\right) + [C]\left([T_C]\left\{q_C'\right\} + [T_I]\left\{q_I'\right\}\right) + [K]\left([T_C]\{q_C\} + [T_I]\{q_I\}\right) = 0 \tag{6.21}$$

and after premultiplying by $[T_C]^T$ and putting the unknown displacements $\{q_C\}$ to the left-hand side of the equation, we get

$$[M_{CC}]\left\{q_C''\right\} + [C_{CC}]\left\{q_C'\right\} + [K_{CC}]\{q_C\} = -[T_C]^T\left([M][T_I]\left\{q_I''\right\} + [C][T_I]\left\{q_I'\right\} + [K][T_I]\{q_I\}\right) \tag{6.22}$$

where $[M_{CC}], [C_{CC}]$ and $[K_{CC}]$ are the mass, damping and stiffness matrices with fixed boundary conditions at the degrees of freedom corresponding to $\{q_I\}$. The subscript C stands for constrained, as the equations of motion in (6.22) correspond to a dynamic problem for which the degrees of freedom corresponding to the imposed motion are fixed, and the imposed displacement, velocity or accelerations correspond to an equivalent mechanical force vector.

As such, it is difficult to build a state-space model from these equations, whether for imposed displacement or acceleration, because only one of the two is imposed and the other is unknown.

6.3.1 State-space models with imposed displacement

For imposed displacement, the typical choice for $[T_I]$ consists in taking a matrix which is limited to the degrees of freedom where the displacement is imposed. All terms on the rows corresponding to the non-imposed degrees of freedom are therefore equal to 0. In this case, the forces acting on the structure are limited to the degrees of freedom which are coupled to the interface through the stiffness, mass, and damping matrices. If the mass matrix is diagonal, there is no such coupling, and if it is not, the coupling is very weak, so that the forces related to inertia can be neglected. This is also the case of the damping forces, as long as the damping is small. In such a case, the equations become

$$[M_{CC}]\left\{q_C''\right\} + [C_{CC}]\left\{q_C'\right\} + [K_{CC}]\left\{q_C\right\} = -\left[T_C\right]^T [K]\left[T_I\right]\left\{q_I\right\} \tag{6.23}$$

where now the forcing vector is only a function of the imposed displacements. With this formulation, a state-space model can be built using the approach detailed in section 6.1.2 , where normal modes of the system with the imposed dofs set to 0 and a static correction to $-\left[T_C\right]^T [K]\left[T_I\right]$ is added. The displacement at the imposed dofs can be directly recovered using the $[d]$ matrix if a sensor is defined at these location, using matrix $[T]_I$.

Note that in (6.23), there are as many inputs as there are degrees of freedom in $\{q_I\}$. This can be simplified to a single input using an expansion vector such that:

$$\{q_I\} = \{L\}\, q_0 \tag{6.24}$$

where q_0 is now a single input scalar displacement to the system, and the right-hand-side of equation (6.23) reduces to a single vector multiplied by the input displacement q_0. The number of necessary static responses to build the state-space model is therefore also reduced to one instead of the number of degrees of freedom in $\{q_I\}$.

The construction of a state-space model with imposed displacement is illustrated below on the same example of the concrete tower. The function `fe2ss` is used to build directly the I/O state-space model from the finite element after defining properly the input (horizontal imposed displacement $Uimp$) and the output (displacement at the top of the tower $d-top$).

In `d_piezo('TutoPzTowerSSUimp-s1')` , the mesh is built and a horizontal displacement is imposed at the basis of the tower.

Then in d_piezo('TutoPzTowerSSUimp-s2') , the transfer function between $Uimp$ and $d - top$ is computed using low-level calls using only the stiffness term in the excitation term, or both the stiffness and mass terms.

```
%% Step 2 : Reference method - exact solution + Inertial term neglected - full model
% Build matrices
[model,Case] = fe_case('assemble NoT -matdes 2 1 Case -SE',model) ;

% Full model
K0 = feutilb('tkt',Case.T,model.K); % Assemble matrices taking into account BCs
F1 = -Case.T'*model.K{2}*Case.TIn; % Loading due to imposed displacement
F2 = -Case.T'*model.K{1}*Case.TIn; % Inertial term
%
% compute response in freq domain
w=linspace(0,100,512);
for i=1:length(w)
    U0(:,i)=Case.T*((K0{2}*(1+0.02*1i)-w(i)^2*K0{1})\(F1-w(i)^2*F2));
    % Take into account mass term
    U1(:,i)=Case.T*((K0{2}*(1+0.02*1i)-w(i)^2*K0{1})\F1); % Stiffness term only
end
CTA = fe_c(model.DOF,21.01); u0 = CTA*U0; u1 = CTA*U1;

% Change output format to be compatible with iicom
C1=d_piezo('BuildC1',w'/(2*pi),u0.','d-top','Uimp'); C1.name='M and K';
C2=d_piezo('BuildC1',w'/(2*pi),u1.','d-top','Uimp'); C2.name='K only';
ci=iiplot;iicom(ci,'curveinit',{'curve',C1.name,C1;'curve',C2.name,C2});
iicom('submagpha')
d_piezo('setstyle',ci);
```

Figure 6.5 shows that the two curves match perfectly, meaning that the inertial term can indeed be neglected for the excitation, allowing to build an accurate modal state-space model with a static correction with the stiffness term only.

This is the approach used in SDT and illustrated in d_piezo('TutoPzTowerSSUimp-s3')

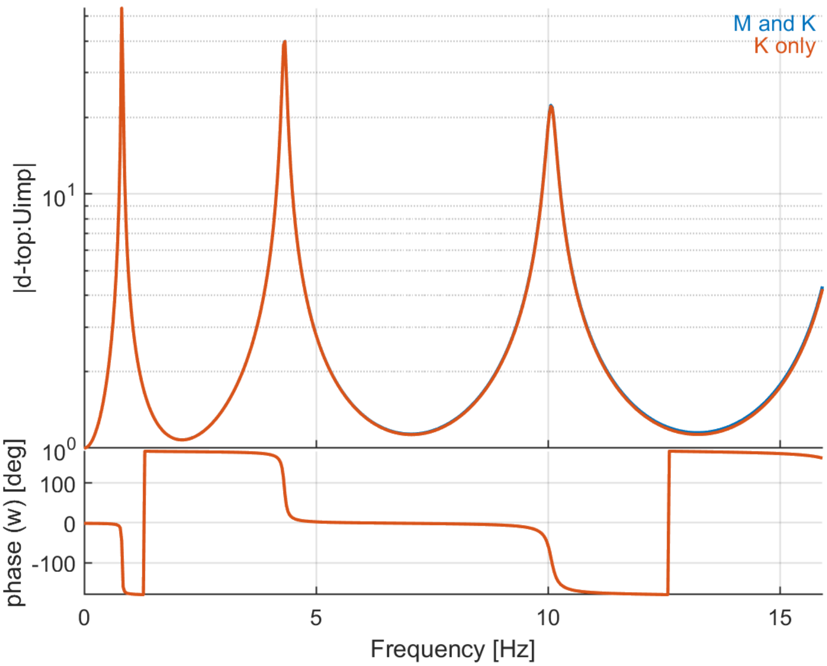


Figure 6.5: Transfer function between horizontal displacement at the bottom/top of the beam using the full model, the loading takes into account both the mass and stiffness terms, or only the stiffness term ($d_{top}/Uimp$)

```
%% Step 3 :  State-space model using SDT (reduced)
sys=fe2ss('free 5 5 0 -dterm',model);
C3=qbode(sys,w,'struct-lab');C3.name='fe2ss-5md+st'; C3.X{2}={'d-top'};
ci=iiplot;iicom(ci,'curveinit',{'curve',C1.name,C1;'curve',C3.name,C3});
iicom('submagpha')
d_piezo('setstyle',ci);
```

Figure 6.6 compares the frequency response functions used with the full model and direct computation, and using a state-space model with 5 modes and a static correction for the stiffness loading term. The match between the two is excellent.

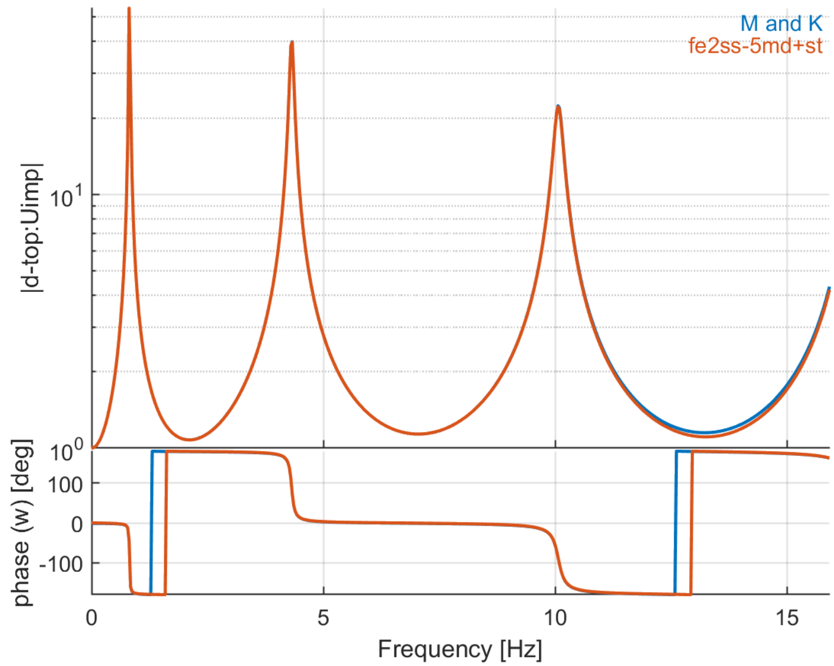


Figure 6.6: Transfer function between horizontal displacement at the bottom/top of the beam using the full model and the reduced state-space model with 5 modes, the loading takes into account only the stiffness term ($d_{top}/Uimp$)

6.3.2 State-space models with imposed acceleration

In the case of imposed acceleration, it is common to write the problem in terms of relative displacements with respect to the base motion, in which case matrix $[T_I]$ is not anymore limited to the imposed displacements. The approach consists in choosing $[T_I]$ as the static response to imposed displacements, so that it relates to all dofs in the structure. Let us assume that the dofs of the structure are rearranged so that the matrices can be decomposed in four blocks as previously :

$$\left[\tilde{K}\right] = \begin{bmatrix} [K_{II}] & [K_{IC}] \\ [K_{CI}] & [K_{CC}] \end{bmatrix} \tag{6.25}$$

where the index I corresponds to the imposed dofs. $[T_I]$ is then given by the static expansion of the imposed displacements:

$$[T_I] = \begin{bmatrix} I \\ -[K_{CC}]^{-1}[K_{CI}] \end{bmatrix} \{q_0\} \tag{6.26}$$

where $\{q_0\}$ represents the spatial shape of the imposed acceleration at the specific locations. Such a representation is general enough to take into account base excitation problems where the excitation of the base is uniform and in a single direction, such as a uniform translation of the base. In this case, $[T_I]$ represents the rigid body translation of the structure in the direction of the base motion, and $\{q_C\}$ is the relative motion of the structure with respect to the translation of the base. The general approach allows also to impose a combination of the six rigid-body modes of the structure. In this case, the quantity that is withdrawn from the global motion in order to represent the relative motion is different at each location in the structure in the case where rotational rigid body motions are present in the base excitation.

Lastly, the approach also allows to take into account base excitations which induce a deformation of the basis, in which case $[T_I]$ is not a rigid body motion but the static expansion of the imposed displacements on all other dofs of the structure (which reduces to rigid body motions if the imposed displacement is a translation and/or rotation of the base).

With this type of decomposition, in the case where $[T_I]$ is a linear combination of rigid-body motions, the equations of motion (6.21) reduce to:

$$[M_{CC}]\left\{q_C{''}\right\}+[C_{CC}]\left\{q_C'\right\}+[K_{CC}]\left\{q_C\right\}=-\left[T_C\right]^T[M]\left[T_I\right]\left\{q_I''\right\} \tag{6.27}$$

because $[T_I]\{q_I\}$ represents a combination of rigid body modes so that $[K][T_I]\{q_I\}=0$. For damping models where the damping matrix is proportional to stiffness, the damping term is also equal to zero, otherwise it can be neglected when damping is small.

When $[T_I]$ is not a combination of rigid body modes (i.e. when the imposed acceleration induces deformation of the base), the stiffness term is given by

$$[T_C]^T[K]\begin{bmatrix} I \\ -[K_{CC}]^{-1}[K_{CI}] \end{bmatrix}\{q_I\}=\Big(-[K_{CC}][K_{CC}]^{-1}[K_{CI}]+[K_{CI}]\Big)\{q_I\}=0 \tag{6.28}$$

so that it is also zero (see also in [5]).
With this simplification, it is possible to build a reduced state-space model where the force vector is related to the inertial term and the imposed acceleration only. Note however that in this case, the state-space model solves for relative displacements, and that it is not straightforward to obtain the absolute displacements (if needed), as it would require to know the base displacement, and here it is the acceleration that is imposed (so one cannot use the $[d]$ matrix here to add the base displacement).

A possible approach is to solve for the absolute displacements by modifying the mass, stiffness and damping matrices as:

$$\left[\hat{M}\right]=\begin{bmatrix} I & 0 \\ 0 & [M_{CC}] \end{bmatrix} \tag{6.29}$$

$$\left[\hat{K}\right]=\begin{bmatrix} 0 & 0 \\ 0 & [K_{CC}] \end{bmatrix} \tag{6.30}$$

$$\left[\hat{C}\right] = \begin{bmatrix} 0 & 0 \\ 0 & [C_{CC}] \end{bmatrix} \tag{6.31}$$

and the displacement and forcing vectors as

$$\{q\} = \begin{Bmatrix} q_I \\ q_C \end{Bmatrix} \tag{6.32}$$

$$\{F\} = \begin{Bmatrix} \{q_0''\} \\ -[T_C]^T [M] [T_I] \{q_0''\} \end{Bmatrix} \tag{6.33}$$

where $\{q_0''\}$ is the vector of imposed accelerations.

A state-space representation can be built by projecting these equations in the subspace of the constrained modeshapes which are the solution of

$$\left(-\omega_j^2 [M_{CC}] + [K_{CC}]\right) \{\phi_j\} = \{0\} \tag{6.34}$$

The reduced basis (keeping NR modes) noted $\left[\hat{T_C}\right]$ is given by :

$$\left[\hat{T_C}\right] = [\phi_{1 \ldots NR}] \tag{6.35}$$

so that

$$\{q_C\} = \left[\hat{T_C}\right] \{\hat{q_C}\} \tag{6.36}$$

which leads to the reduced matrices (with mass normalized modeshapes)

$$\left[\hat{M_{CC}}\right] = \left[\hat{T_C}\right]^T [M_{CC}] \left[\hat{T_C}\right] = I \quad , \quad \left[\hat{K_{CC}}\right] = \left[\hat{T_C}\right]^T [K_{CC}] \left[\hat{T_C}\right] = \left[\backslash \omega_j^2 \backslash\right] \tag{6.37}$$

The state-space representation is then given by:

$$\begin{Bmatrix} \dot{q}_I \\ \dot{\hat{q_C}} \\ \ddot{q}_I \\ \ddot{\hat{q_C}} \end{Bmatrix} = \begin{bmatrix} [0] & [I] \\ \begin{bmatrix} 0 \\ -\left[\hat{T_C}\right]^T [K] \left[T_I \ \ \hat{T_C}\right] \end{bmatrix} & \begin{bmatrix} 0 \\ -\left[\hat{T_C}\right]^T [C] \left[T_I \ \ \hat{T_C}\right] \end{bmatrix} \end{bmatrix} \begin{Bmatrix} q_I \\ \hat{q_C} \\ \dot{q}_I \\ \dot{\hat{q_C}} \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & I \\ \left[\hat{T_C}\right]^T b & \left[\hat{T_C}\right]^T [T_I] \end{bmatrix} \begin{Bmatrix} u_F \\ \ddot{q}_0 \end{Bmatrix} \tag{6.38}$$

$$\{y\} = \left[c\,[T_I] \quad c\left[\hat{T_C}\right] \quad 0 \quad 0\right] \begin{Bmatrix} q_I \\ \hat{q_C} \\ \dot{q}_I \\ \dot{\hat{q_C}} \end{Bmatrix} + [0] \begin{Bmatrix} u_F \\ q_0'' \end{Bmatrix}$$

With this choice of $[T]_I$, the submatrix $-\left[\hat{T_C}\right]^T [K] [T_I] = 0$ and the states are fully decoupled. The method can be used with the alternative representation where T_I is taken as the matrix limited to the imposed displacement (as for the imposed displacement formulation, in which case the submatrix $-\left[\hat{T_C}\right]^T [K] [T_I]$ is non-zero and remains in the left-hand-side because $\{q_I\}$ is unknown. The disadvantage of this formulation is the fact that the state-variables remain coupled in the modal domain, and that it is necessary to augment the modal basis with static responses to $[M] [T_I] \{q_0''\}$.

The general method implemented in SDT to build state-space models with imposed displacements corresponds to the second case where the states remain coupled, it requires the combination of the use of fe_reduc and nor2ss function and cannot be performed directly with fe2ss. An illustration is given in the script below.

In d_piezo('TutoPzTowerSSAimp-s1') , the model is built and the actuator and sensors are defined.

The in d_piezo('TutoPzTowerSSAimp-s2') , the first calculation is explicit and performed in the frequency domain, with low level calss. The problem is solved using relative displacements.

```
%% Step 2 : Regular method with RHS M*Tin (relative displacement)
% --------- full model
[model,Case] = fe_case('assemble NoT -matdes 2 1 Case -SE',model) ;

K0 = feutilb('tkt',Case.T,model.K); % Assemble matrices taking into account BCs
% Compute TIn as static response to imposed displacement
TIn=fe_simul('static',model); TIn=TIn.def;
% Loading due to imposed acceleration
F = -Case.T'*model.K{1}*TIn;

% compute response in freq domain (relative displacement)
w=linspace(1,100,512);
for i=1:length(w)
    U0r(:,i)=Case.T*((K0{2}*(1+0.02*1i)-w(i)^2*K0{1})\F);
end
Uimp=1./(-w.^2); % Imposed displacement for a unit imposed acceleration

% Absolute displacement
for i=1:length(w)
U0(:,i)=U0r(:,i)+Uimp(i)*TIn;
```

```
end

CTA = fe_c(model.DOF,21.01); u0 = CTA*U0; u0r = CTA*U0r;

% Change output format to be compatible with iicom
C0=d_piezo('BuildC1',w'/(2*pi),u0.','d-top','Aimp'); C0.name='Full';
C1=d_piezo('BuildC1',w'/(2*pi),u0r.','dr-top','Aimp'); C1.name='Full';
```

In d_piezo('TutoPzTowerSSAimp-s3') , we build a state-space model with fe2ss using the adequate load based on imposed accelerations. The output of the state-space model is then a relative displacement.

```
%% Step 3 - State-space with SDT(relative) using a DofLoad
model2=d_piezo('Meshtower');
model2=fe_case(model2,'FixDof','Clamped',[1.01 1.06]); % Block all interface dofs
model2=fe_case(model2,'Remove','F-top'); % Remove point force
[model2,Case2] = fe_case('assemble NoT -matdes 2 1 Case -SE',model2) ;

% Load when using relative displacements
SET.DOF=model2.DOF; SET.def=Case2.T*F;
model2=fe_case(model2,'DofLoad','Aimp',SET); %

% state-space model
sysr=fe2ss('free 5 5 0 -dterm',model2);
```

Figure 6.7 shows the comparison between the response ($d_{r,top}/Aimp$) of the full model computed in the frequency domain, and of the reduced state-space model using 5 modes and a static correction to the applied load (built from the acceleration and mass matrix).

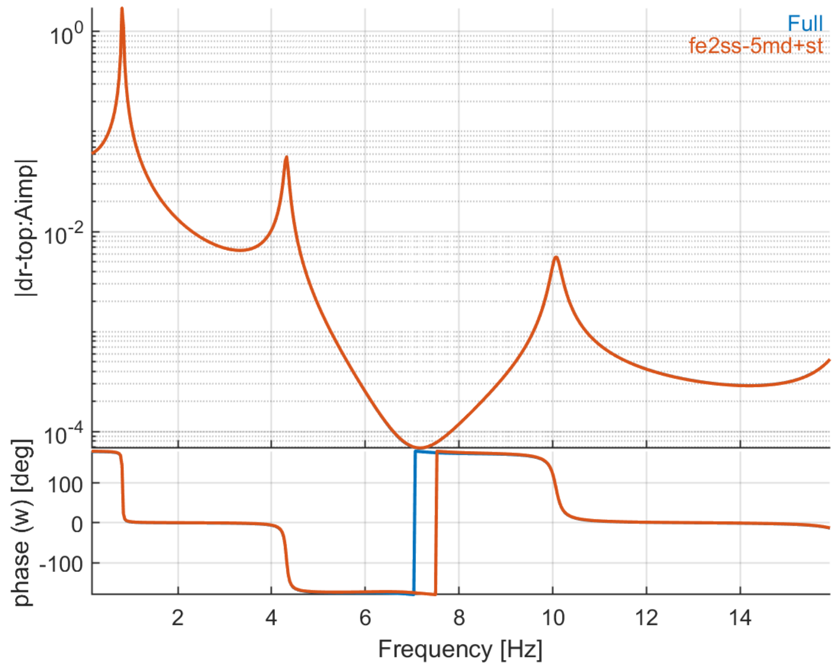


Figure 6.7: comparison between the response (relative displacement at the top of the tower divided by imposed acceleration) of the full model computed in the frequency domain, and of the reduced state-space model using 5 modes and a static correction ($d_{r,top}/Aimp$)

In d_piezo('TutoPzTowerSSAimp-s4') , the approach used in SDT to build a state-space model allowing to obtain directly the absolute displacement is illustrated.

```
%% Step 4 : state-space model for absolute displacements - Aimp

% This is a CB basis which is renormalized (so free BCs and rigid body mode)
% TR2.data is needed for nor2ss hence the normalization.
TR2 = fe2ss('craigbampton 5 5 -basis',model);
sysu= nor2ss(TR2,model) ;
```

Figure 6.8 shows that the state-space model obtained with this approach in SDT gives a response very close to the full model computed in the frequency domain.

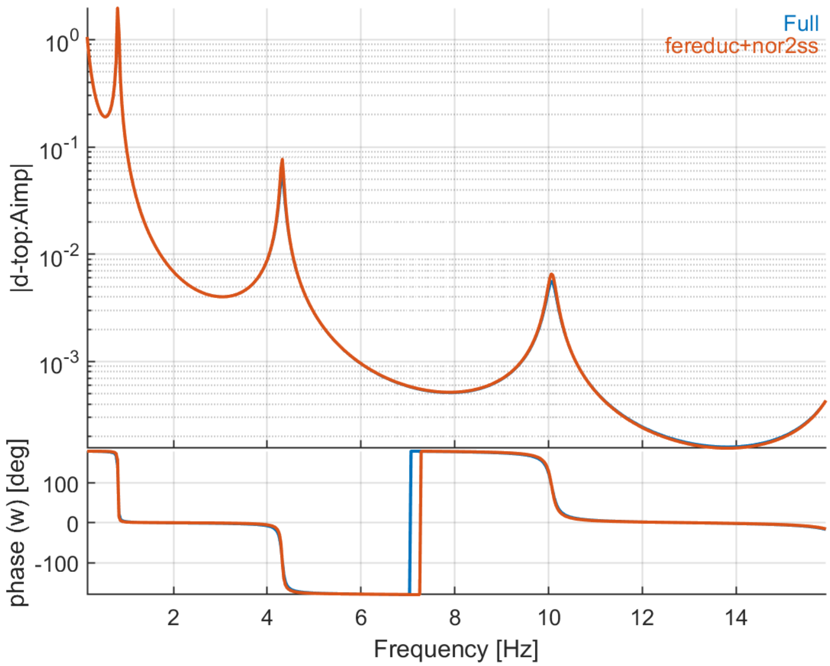


Figure 6.8: Transfer function between horizontal acceleration at the bottom and the absolute displacement at the top of the beam and the imposed horizontal acceleration at the bottom - comparison of the full model (freq domain computation) and the reduced state-space model with 5 modes and static correction ($d_{top}/Aimp$)

6.3.3 Model reduction and state-space models for piezoelectric structures

When building reduced or state-space models to allow faster simulation, the validity of the reduction is based on assumptions on bandwidth, which drive modal truncation, and considered loads which lead to static correction vectors.

Modes of interest are associated with boundary conditions in the absence of excitation. When using piezoelectric actuators driven in voltage mode, for the electric part, these are given by potential set to zero (grounded or shorted electrodes) and enforced by actuators (defined as `DofSet` in SDT in the case of voltage actuators) which in the absence of excitation is the same as shorting.

Excitation can be mechanical $\{F_{mech}\}$, charge on free electric potential DOF $\{Q\}_{In}$ and imposed voltage $\{V\}_{In}$. One thus seeks to solve a problem of the form

$$\begin{bmatrix} Z_{qq}(s) & Z_{qV} \\ Z_{Vq} & Z_{VV} \end{bmatrix} \begin{Bmatrix} q \\ V \end{Bmatrix} = \begin{Bmatrix} F_{mech} \\ Q_{In} \end{Bmatrix} - \begin{bmatrix} Z_{qV_{In}} \\ Z_{VV_{In}} \end{bmatrix} \{V_{In}\} \tag{6.39}$$

The imposed voltage $\{V\}_{In}$ is enforced using a `DofSet` in SDT and is therefore analogous to an imposed displacement. The general form of the loads given above can be simplified due to the fact that there are no coupling terms in the mass matrix between the mechanical and electrical DOFs for full models of piezoelectric structures (see (4.6)), and the fact that the capacitance matrix $[K_{VV}]$ is diagonal. The problem is then reduced to

$$\begin{bmatrix} Z_{qq}(s) & Z_{qV} \\ Z_{Vq} & Z_{VV} \end{bmatrix} \begin{Bmatrix} q \\ V \end{Bmatrix} = \begin{Bmatrix} F_{mech} \\ Q_{In} \end{Bmatrix} - \begin{bmatrix} K_{qV_{In}} \\ 0 \end{bmatrix} \{V_{In}\} \tag{6.40}$$

Using the classical modal synthesis approach (implemented as `fe2ss('free')`), one builds a Ritz basis combining modes with grounded electrodes ($V_{In} = 0$), static responses to mechanical and charge loads and static response to enforced potential:

$$\begin{Bmatrix} q \\ V \\ V_{In} \end{Bmatrix} = \left[\begin{bmatrix} \phi_q \\ \phi_V \\ 0 \end{bmatrix} \begin{bmatrix} Z(0)^{-1} \begin{Bmatrix} F_{mech} \\ Q_{In} \end{Bmatrix} \\ 0 \end{bmatrix} \begin{bmatrix} Z(0)^{-1} \begin{bmatrix} K_{qV_{In}} \\ 0 \end{bmatrix} \\ I \end{bmatrix} \right] \begin{Bmatrix} q_{mode} \\ q_{stat} \\ V_{In} \end{Bmatrix} \tag{6.41}$$

In this basis, one notes that the static response associated with enforced potential V_{In} does not verify the boundary condition of interest for the state-space model where $V_{In} = 0$. Since it is desirable to retain the modes with this boundary condition as the first vectors of basis (6.41) and to include static correction as additional vectors, the strategy used here is to rewrite reduction as

$$\{q\} = \left[\begin{bmatrix} \phi_q \\ \phi_V \\ 0 \end{bmatrix} \begin{bmatrix} Z(0)^{-1} \begin{bmatrix} F_{mech} & K_{qV_{In}} \\ Q_{In} & 0 \end{bmatrix} \\ 0 \end{bmatrix} \right] \{q_R\} + \begin{Bmatrix} 0 \\ 0 \\ V_{In} \end{Bmatrix} \tag{6.42}$$

where the response associated with reduced DOFs q_R verifies $V_{In} = 0$ and the total response is found by adding the enforced potential on the voltage DOF only. The presence of this contribution corresponds to a `d` term in state-space models. The usual SDT default is to include it as a residual vector as shown in (6.41), but to retain the shorted boundary conditions, form (6.42) is prefered.

6.3.4 Example of a cantilever beam with 4 piezoelectric patches

The example of the cantilever beam with 4 piezoelectric transducers detailed in section 4.6.2 is considered again. A state-space model is built using the first 10 modes and static corrections discussed above, and the dynamic response is compared to the full-model response.

In `d_piezo('TutoPzPlate4pztSS-s1')` , the model is built and the actuators and sensors defined.

Then in `d_piezo('TutoPzPlate4pztSS-s2')` , the transfer functions between 'V-Act', defined on the first piezoelectric patch, and , 'Q-Act', 'QS1-3', defined as charge sensors on patches 1 to 4, as well as 'Tip' (tip displacement) are computed using the full model with `fesimul` and the reduced state-space model built with `fe2ss`.

Figures 6.9 and 6.10 show that there is an excellent match between the responses computed with the full model and the state-space model with 10 modes and static corrections.

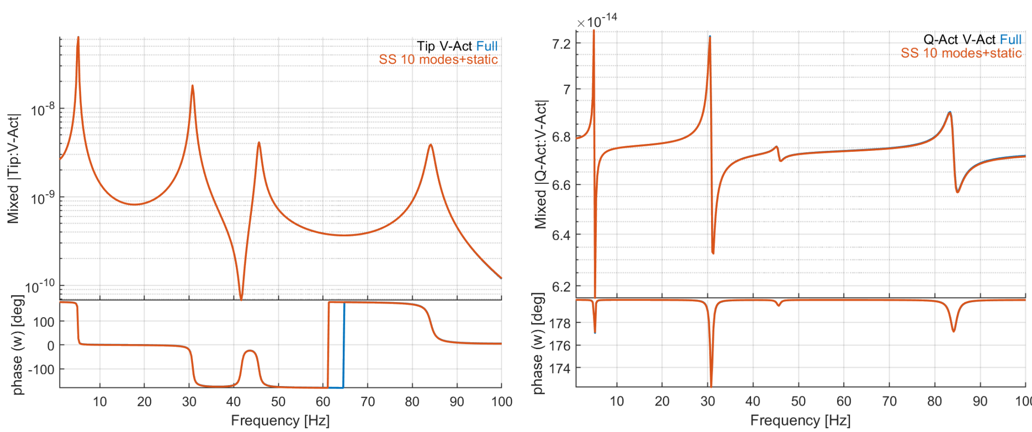


Figure 6.9: Open-loop transfer function between V-Act and tip displacement (left), Q-Act sensor (right), comparison between, full model and reduced state-space model (10 modes + static corrections - dterm)

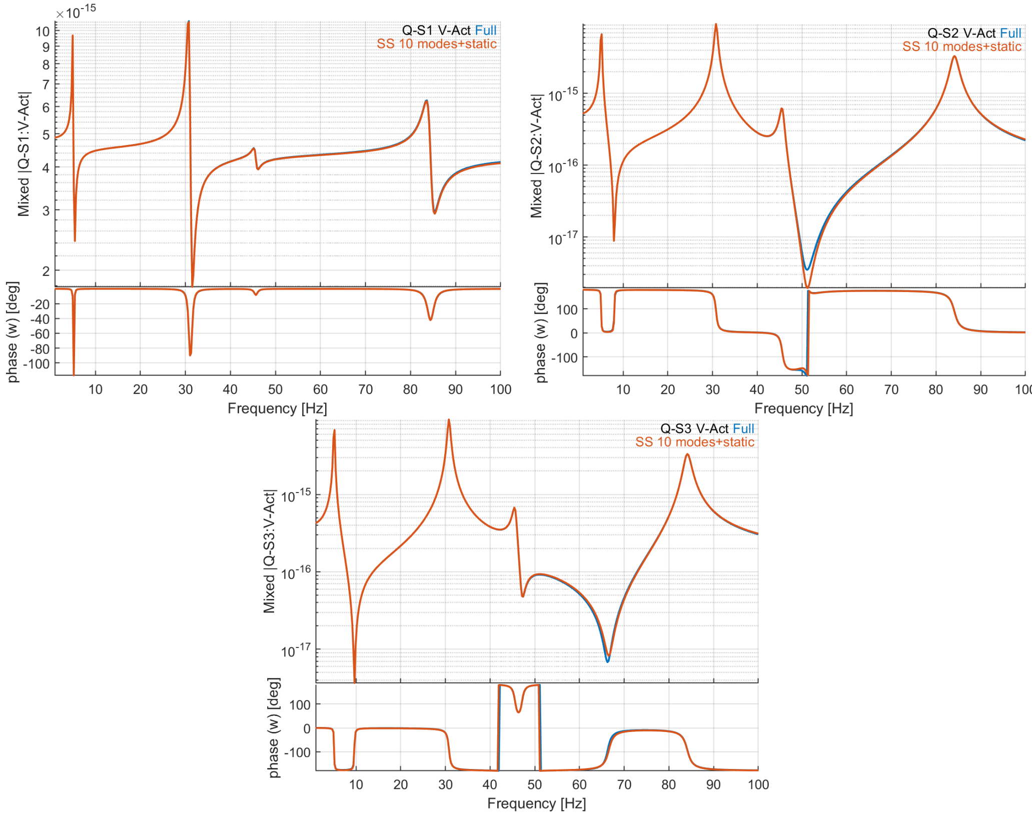


Figure 6.10: Open-loop transfer function between V-Act and Q-S1 sensor (top-left),Q-S2 sensor (top-right), Q-S3 sensor (bottom), comparison between, full model and reduced state-space model (10 modes + static corrections - dterm)

In the next script, we illustrate the use of sensors and actuators combinations.

In `d_piezo('TutoPzPlate4pztSSComb-s1')` , we build the mesh and define actuator combinations: the first and second patch are actuated together with an opposite voltage to induce pure bending, and the static response is computed (Figure 6.11).

Then in `d_piezo('TutoPzPlate4pztSSComb-s2')` , we define two sensors, consisting in charge combination with opposite signs for nodes 1057 and 1058 and voltage combination with opposite signs for the same nodes (Figure 6.12 shows the case of charge combination).

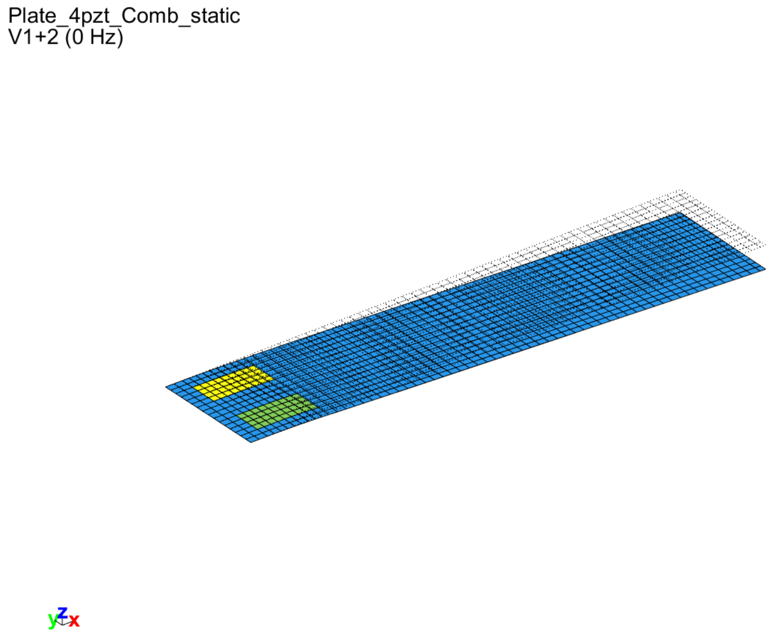


Figure 6.11: Static deflection due to combination of actuators to induce pure bending

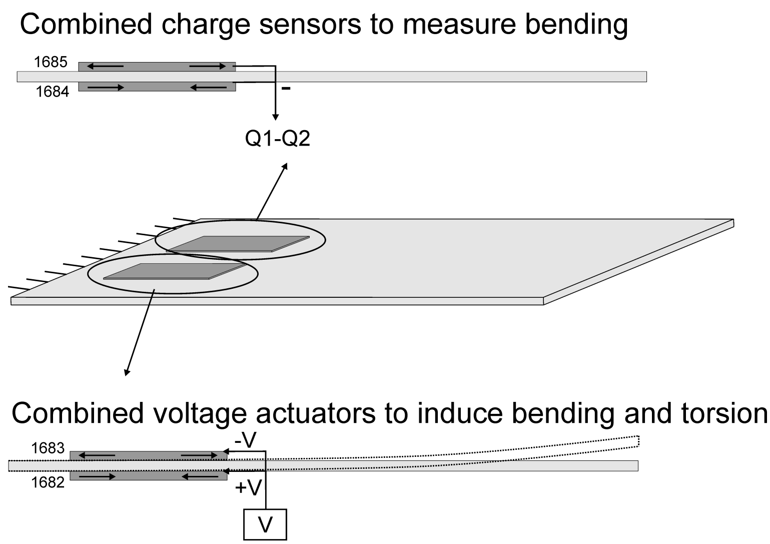


Figure 6.12: Example of combination of charge sensors and voltage actuators to measure bending

```
%% Step 2 - Define sensor combinations
% Combined charge output (SC electrodes) % difference of charge 1684-1685
r1=struct('cta',[1 -1],'DOF',edofs(3:4),'name','QS3+4');
model=p_piezo('ElectrodeSensQ',model,r1);
% Combined voltage output (OC electrodes) % difference of voltage 1684-1685
r1=struct('cta',[1 -1],'DOF',edofs(3:4),'name','VS3+4');
model=fe_case(model,'SensDof',r1.name,r1);
model=fe_case(model,'pcond','Piezo','d_piezo(''Pcond 1e8'')');
```

By default, the electrodes are in 'open-circuit' condition for sensors, except if the sensor is also used as voltage actuator which corresponds to a 'short-circuit' condition. Therefore, as the voltage is left 'free' on nodes 1057 and 1058, the charge is zero and the combination will also be zero. If we wish to use the patches as charge sensors, we need to short-circuit the electrodes, which will result in a zero voltage and in a measurable charge.

This is illustrated in `d_piezo('TutoPzPlate4pztSSComb-s3')` by computing the response in both configurations (open-circuit by default, and short-circuiting the electrodes for nodes 1057 and 1058):

The FRF for the combination of charge sensors is not exactly zero but has a negligible value in the 'open-circuit' condition, while the voltage combination is equal to zero in the 'short-circuit' condition. Charge sensing in the short-circuit condition and voltage sensing in the open-circuit condition are compared by scaling the two FRFs to the static response ($f = 0Hz$) and the result is shown on Figure 6.13. The FRFs are very similar but the eigenfrequencies are slightly lower in the case of charge sensing. This is due to the well-known fact that open-circuit always leads to a stiffening of the piezoelectric material. The effect on the natural frequency is however not very strong due to the small size of the piezoelectric patches with respect to the full plate.

The stiffening effect due to the presence of an electric field in the piezoelectric material when the electrodes are in the open-circuit condition is a consequence of the piezoelectric coupling. One can look at the level of this piezoelectric coupling by comparing the modal frequencies with the electrodes in open and short-circuit conditions.
This is done in `d_piezo('TutoPzPlate4pztSSComb-s5')`

```
%% Step 5 - Compute OC and SC frequencies
model=d_piezo('MeshULBplate -cantilever');
% Open circuit : do nothing on electrodes
d1=fe_eig(model,[5 20 1e3]);
% Short circuit : fix all electric DOFs
DOF=p_piezo('electrodeDOF.*',model);
d2=fe_eig(fe_case(model,'FixDof','SC',DOF),[5 20 1e3]);
```

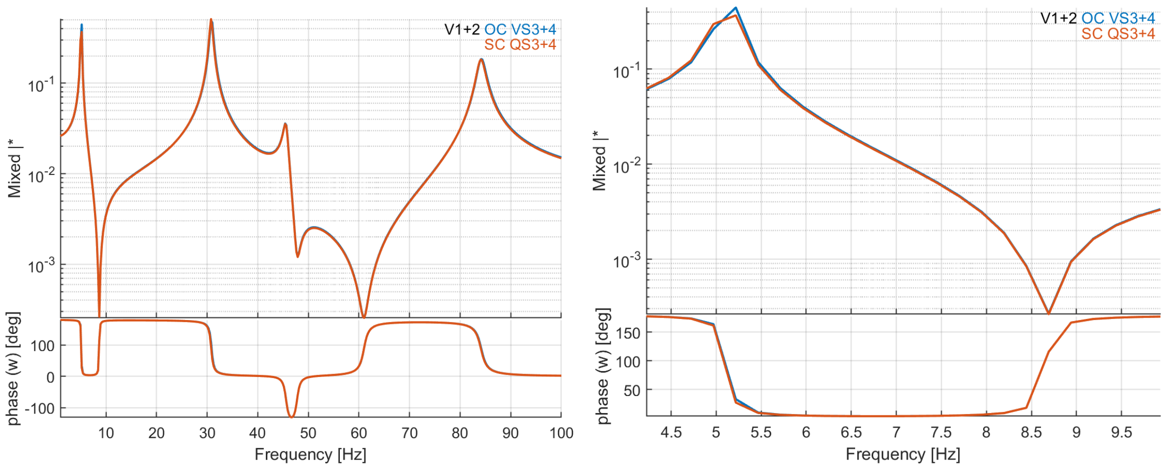


Figure 6.13: Comparison of FRFs (scaled to the static response) for voltage (green) and charge (blue) sensing. Zoom on the third eigenfrequency (right)

```
r1=[d1.data(1:end)./d2.data(1:end)];
gf=figure;plot(r1,'*','linewidth',2);axis tight; set(gca,'Fontsize',15)
xlabel('Mode number');ylabel('f_{OC}/f_{SC}');
```

Figure 6.14 shows the ratio of the eigenfrequencies in the open-circuit vs short-circuited conditions. The difference depends on the mode number but is always lower than 1%. Higher stiffening effects occur when more of the strain energy is contained in the piezoelectric elements, and the coupling factor is higher.

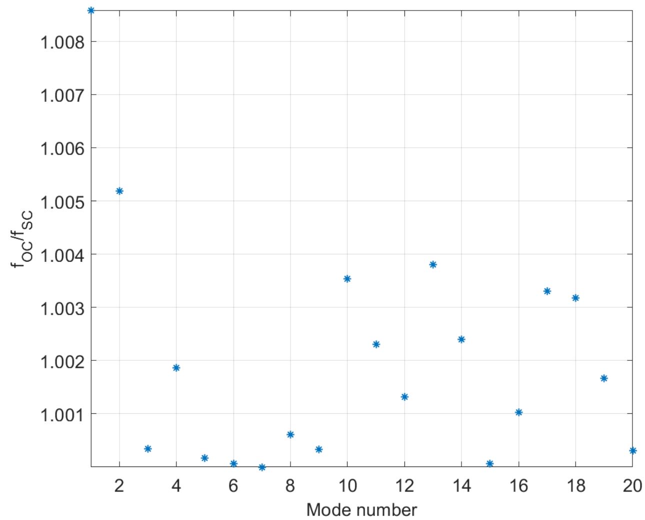


Figure 6.14: Ratio of the natural frequencies of modes 1 to 20 in open-circuit vs short-circuit conditions illustrating the stiffening of the piezoelectric material in the open-circuit condition

6.4 State-space models and Craig-Bampton model reduction

For coupled problems linked to model substructuring, it is traditional to state the problem in terms of imposed displacements rather than loads. Assuming that the imposed displacements correspond to DOFs, one seeks solutions of problems of the form

$$\begin{bmatrix} Z_{II} & Z_{IC} \\ Z_{CI} & Z_{CC} \end{bmatrix} \begin{Bmatrix} q_I \\ q_C \end{Bmatrix} = \begin{Bmatrix} R_I \\ 0 \end{Bmatrix} \tag{6.43}$$

where the displacement q_I are given and the reaction forces R_I are non-zero. The exact response to an imposed harmonic displacement q_I is given by

$$\{q\} = \begin{bmatrix} I \\ -Z_{CC}^{-1} Z_{CI} \end{bmatrix} \{q_I\} \tag{6.44}$$

The first level of approximation is to use a quasistatic evaluation of this response , that is to use $Z(0) = [K]$. Model reduction on this basis is known as **static or Guyan condensation** [6].

This reduction does not fulfill the requirement of validity over a given frequency range. Craig and Bampton [7] thus complemented the static reduction basis by **fixed interface modes** : normal modes of the structure with the imposed boundary condition $q_I = 0$. These modes correspond to singularities in Z_{CC} so their inclusion in the reduction basis allows a direct control of the range over which the reduced model gives a good approximation of the dynamic response.

The Craig-Bampton reduction basis takes the special form

$$\left\{ \begin{array}{c} q_I \\ q_C \end{array} \right\} = \left[\begin{array}{cc} I & 0 \\ -K_{CC}^{-1} K_{CI} & \phi_C \end{array} \right] \{q_R\} \tag{6.45}$$

where the fact that the additional fixed interface modes have zero components on the interface DOFs is very useful to allow direct coupling of various component models. fe_reduc provides a solver that directly computes the Craig-Bampton reduction basis.

A major reason of the popularity of the Craig-Bampton (CB) reduction basis is the fact that the interface DOFs q_I appear explicitly in the generalized DOF vector q_R, and is a major reason why the use of Craig-Bampton reduction methods is very popular in the industry.

The major finite elements software such as Nastran, Abaqus or Ansys all propose a CB reduction. In the industry, it is a common practice to share models via CB reduced matrices, therefore keeping the details of the geometry and material properties confidential. Generally, the degrees of freedom which are kept in the model are the essential ones, i.e. the ones where forces are applied, or where the magnitude of displacement, velocity or acceleration needs to be assessed, as well as interface degrees of freedom if the structure needs to be coupled to one or several others. The question we address in the next section is how to build an accurate state-space model based on these reduced matrices only.

6.4.1 State-space models with imposed displacements using CB matrices

As detailed in section 6.3.1 , when dealing with a full FE model, the imposed displacement can be replaced by a force equal to $-[T_C][K][T_I]\{q_I\}$, neglecting the term related to acceleration. It turns out that this assumption is not valid after a CB reduction.

It can be easily understood if we consider that only the imposed degrees of freedom are retained in the CB basis. In that case, $[T_C][K][T_I] = [K_{CI}]$ which, according to (6.28) is equal to zero, would result in no load applied to the system. In this specific case, the inertial term is the most important one, while the stiffness term is zero, hence it is not possible to build a state-space model based on the standard approach presented before.

In `d_piezo('TutoPzTowerSSUimpCB-s1')` , we use low level calls to make an explicit computation of the response after CB reduction, using either the inertia, the stiffness of both terms for the loading.

```
% See full example as MATLAB code in d_piezo('ScriptTutoPzTowerSSUimpCB')
d_piezo('DefineStyles');

%% Step 1 : Build Model, CB reduction, explicit response
model=d_piezo('meshtower');
model=fe_case(model,'FixDof','Clamped',[1.06]); % Leave x free for imposed displ
model=fe_case(model,'Remove','F-top'); % Remove point force
model=fe_case(model,'DOFSet','UImp',[1.01]); % Set imposed displ

%Full model matrices
[model,Case] = fe_case('assemble NoT -matdes 2 1 Case -SE',model) ;
K0 = feutilb('tkt',Case.T,model.K); % Full-model

% CB matrices and T matrix
CB    = fe_reduc('craigbampton 5 5',model); TR = CB.TR; KCB=CB.K;

% Build loads full and reduced models
F = -Case.T'*model.K{2}*Case.TIn;
F1=-CB.K{2}(2:end,1);
F2=-CB.K{1}(2:end,1);

w=linspace(1,100,512);

for i=1:length(w)
 U0(:,i)=Case.T*((K0{2}*(1+0.02*1i)-w(i)^2*K0{1})\F); % full model
 U1r(:,i)=((KCB{2}(2:end,2:end)*(1+0.02*1i)-w(i)^2*KCB{1}(2:end,2:end)) ...
  \(F1-w(i)^2*F2)); % CB stiffness and inertia
 U2r(:,i)=((KCB{2}(2:end,2:end)*(1+0.02*1i)-w(i)^2*KCB{1}(2:end,2:end)) ...
    \(F1)); % CB stiffness
 U3r(:,i)=((KCB{2}(2:end,2:end)*(1+0.02*1i)-w(i)^2*KCB{1}(2:end,2:end))...
    \(-w(i)^2*F2)); %CB inertia
end

CTA = fe_c(model.DOF,21.01); u0 = CTA*U0;
U1=TR.def*[ones(1,length(w)); U1r]; u1 = CTA*U1;
U2=TR.def*[ones(1,length(w)); U2r]; u2 = CTA*U2;
U3=TR.def*[ones(1,length(w)); U3r]; u3 = CTA*U3;

% Change output format to be compatible with iicom
```

```
C1=d_piezo('BuildC1',w'/(2*pi),u0.','d-top','Uimp'); C1.name='full model';
C2=d_piezo('BuildC1',w'/(2*pi),u1.','d-top','Uimp'); C2.name='CB K and M';
C3=d_piezo('BuildC1',w'/(2*pi),u2.','d-top','Uimp'); C3.name='CB K';
C4=d_piezo('BuildC1',w'/(2*pi),u3.','d-top','Uimp'); C4.name='CB M';
ci=iiplot;iicom(ci,'curveinit',{'curve',C1.name,C1;'curve',C2.name,C2; ...
'curve',C3.name,C3;'curve',C4.name,C4});
iicom('submagpha')
d_piezo('setstyle',ci);
```

Figure 6.15 illustrates clearly the fact that the forcing term related to acceleration is the most important one to represent the response.

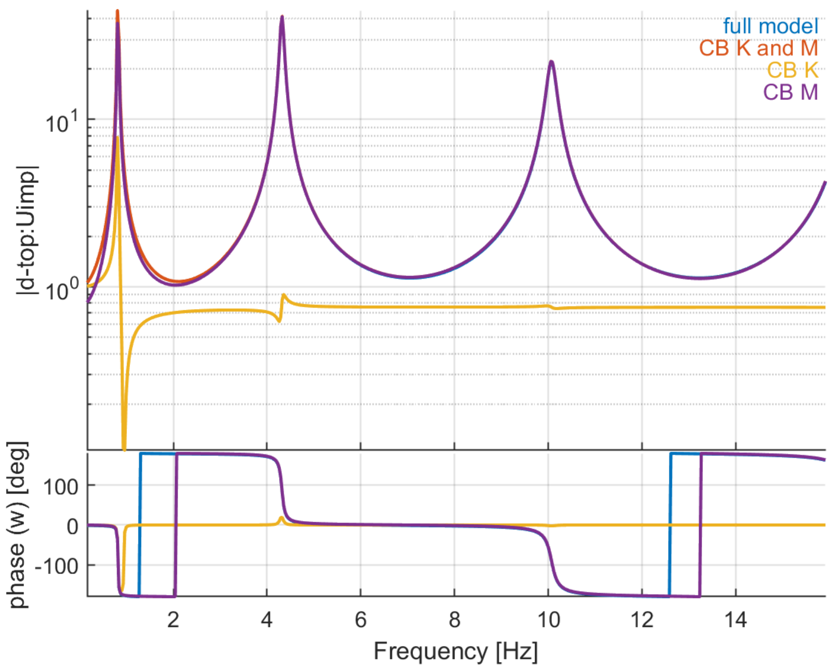


Figure 6.15: Transfer function between horizontal displacement at the bottom and the absolute displacement at the top of the beam

In practice however, the DOFs retained with Craig-Bampton are not limited to the imposed displacements. As an illustrative example, we consider that the translation DOF at the top of the tower is also kept in the CB reduction.

In `d_piezo('TutoPzTowerSSUimpCB-s2')` , based on these reduced matrices, a state-space model is built with fe2ss and the response is compared to the full model.

```
%% Step 2: keep bottom and top translation in the CB basis

% Ss from full model - 5 modes + static
sys0=fe2ss('free 5 5 0 -dterm',model);
C0=qbode(sys0,w,'struct-lab');C0.name='fe2ss'; C0.X{2}={'d-top'}

% Top and bottom DOF to be kept in CB reduction
SET.DOF=[1.01; 21.01]; SET.def=eye(2);
model=fe_case(model,'DOFSet','UImp',SET);

% Build CB reduced model
model = stack_set(model,'info','EigOpt',[5 5 0]);
SE1= fe_reduc('CraigBampton -SE -matdes 2 1 3 4 ',model); %
% To keep only the matrices
SE1 = rmfield(SE1,{'il','pl','Stack','TR','mdof'}) ;
SE1.Node=feutil('getnode',SE1,unique(fix(SE1.DOF)));SE1.Elt=[];

% Define input/output
SE1=fe_case(SE1,'SensDOF','d-top',21.01);
SE1=fe_case(SE1,'DOFSet','UImp',[1.01]); %

% Build state-space based on CB reduced model
sys=fe2ss('free 5 5 0 -dterm',SE1);
```

Figure 6.16 shows that using fe2ss after a CB reduction leads to a very bad reduced state-space representation of the dynamics of the beam with imposed displacement.

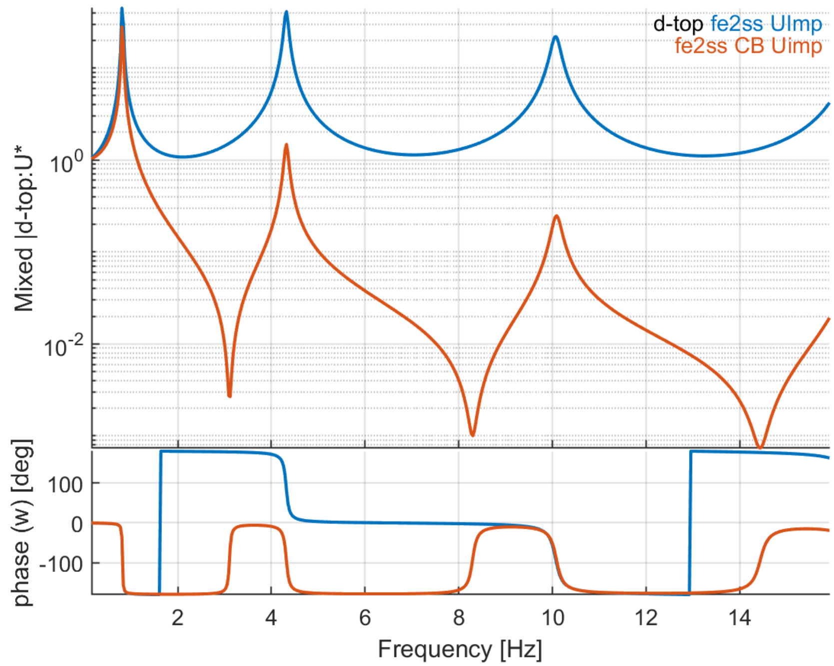


Figure 6.16: Transfer function between horizontal displacement at the bottom and the absolute displacement at the top of the beam

An alternative is to use a transformation of the CB mass and stiffness matrices in order to come back to a situation where the inertial term in the excitation is zero, and only the stiffness term is non-zero, so that `fe2ss` can be applied to build the state-space model. The transformation is applied so that the retained DOFs are untouched (so that they represent physical DOFs and can be used to simply apply force and measure output quantities), as proposed in [8]. Raze's transform takes the form:

$$\{\tilde{q_C}\} = -\left[M_{CC}\right]^{-1}\left[M_{CI}\right]\{q_I\} + \{q_c\} \tag{6.46}$$

Therefore the matrices can be transformed with the following projection matrix

$$\left[\tilde{T}\right] = \begin{bmatrix} I & 0 \\ -\left[M_{CC}\right]^{-1}\left[M_{CI}\right] & I \end{bmatrix} \tag{6.47}$$

It is straightforward to show that after applying the transformation, matrix $\left[\tilde{M_{CI}}\right] = 0$ (hence the option name `noMCI` in `fe_reduc`), so that now only the stiffness term appears in the excitation term. After transforming the matrices, `fe2ss` can be used to build an accurate reduced state-space model.

In In d_piezo('TutoPzTowerSSUimpCB-s3') , we first perform the transformation manually with low-level calls and replace the matrices in the reduced model, then use fe_reduc('CraigBampton -noMCI') for a direct application.

```
%% Step 3 : Apply Raze transform before making the state-space model

% Transform the initial CB matrices
KCB=SE1.K;
N=size(KCB{1},1);
T1=[ eye(2) zeros(2,N-2) ; - KCB{1}(3:end,3:end)\KCB{1}(3:end,1:2) eye(N-2)];

Mr=T1'*(KCB{1})*T1;
Kr=T1'*(KCB{2})*T1;

% Replace matrices in the CB model
SE2=SE1;
SE2.K{1}=Mr;
SE2.K{2}=Kr;

sys2=fe2ss('free 5 5 0 -dterm',SE2);

% Make CB matrices with Raze's transform directly -noMCI
SE3= fe_reduc('CraigBampton -SE -matdes 2 1 3 4 -noMCI',model);

% To keep only the matrices
SE3 = rmfield(SE3,{'il','pl','Stack','TR','mdof'}) ;
SE3.Node=feutil('getnode',SE3,unique(fix(SE3.DOF)));SE3.Elt=[];

% Define input/output
SE3=fe_case(SE3,'SensDOF','d-top',21.01);
SE3=fe_case(SE3,'DOFSet','UImp',[1.01]); %

% Build state-space based on CB reduced model
sys3=fe2ss('free 5 5 0 -dterm',SE3);
```

Figure 6.17 shows that using fe2ss after a CB and Raze's transformation given by (6.47) leads to an accurate model in the frequency band of interest

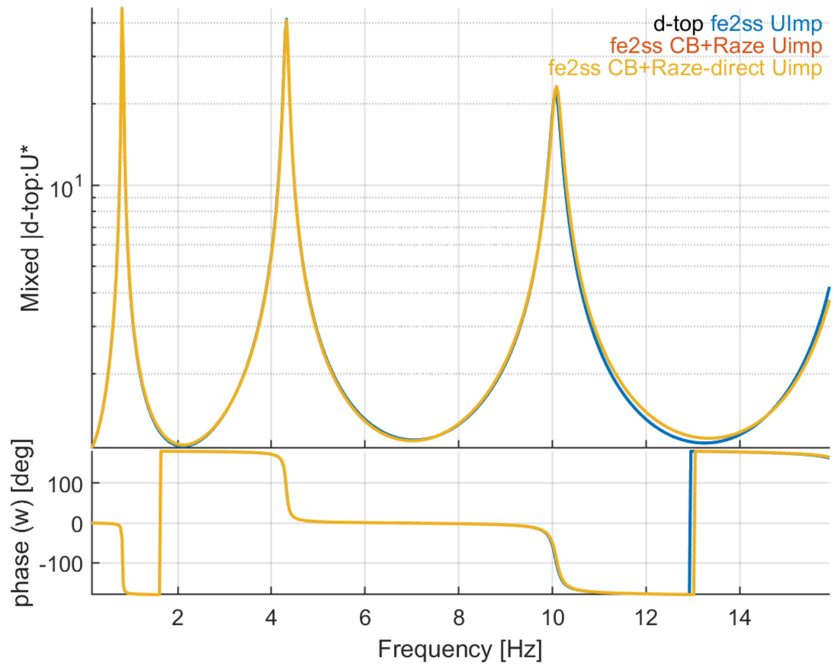


Figure 6.17: Transfer function between horizontal displacement at the bottom and the absolute displacement at the top of the beam

6.4.2 State-space models with imposed accelerations

The case of imposed acceleration is much simpler to treat as `fe2ss` can be used directly after CB reduction in the case of relative displacement. If one is interested in absolute displacement, `nor2ss` can be applied after computing the modeshapes based on the reduced CB matrices and having orthonormalized them. The first script illustrates the construction of a state-space model for relative displacement, using `fe2ss` and the definition of an equivalent load.

In `d_piezo('TutoPzTowerSSAimpCB-s1')` , the reference solution is computed using explicit computations with low-level calls.

Then in `d_piezo('TutoPzTowerSSAimpCB-s2')` , a CB reduction is performed keeping the displacement in the x-direction for the nodes at the bottom and at the top of the tower and 5 modes.

```
%% Step 1: reference solution -direct computation

% Build model
```

```
model=d_piezo('meshtower');
model=fe_case(model,'FixDof','Clamped',[1.06]);
% Leave x free for imposed displ
model=fe_case(model,'Remove','F-top'); % Remove point force

% Initial matrices without imposed displacements
[model0,Case0] = fe_case('assemble NoT -matdes 2 1 Case -SE',model) ;
K0 = feutilb('tkt',Case0.T,model0.K); % Full-model 41x41

% Matrices with Block displ at basis
model=fe_case(model,'DOFSet','UImp',[1.01]); %

% Full model matrices
[model,Case] = fe_case('assemble NoT -matdes 2 1 Case -SE',model) ;
Kcc = feutilb('tkt',Case.T,model.K); % Blocked at basis 40x40

% Put manually unitary values to be exact
TIn=zeros(40,1); TIn(1:2:end)=1;

% The load is built after applying the boundary condition to the matrices.
F = -(Kcc{1}*TIn);

w=linspace(0,100,512);
eta=0.02;

% Reference solution - full model
for i=1:length(w)
    U0(:,i)=Case.T*((Kcc{2}*(1+eta*1i)-w(i)^2*Kcc{1})\F);
end

CTA = fe_c(model.DOF,21.01); u0 = CTA*U0;
% Change output format to be compatible with iicom
C0=d_piezo('BuildC1',w'/(2*pi),u0.','dr-top','Aimp'); C0.name='Full';


%% Step 2 : state-space model based on CB reduction

% CB reduction on two DOFs
SET.DOF=[1.01; 21.01]; SET.def=eye(2);
model=fe_case(model,'DOFSet','UImp',SET);
```

```
% Build CB reduced model
model = stack_set(model,'info','EigOpt',[5 5 0]); % No mass shift!
SE1= fe_reduc('CraigBampton -SE -matdes 2 1 3 4 -bset ',model); %

SE0=SE1; % Save superelement model

% To keep only the matrices
SE1 = rmfield(SE1,{'il','pl','Stack','TR','mdof'}) ;
SE1.Node=feutil('getnode',SE1,unique(fix(SE1.DOF)));SE1.Elt=[];

% Force vector using CB matrices
KCB=SE1.K;
TInCB= [1; -KCB{2}(2:end,2:end)\KCB{2}(2:end,1)];
FCB= -KCB{1}*TInCB; % From CB matrices only


% Block 1.01 and define a DofLoad and a new observation matrix instead
SE1=fe_case(SE1,'FixDOF','BC',1.01); % BC for relative displacement
SET.DOF=SE1.DOF(2:end); SET.def= FCB(2:end);
SE1=fe_case(SE1,'DOFLoad','UImp',SET); % Equivalent load
SE1=fe_case(SE1,'SensDOF','Output',SET.DOF(1));

sys=fe2ss('free 5 5 0 -dterm',SE1);
```

Figure 6.18 shows that using fe2ss with applied load leads to an accurate model in the frequency band of interest

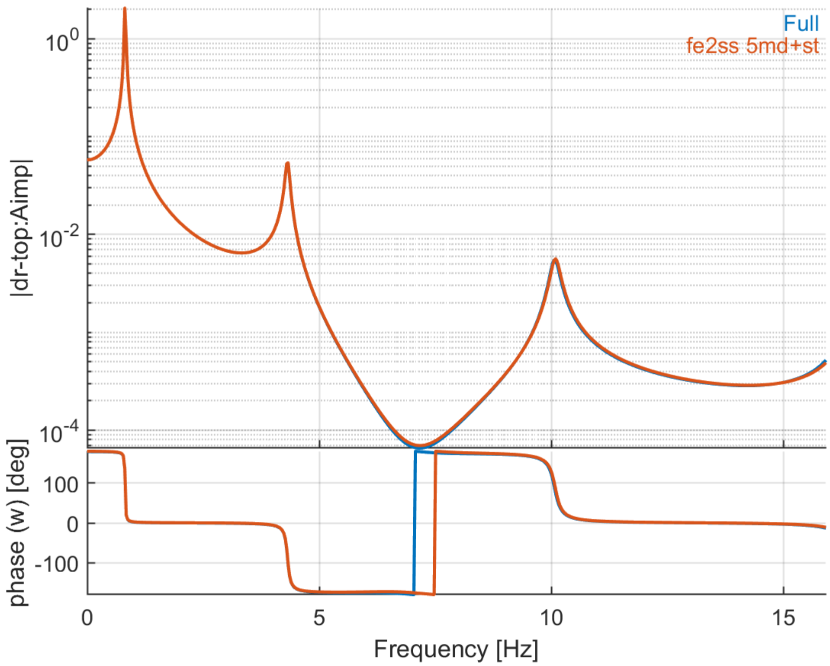


Figure 6.18: Transfer function between horizontal acceleration at the bottom and the relative displacement at the top of the beam

In d_piezo('TutoPzTowerSSAimpCB-s3') , the state-space model with an output in absolute displacements is built using nor2ss and the response iscompared to the reference solution.

```
%% Step 3: State-space model with absolute displacement

SE2=SE0; % Initial model without BCs
TR2=fe_eig(SE2,[5 7 0]); % Compute all7  modes of reduced model

% Imposed acceleration at base
SET.DOF=[1.01]; SET.def=eye(1); %
SE2=fe_case(SE2,'DOFSet','UImp',SET); % Impose acc at bottom

% build state-space model with nor2ss
sys2= nor2ss(TR2,SE2) ;
```

Figure 6.19 shows that using nor2ss with imposed acceleration leads to an accurate model in the frequency band of interest

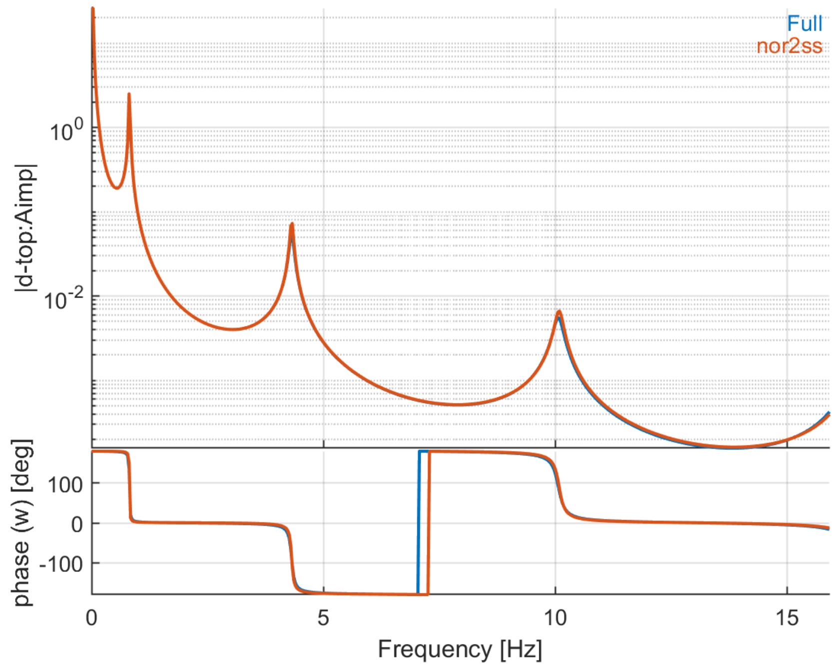


Figure 6.19: Transfer function between horizontal acceleration at the bottom and the absolute displacement at the top of the beam

6.4.3 State-space models with imposed voltage (piezoelectric actuators)

We are considering again the aluminum plate with 4 piezoelectric patches shown in Figure 4.19 and with the material properties in Table 4.1. In `d_piezo('TutoPzPlate4pztSSCB-s1')` , a reference state-space model is first built with `fe2ss` using 5 modes in the reduced basis and the static correction linked to the voltage actuator.

In `d_piezo('TutoPzPlate4pztSSCB-s2')` , a Craig-Bampton model reduction is performed choosing the 4 electrical DOFs corresponding to each patch and the displacement sensor at the tip of the beam as reference DOFs. The reduced matrices are used to build a super-element, and the response is computed as for the reference using directly `fe2ss`.

Figure 6.20 shows the errors introduced due to the fact that there is a mass coupling term after CB.

```
%% Step 2 - CB reduction
model=fe_case(model,'DOFSet','In',[nd+.03; i1+.21]); %
```

```
% Build CB matrices
model = stack_set(model,'info','EigOpt',[5 10 0]);
SE1= fe_reduc('CraigBampton -SE -matdes 2 1 3 4 -bset',model); %
% To keep only the matrices
SE1 = rmfield(SE1,{'il','pl','Stack','TR'}) ;
SE1.Node=feutil('getnode',SE1,unique(fix(SE1.DOF)));SE1.Elt=[];

% Define node names
SE1=sdth.urn('nmap.Node.set',SE1, ...
    {'Tip',nd;'Act',i1(1);'S1',i1(2);'S2',i1(3);'S3',i1(4)});

% Define input/output
SE1=fe_case(SE1,'SensDof','Sensors',scell);
SE1=fe_case(SE1,'DofSet','In',rmfield(dIn,'Elt')); % Elt defined for display
SE1=stack_set(SE1,'info','DefaultZeta',1e-2);

% Build state-space model based on reduced CB matrices
s1=fe2ss('free 5 10 0 -dterm',SE1);
```

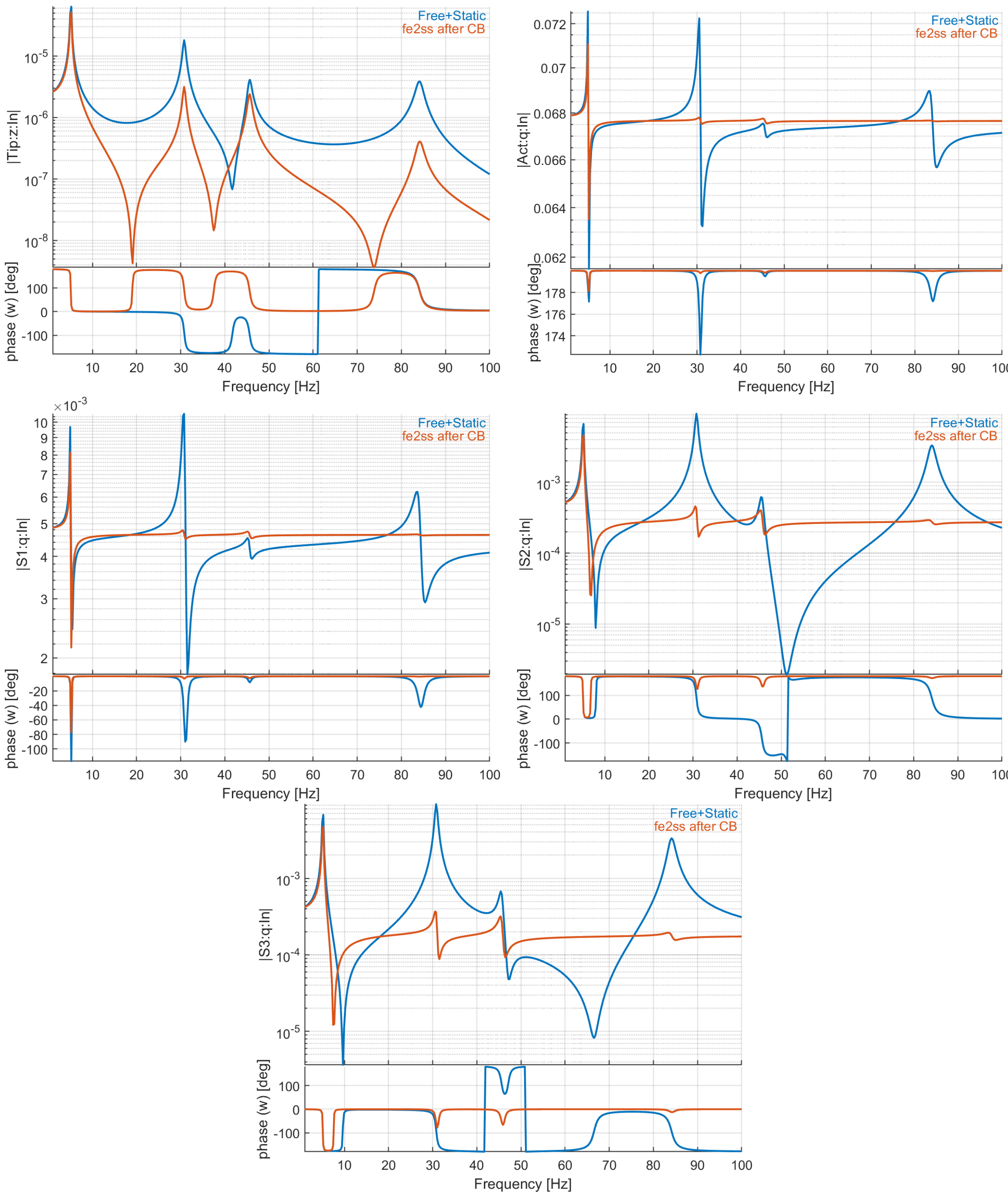


Figure 6.20: Tip displacement and charges on the 4 patches due to voltage actuator on patch 1. The reference solution is computed with a direct modal reduction (10 modes) with static correction, while the second curve represents the response after using CB reduction (10 modes + 5 retained DOFs) before building the state-space model. The second curve is wrong due to the coupling in the mass matrix introduced by the CB reduction

In In `d_piezo('TutoPzPlate4pztSSCB-s3')` , we use Raze's transform to remove the coupling terms in the mass matrix (-noMCI option in `fe_reduc`).

Figure 6.21 shows the result when making the transform to remove mass coupling term after CB. The match with the referene solution is excellent.

```
%% Step 3: Now with Raze's transform, using -noMCI option in fe_reduc
SE2= fe_reduc('CraigBampton -SE -matdes 2 1 3 4 -noMCI',model); % Removes M-coupling
% To keep only the matrices
SE2 = rmfield(SE2,{'il','pl','Stack','TR'}) ;
SE2.Node=feutil('getnode',SE2,unique(fix(SE2.DOF)));SE2.Elt=[];

%Define node names
SE2=sdth.urn('nmap.Node.set',SE2, ...
    {'Tip',nd;'Act',i1(1);'S1',i1(2);'S2',i1(3);'S3',i1(4)});
% Define input/output
SE2=fe_case(SE2,'SensDof','Sensors',scell);
SE2=fe_case(SE2,'DofSet','In',rmfield(dIn,'Elt')); % Elt defined for display
SE2=stack_set(SE2,'info','DefaultZeta',1e-2);

% Build state-space model based on reduced CB matrices
s2=fe2ss('free 5 10 0 -dterm',SE2);
```

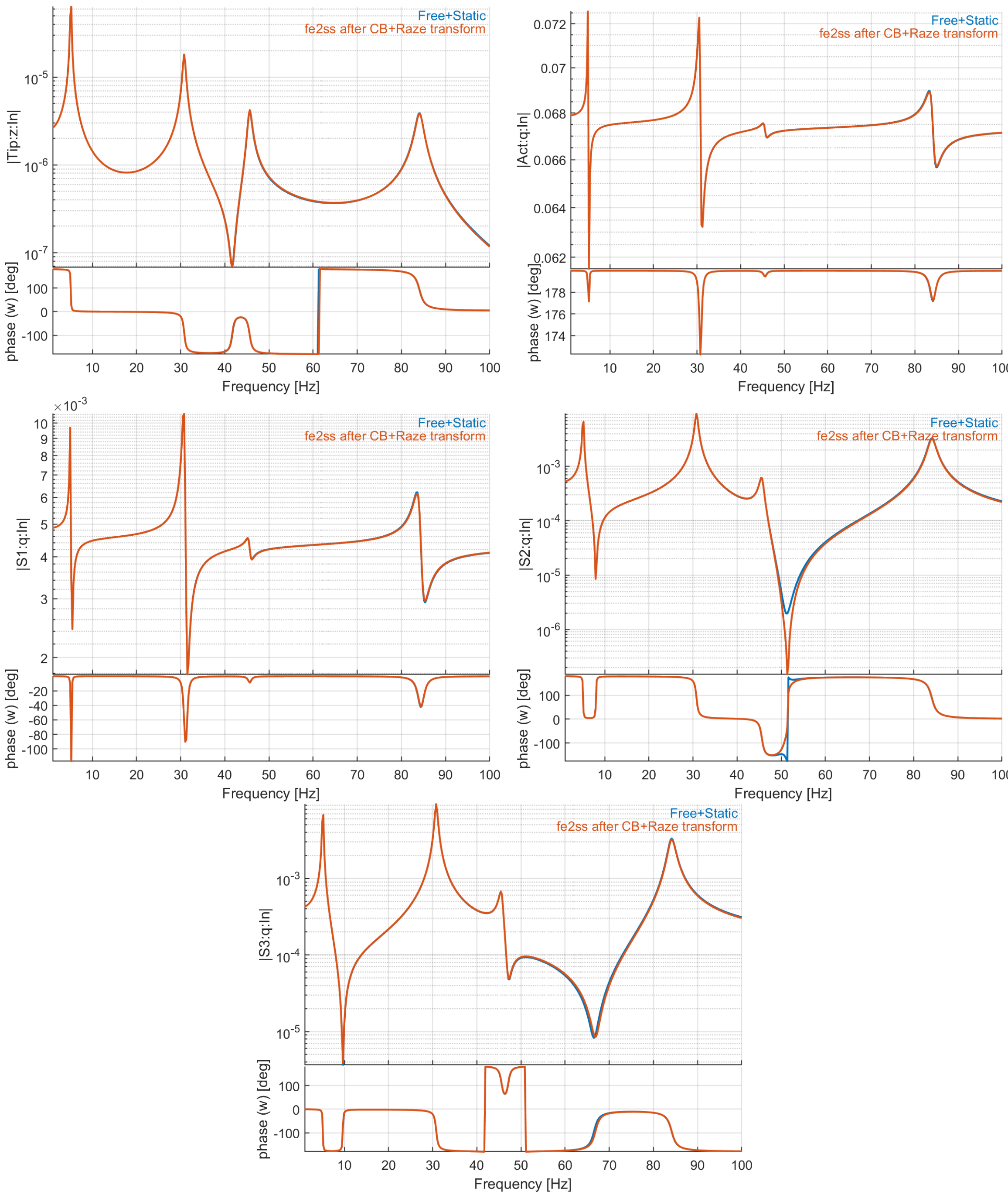


Figure 6.21: Tip displacement and charge on patches due to voltage actuator on patch 1. The reference solution is computed with a direct modal reduction (10 modes) with static correction, while the second curve represents the response after using CB reduction and Raze's transform (10 modes + 5 retained DOFs) before building the state-space model. There is an excellent match between the two curves

7

Composite piezoelectric actuators and sensors

Contents

7.1 Working principle of piezocomposite transducers

PZT ceramics are commonly used due to their good actuation capability and very wide bandwidth. The major drawbacks of these ceramics are their brittle nature, and the fact that they cannot be easily attached to curved structures. In order to overcome these drawbacks, several packaged PZT composites have appeared on the market. A typical piezocomposite transducer is made of an active layer sandwiched between two soft thin encapsulating layers (Figure 7.1).

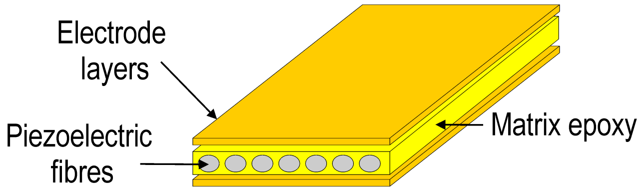


Figure 7.1: General layout of a piezoelectric composite transducer

The packaging plays two different roles : (i) applying prestress to the active layer in order to avoid cracks, and (ii) bringing the electric field to the active layer through the use of a specific surface electrode pattern. Due to the difficulty to ensure contact between cylindrical fibers and the electrodes, rectangular fibers have been developed, leading to the 'Macro Fiber Composite' transducers initially developed by the NASA [9] and currently manufactured by the company Smart Material (http://www.smart-material.com). As this type of transducer is widely used in the research community, this section shows how to integrate MFCs in piezoelectric plate models in SDT. Note however that all types of piezocomposites can be modeled, providing sufficient material data is available, which is rarely the case, as highlighted in the following for the case of MFCs.

Both d_{31} and d_{33} MFCs have been developed. The d_{31} MFCs are based on the same concept as the bulk ceramic patches where poling is made through the thickness (Figure 7.2a). The d_{33} MFCs are aimed at exploiting the d_{33} actuation/sensing mode. This can be done by aligning the poling direction and the electric field with the fiber direction. The solution generally adopted is to use inter-digitated electrodes (IDE) as shown in Figure 7.2b, which results in curved electric field lines, with the majority of the electric field aligned in the fiber direction. The general layout of both types of MFCs is represented in Figure 7.3. In the Smart Material documentation, the d_{33}-type is referred to as $P1$-type elongator (because $d_{33} > 0$) and the d_{31}-type is referred to as $P2$-type contractor (because $d_{31} < 0$). Piezoelectric plate elements implemented in SDT are based on the hypothesis that the poling direction is through the thickness, which is suitable for the $P2$-type MFCs, but not for the $P1$-types. It is possible however with a simple analogy to model a $P1$-type MFC using the piezoelectric plate elements of SDT. The analogy is based on the equality of free in-plane strain of the transducer due to an applied voltage and capacitance. This requires two steps. The first one is

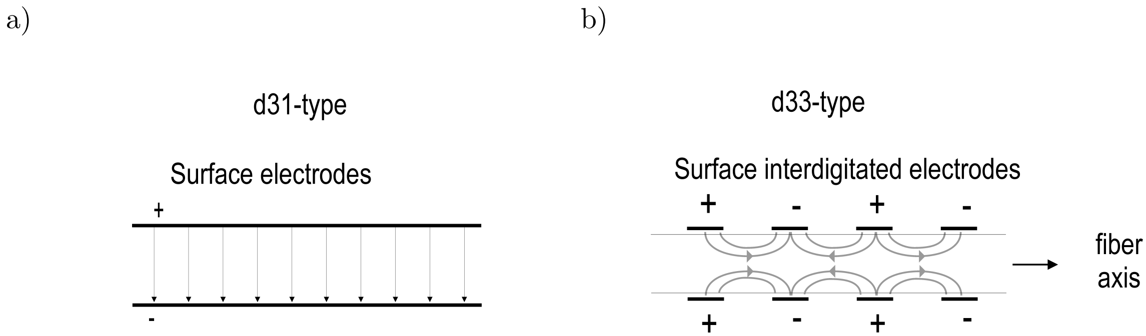


Figure 7.2: Electric fields in a) d_{31} and b) d_{33} piezocomposites

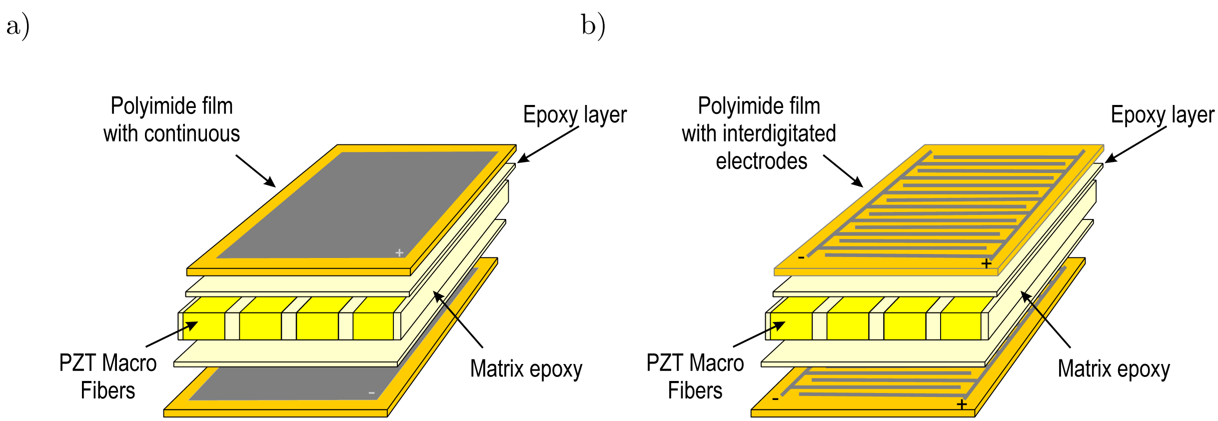


Figure 7.3: General layout of a) d_{31}-type MFCs and b) d_{33}-type MFCs

to replace the curved electric field lines by a uniform field aligned with the poling direction, equal to $E = V/p$ where p is the distance between the fingers of the interdigitated electrodes (Figure 7.4).

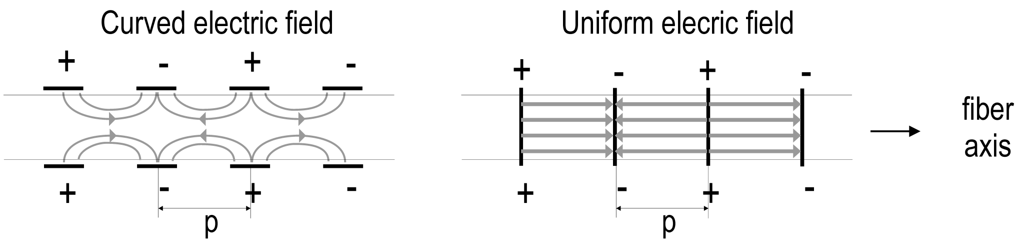


Figure 7.4: The curved electric field can be replaced by an equivalent electric field $E = V/p$

The second step is to express the equivalence in terms of free strains taking into account the difference of local axes (this is because direction 3 is the poling direction which is different in both cases, see Figure 7.5).

$$\begin{array}{l} S_1|_{P2} = d_{31}|_{P2}\frac{V}{h} = S_3|_{P1} = d_{33}|_{P1}\frac{V}{p} \longrightarrow d_{31}|_{P2} = d_{33}|_{P1}\frac{h}{p} \\ S_2|_{P2} = d_{32}|_{P2}\frac{V}{h} = S_2|_{P1} = d_{32}|_{P1}\frac{V}{p} \longrightarrow d_{32}|_{P2} = d_{32}|_{P1}\frac{h}{p} \end{array} \tag{7.1}$$

As the thickness of the patch h is generally different from the distance between the fingers of the inter-digitated electrodes p, a h/p factor must be used. For MFCs, $h = 0.180mm$ and $p = 0.500mm$ so that $h/p = 0.36$. The PZT material used in MFCs has properties similar to the Ceramtec P502 material described in Section section 2.3 . If we assume that the active layer of the MFC is made of a bulk piezoceramic of that type, the equivalent d_{31} is given by :

$$S_1|_{P2} = d_{33}|_{P1}\frac{h}{p} = 440\frac{0.18}{0.5}(pC/N) = 158.4(pC/N) \tag{7.2}$$

This value is lower than the d_{31} coefficient of the bulk ceramic (185 pC/N). This shows that although the d_{33} coefficient is larger, the spacing of the fingers of IDE reduces the equivalent strain per Volt (ppm/V). This spacing cannot however be made much smaller as the part of the electric field aligned with the plane of the actuator would be significantly reduced. These findings are confirmed by the tabulated values of free strain per volt (ppm/V) in the fiber direction given in the datasheet of MFCs. Note that because the limiting value for actuation is the electric field and not the voltage, the $P1$-type MFCs have a much higher maximum voltage limit (1500 V) than the $P2$-type MFCs (360 V), leading to the possibility to achieve higher free strain, but at the cost of very high voltage values.

Similarly, the dielectric permittivity must be adapted to model $P1$-type MFCs using an equivalent $P2$-type MFC. This is done by expressing the equality of the capacitance:

$$C|_{P2} = \frac{\varepsilon_{33}^T|_{P2}\, b\, p}{h} = C|_{P1} = \frac{\varepsilon_{33}^T|_{P1}\, b\, h}{p} \longrightarrow \varepsilon_{33}^T|_{P2} = \varepsilon_{33}^T|_{P1}\left(\frac{h}{p}\right)^2 \tag{7.3}$$

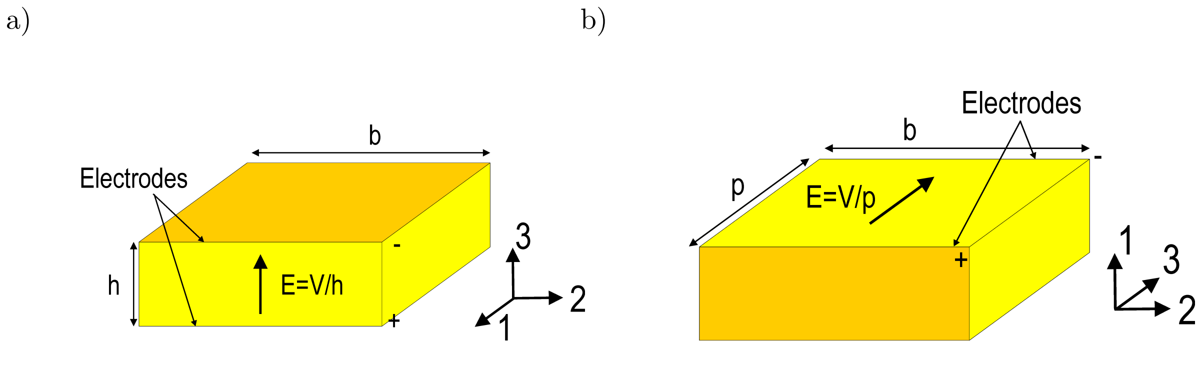


Figure 7.5: Electric fields and local axes used to model a $P1$-type MFC (b) with an equivalent $P2$-type MFC (a)

The fact that piezoelectric fibers are mixed with an epoxy matrix introduces some orthotropy both at the mechanical and the piezoelectric levels. This means that all the parameters of the compliance matrix of an orthotropic material (2.29) must be identified, together with all piezoelectric coefficients (2.14). As we are integrating these transducers in plate structures, and assuming that the poling is in the direction of the thickness with electrodes on top and bottom of the piezoelectric layers, the compliance matrix reduces to:

$$\left[s^E\right] = \begin{bmatrix} \frac{1}{E_x} & \frac{-\nu_{yx}}{E_y} & 0 & 0 & 0 \\ \frac{-\nu_{xy}}{E_x} & \frac{1}{E_y} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{G_{yz}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xz}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix} \tag{7.4}$$

with $\frac{\nu_{yx}}{E_y} = \frac{\nu_{xy}}{E_x}$, the matrix of piezoelectric coefficients to:

$$[d] = \begin{bmatrix} d_{31} & d_{32} & 0 & 0 & 0 \end{bmatrix}; \tag{7.5}$$

and the matrix of dielectric permittivities to a scalar:

$$\left[\varepsilon^T\right] = \begin{bmatrix} \varepsilon^T_{33} \end{bmatrix} \tag{7.6}$$

In order to model such orthotropic transducers, it is therefore necessary to have access to 6 mechanical properties $E_x, E_y, \nu_{xy}, G_{xy}, G_{xz}, G_{yz}$, two piezoelectric coefficients d_{31}, d_{32}, and one dielectric constant ε^T_{33}. In the following, direction x will be replaced by L standing for 'longitudinal' (i.e. in the fiber direction) and y by T for 'transverse' (i.e. perpendicular to the fiber direction).

As there are no established and standardised techniques for testing piezocomposite transducers and identifying their full set of properties, it is common to find only a limited set of these coefficients in the datasheet of manufacturers. In addition, when such properties are given, they are measured on the full packaged piezocomposite transducer, which makes it difficult to translate them to the properties of each layer without making strong assumptions. The strategy adopted in this tutorial is to consider that an MFC is made of 5 layers (Figure 7.6). The electrode layer is slightly orthotropic due to the presence of the copper, but this effect can be neglected as it does not influence the overall behavior of the transducer. The 4 outer layers are therefore considered as homogeneous layers with the properties given in Table 7.1.

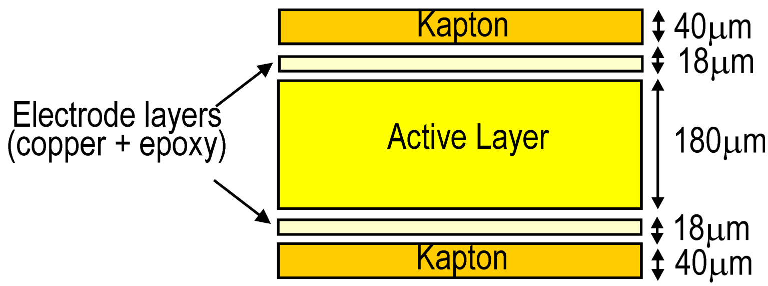


Figure 7.6: A MFC can be modeled as a 5-layer composite with an inner active layer and four passive layers

Material property	value	unit
Epoxy		
E	2.6	GPa
ν	0.33	
ρ	1500	kg/m^3
Kapton		
E	2.8	GPa
ν	0.3	
ρ	1580	kg/m^3

Table 7.1: Mechanical properties of the passive layers of MFCs

The mechanical, piezoelectric and dielectric properties of the active layers can be computed from their constituents using piezoelectric homogenization, assuming that the PZT material is Ceramtec P502, the matrix is epoxy with the properties given in Table 7.1, and the volume fraction of fibers is 86%. An analytical approach validated with detailed numerical computations has been developed in

[3] and [10]. The homogenized properties found in these studies are given in Tables 7.2 and 7.3. They correspond to the *MFC_P2_AL* and *MFC_P1_AL* properties in `m_piezo` `Database`. For the $P1$-type MFCs, the values from Table 7.3 have been corrected with the h/p factor for the piezoelectric properties, and the $(h/p)^2$ factor for the dielectric constant. A value of $h/p = 0.36$ has been used.

$P2$ MFC Homogenized Properties	Symbol	Unit	Mixing rules
Young's modulus	E_L	GPa	47.17
	E_T	GPa	16.98
Shear Modulus	G_{LT}	GPa	6.03
	G_{Tz}	GPa	6.06
	G_{Lz}	GPa	17.00
Poisson's ratio	ν_{LT}	-	0.395
Piezoelectric charge constants	d_{31}	pC/N	-183
	d_{32}	pC/N	-153
Dieletric relative constant (free)	$\varepsilon_{33}^T/\varepsilon_0$	-	1600

Table 7.2: Homogenized properties of the active layer of $P2$-MFCs calculated using the analytical mixing rules of [10]

P_1 MFC Homogenized Properties	Symbol	Unit	Mixing rules
Young's modulus	E_L	GPa	42.18
	E_T	GPa	16.97
Shear Modulus	G_{LT}	GPa	6.03
	G_{Tz}	GPa	17
	G_{Lz}	GPa	6.06
Poisson's ratio	ν_{LT}	-	0.380
Piezoelectric charge constants	d_{32}	pC/N	-176
	d_{33}	pC/N	436
Dieletric relative constant (free)	$\varepsilon_{33}^T/\varepsilon_0$	-	1593

Table 7.3: Homogenized properties of the active layer of $P1$-MFCs calculated using the analytical mixing rules of [10] (correction factor not included)

The homogenized properties can also be obtained using numerical homogenization. This allows to represent more accurately the curved electric field due to the interdigitated electrodes. This is illustrated in the next section.

7.2 Numerical modeling of piezoelectric layers with IDE

Our first numerical example concerns a bulk piezoelectric patch with inter-digitated electrodes (IDE). The principle of such electrodes is illustrated in Figure 7.7 [11]. The continuous electrodes are replaced by thin electrodes in the form of a comb with alternating polarity. This results in a curved electric field. Except close to the electrodes, the electric field is aligned in the plane of the actuator. In doing so, the extension of the patch in the plane is due to both the d_{31}-mode and d_{33}-mode. The d_{33}-mode is interesting because the value of d_{33} is 2 to 3 times higher than the d_{31}, d_{32} coefficients. In addition, as d_{33} and d_{31} have opposite sign, the application of a voltage across the IDE will lead to an expansion in the longitudinal direction, and a contraction in the lateral direction, and the amplitudes will be different.

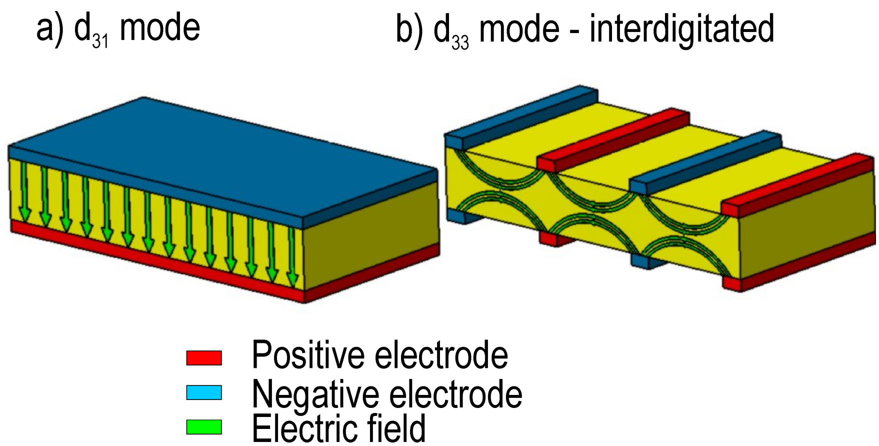


Figure 7.7: Electric field for a) continuous electrodes b) Inter-digitated electrodes

The behavior of a piezoelectric patch with interdigitated electrodes can be studied by considering a representative volume element as shown in Fig 7.8.

Let us consider such a piezoelectric patch whose geometrical properties are given in Table 7.4. The default material considered is again *SONOX_P502_iso* (Table 4.3).

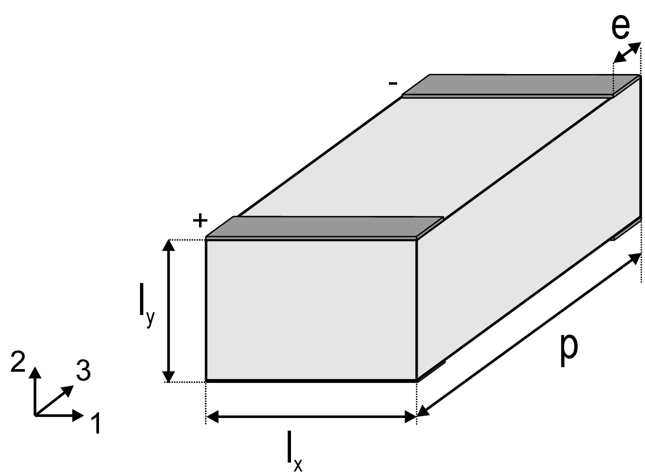


Figure 7.8: Definition of a representative volume element to study the behavior of a piezoelectric patch with IDEs

Property	Value
l_x	0.4 mm
l_y	0.3 mm
p	0.7 mm
e	0.05 mm

Table 7.4: Geometrical properties of the RVE

In `d_piezo('TutoPzPtchNumIDE-s1')` , we compute the static response due to a unit voltage applied across the electrodes, and represent the curved electric field. The mesh is first built in SI unit, and then transformed in mm.

`d_piezo('TutoPzPtchNumIDE-s2')` allows to visualize the electric field. The resulting deformation and electric field are represented in Fig 7.9.

```
%% Step 2 - visualize electric field
cf.sel(1)={'groupall','colorface none -facealpha0 -edgealpha.1'};
p_piezo('viewElec EltSel "matid1" DefLen 50e-3 reset',cf);
fecom('scd 1e-10')
p_piezo('electrodeview -fw',cf); % to see the electrodes on the mesh
iimouse('zoom reset')
```

```
cf.mdl.name='patch_IDE_EField';d_piezo('SetStyle',cf); feplot(cf);
```

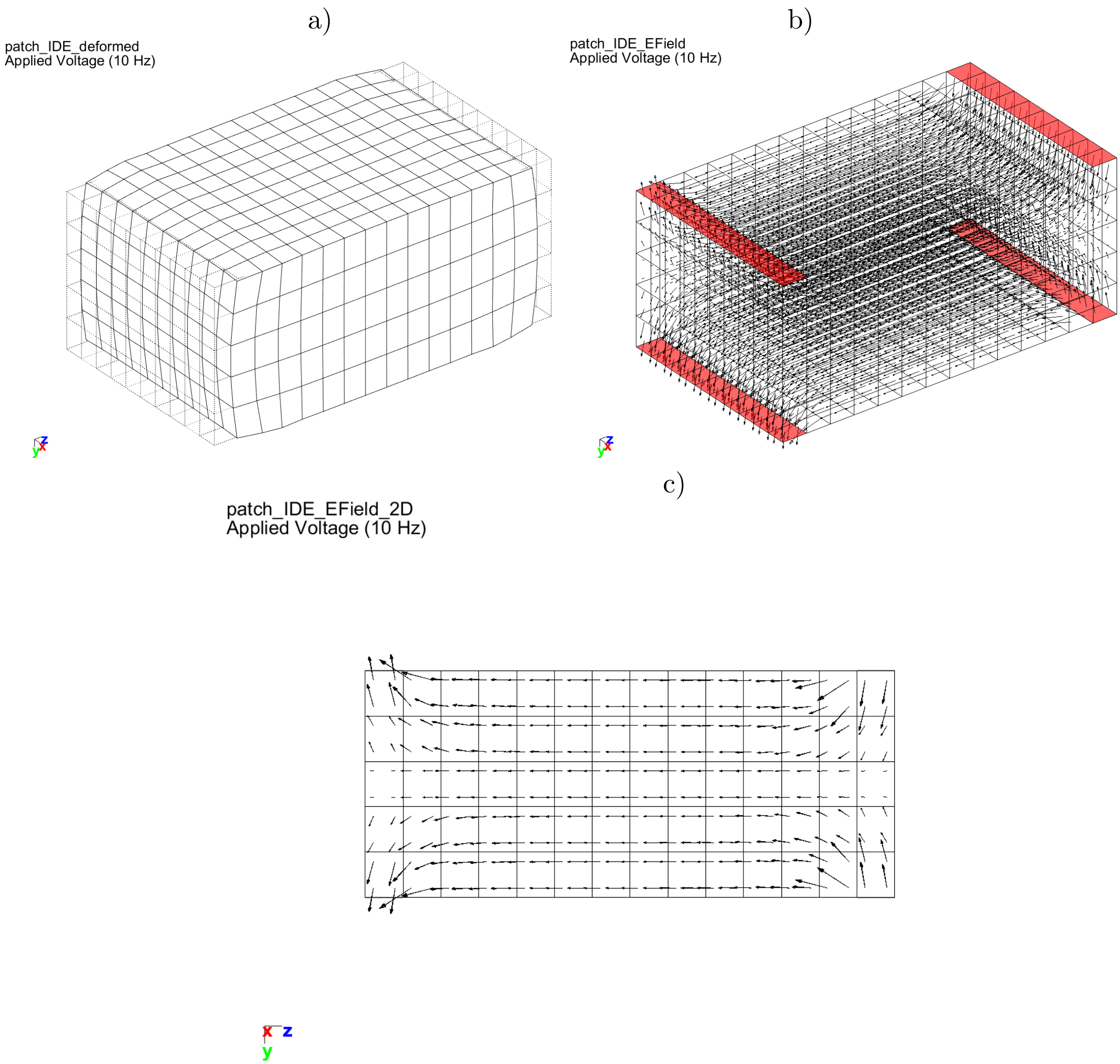


Figure 7.9: a) Free deformation of the IDE patch under unit voltage actuation b) 3D Electric field distribution and c) 2D Electric field

In `d_piezo('TutoPzPtchNumIDE-s3')` , the mean strains S_1, S_2 and S_3 and the mean electric field in direction 3 (poling direction) are then computed. In this example, the electric field is aligned with the poling direction, but is of opposite direction, resulting in a negative value of the mean strain S_3 (d_{33} is positive). Because d_{31} and d_{32} are negative, the mean values of S_1 and S_2 are positive: when the patch contracts in direction 3, it expands in directions 1 and 2. By dividing the mean strains by the mean electric field in the poling direction, one should recover the d_{31}, d_{32} and d_{33} coefficients of the material. The mean value of E_3 is however different from the value which would be obtained if the electric field was uniform (continuous electrodes on the sides of the patch). This value is considered here as the reference analytical value given by $E_3 = \frac{V}{p}$. Note that due to the conversion to mm, the voltage is expressed in μV.

```
Relation between mean strain on free structure and d_3i
    {'E3 mean'}    {[-1.1442]}    {[-1.4286]}    {'E3 analytic'}

    {'Sx/E3'}    {[-1.8500e-13]}    {[-1.8500e-13]}    {'d_31(mm/muV)'}
    {'Sy/E3'}    {[-1.8500e-13]}    {[-1.8500e-13]}    {'d_32(mm/muV)'}
    {'Sz/E3'}    {[ 4.4000e-13]}    {[ 4.4000e-13]}    {'d_33(mm/muV)'}
```

The ratio between the mean of E_3 and the analytical value is about 0.80, which means that the free strain of an IDE patch with the geometrical properties considered in this example will be about 20 % lower than if the electric field was uniform. In fact, the spacing of the electrodes in an IDE patch is a compromise between the loss of performance due to the part of the piezoelectric material in which the electric field is not aligned with the poling direction, and the distance between the electrodes which, when increased, decreases the effective electric field for a given applied voltage.

In `d_piezo('TutoPzPtchNumIDE-s4')` , the total charge on the electrodes and the charge density are computed, which allows also to compute the value of the capacitance. Figure 7.10 shows the distribution of the charge density on the electrodes.

The capacitance of the patch can be computed and compared to the analytical value (for a uniform field) given by $C^T = \frac{\varepsilon^T (l_x\, l_y)}{p}$:

```
    {'C_{IDE}'}    {[2.5019e-18]}    {[2.8080e-18]}    {'C analytic'}
```

The capacitance of the IDE patch is about 10% lower than the analytical value.

In `d_piezo('TutoPzPtchNumIDE-s5')` , we use `fe_stress` to compute and visualize the stress and strain maps in the patch with IDE electrodes. Figure 7.11 shows the colormap of T_1, T_2 and T_3. It is clear that the curved electrical field induces important stress concentrations in the area close to the electrodes

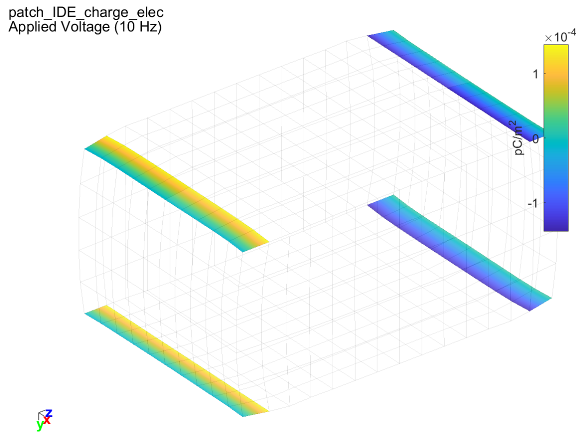


Figure 7.10: Charge density on the electrodes resulting from a static unit voltage applied to the IDEs

```
%% Step 5 - Stress and strain  visualisation
% Stress field using fe_stress
c1=fe_stress('stressAtInteg -gstate',model,def);
cf.sel='reset';cf.def=fe_stress('expand',model,c1);
cf.def.lab={'T11';'T22';'T33';'T23';'T13';'T12';'D1';'D2';'D3'}; %
fecom('colordata 99 -edgealpha.1');
fecom('colorbar',d_imw('get','CbTR','String','Stress/Voltage [kPa/muV]'));
iimouse('trans2d 0 0 0 1.6 1.6 1.6')
```

The strains and electrical displacement can also be computed with `fe_stress`, but this requires to replace the piezoelectric material with a 'PiezoStrain' material:

```
%- Strain  visualisation
% Replace with 'PiezoStrain' material
mo2=model; mo2.pl=m_piezo('dbval 1 PiezoStrain')
% Now represent strain fields using fe_stress
c1=fe_stress('stressAtInteg -gstate',mo2,def);
cf.sel='reset';cf.def=fe_stress('expand',mo2,c1);
cf.def.lab={'S11';'S22';'S33';'S23';'S13';'S12';'E1';'E2';'E3'};
fecom('colordata 99 -edgealpha.1');
fecom('colorbar',d_imw('get','CbTR','String','Strain/Voltage [1/muV]'));
iimouse('trans2d 0 0 0 1.6 1.6 1.6')
```

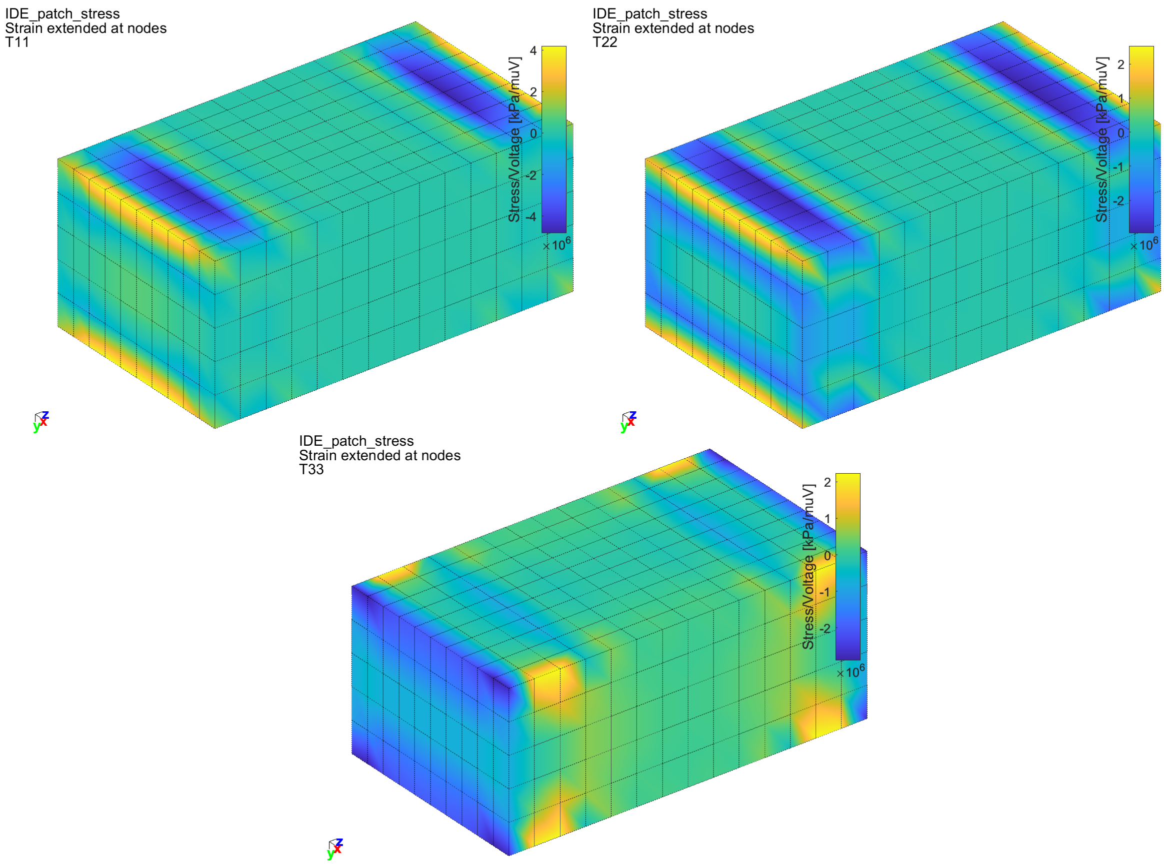


Figure 7.11: Colormap of stresses T_1, T_2 and T_3 due to applied voltage on the patch with IDE

Figure 7.12 shows the colormap of S_1, S_2 and S_3. The strain is uniform in the central region, but there are strong variations in the areas under the electrodes.

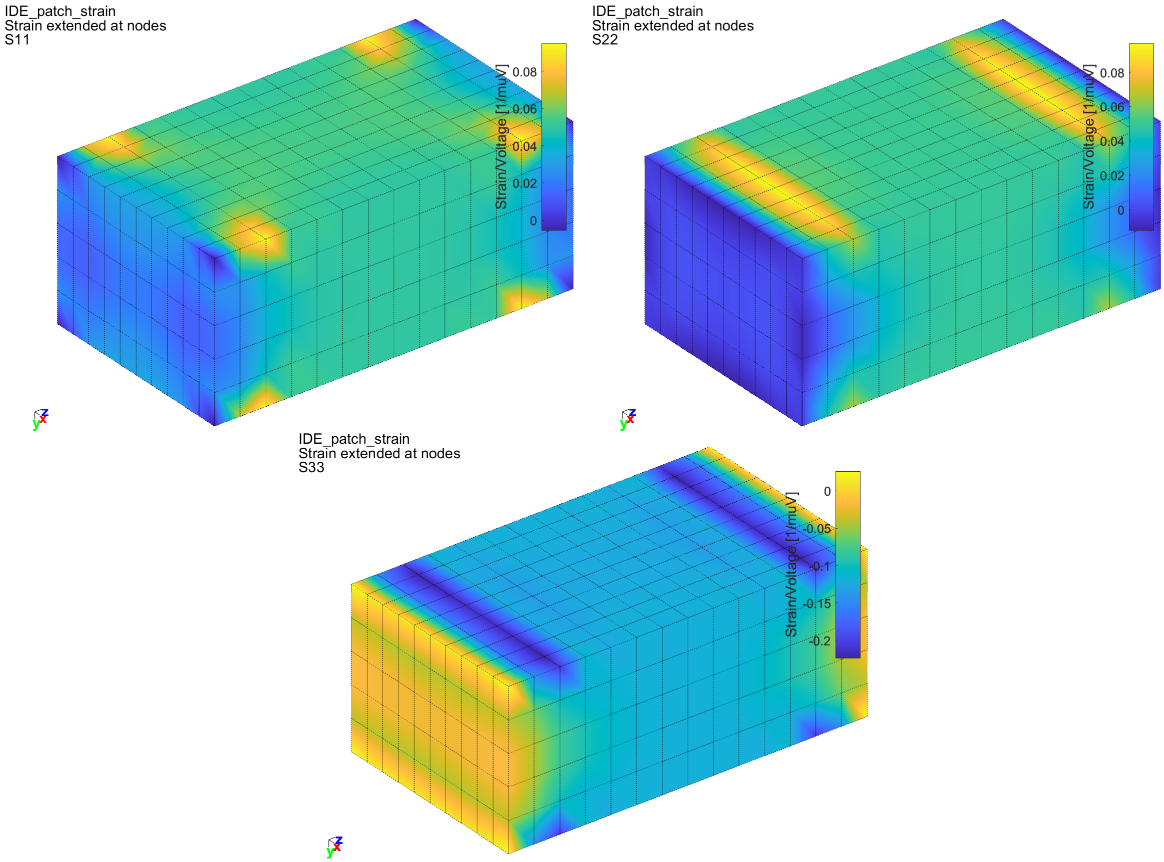


Figure 7.12: Colormap of strains S_1, S_2 and S_3 due to applied voltage on the patch with IDE

Note that in this example, the poling direction has been considered to be aligned with the z-axis. In practice, as the IDE patch is usually poled using the IDE electrodes, the poling direction is aligned with the curved electric field lines. This difference of poling only concerns a few elements in the mesh, and from a global point of view, it does not have an important impact on the assessment of the performance of the patch, but it may have an important impact on the prediction of strains and stresses around the electrode areas. Aligning the poling direction with the curved electric field lines is possible but requires the handling of local basis which are oriented based on the computed electric field lines.

7.3 Periodic homogenization of piezocomposite layers

In this example, we will show how to compute the homogeneous equivalent mechanical, piezoelectric and dielectric properties of both $P1$ and $P2$-type MFCs. The methodology is general and can be extended to other types of piezocomposites.

7.3.1 Constitutive equations

For d_{31} patches, the poling direction (conventionally direction 3) is normal to the plane of the patches (Figure 7.13a) and according to the plane stress assumption $T_3 = 0$. The electric field is assumed to be aligned with the polarization vector ($E_2 = E_1 = 0$). The constitutive equations reduce to:

$$\begin{Bmatrix} T_1 \\ T_2 \\ T_4 \\ T_5 \\ T_6 \\ D_3 \end{Bmatrix} = \begin{bmatrix} c_{11}^{E*} & c_{12}^{E*} & 0 & 0 & 0 & -e_{31}^{*} \\ c_{12}^{E*} & c_{22}^{E*} & 0 & 0 & 0 & -e_{32}^{*} \\ 0 & 0 & c_{44}^{E*} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{55}^{E*} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{66}^{E*} & 0 \\ e_{31}^{*} & e_{32}^{*} & 0 & 0 & 0 & \varepsilon_{33}^{S*} \end{bmatrix} \begin{Bmatrix} S_1 \\ S_2 \\ S_4 \\ S_5 \\ S_6 \\ E_3 \end{Bmatrix} \tag{7.7}$$

where the superscript * denotes the properties under the plane stress assumption (which are not equal to the properties in 3D).

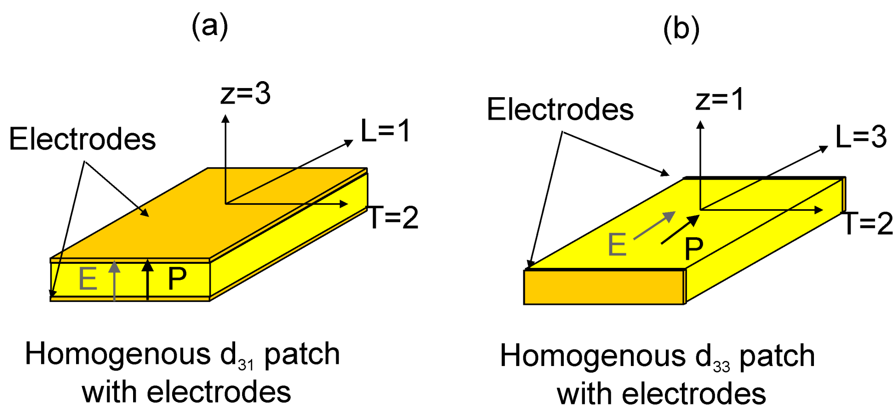


Figure 7.13: Homogeneous models of the piezoelectric layers with electrodes : d_{31} and d_{33} piezoelectric layers

For d_{33} patches, although the electric field lines do not have a constant direction, when replacing the active layer by an equivalent homogeneous layer, we consider that the poling direction is that of the fibers (direction 3, Figure 7.13b), and that the electric field is in the same direction. With this reference frame, the plane stress hypothesis implies that $T_1 = 0$. The constitutive equations are given by

$$\left\{\begin{array}{c} T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ D_3 \end{array}\right\} = \left[\begin{array}{cccccc} c_{22}^{E*} & c_{23}^{E*} & 0 & 0 & 0 & -e_{32}^{*} \\ c_{32}^{E*} & c_{33}^{E*} & 0 & 0 & 0 & -e_{33}^{*} \\ 0 & 0 & c_{44}^{E*} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{55}^{E*} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{66}^{E*} & 0 \\ e_{32}^{*} & e_{33}^{*} & 0 & 0 & 0 & \varepsilon_{33}^{S*} \end{array}\right] \left\{\begin{array}{c} S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ E_3 \end{array}\right\} \tag{7.8}$$

7.3.2 Periodic homogenization

Homogenization techniques are widely used in composite materials. They consist in computing the homogeneous, equivalent properties of multi-phase heterogeneous materials. The homogenization is performed on a so-called representative volume element (RVE) which is a small portion of the composite which, when repeated in 1, 2 or 3 directions forms the full composite. In the case of flat transducers considered here, the composite is periodic in 2 directions (the directions of the plane of the composite). Equivalent properties are obtained by writing the constitutive equations (Equation ((7.7)) or ((7.8)) in this case) in terms of the average values of T_i, S_i, D_i, E_i on the RVE:

$$\begin{array}{ll} \overline{T_i} = \frac{1}{V}\int_V T_i dV & \qquad \overline{D_i} = \frac{1}{V}\int_V D_i dV \\ \overline{S_i} = \frac{1}{V}\int_V S_i dV & \qquad \overline{E_i} = \frac{1}{V}\int_V E_i dV \end{array}$$

where $^-$ denotes the average value.

For both types of piezocomposites, matrix $\left[c^{E*}\right]$ is a function of the longitudinal (in the direction of the fibers) and transverse in-plane Young's moduli (E_L and E_T), the in plane Poisson's ratio ν_{LT}, the in-plane shear modulus G_{LT}, and the two out-of-plane shear moduli G_{Lz} and G_{Tz}. Matrix $[e^*]$ is given by

$$[e^*] = [d]\left[c^{E*}\right]$$

where

$$[d] = \left[\begin{array}{ccccc} d_{31} & d_{32} & 0 & 0 & 0 \end{array}\right]$$

in the case of d_{31} piezocomposites and

$$[d] = \left[\begin{array}{ccccc} d_{32} & d_{33} & 0 & 0 & 0 \end{array}\right]$$

in the case of d_{33} piezocomposites. Note that the coefficients d_{ij} are unchanged under the plane stress hypothesis.

When used as sensors or actuators, piezocomposite transducers are typically equipped with two electrodes. These electrodes impose an equipotential voltage on their surfaces, and the electrical variables are the voltage difference V across the electrodes, and the electrical charge Q. These two variables are representative of the electrical macro variables which will be used in the numerical models of structures equipped with such transducers : transducers are used either in open-circuit conditions ($Q = 0$ or imposed) or short-circuit conditions ($V = 0$ or imposed). Instead of the average values of D_i and E_i, the macro variables Q and V are therefore used in the homogenization process. For a homogeneous d_{33} transducer (Figure 7.14), the constitutive equations can be rewritten in terms of these macro variables:

$$\left\{ \begin{array}{c} T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ Q \end{array} \right\} = \left[\begin{array}{cccccc} c_{22}^{(SC*)} & c_{23}^{(SC*)} & 0 & 0 & 0 & -e_{32}^*/p \\ c_{32}^{(SC*)} & c_{33}^{(SC*)} & 0 & 0 & 0 & -e_{33}^*/p \\ 0 & 0 & c_{44}^{(SC*)} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{55}^{(SC*)} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{66}^{(SC)*} & 0 \\ e_{32}^*A & e_{33}^*A & 0 & 0 & 0 & \varepsilon_{33}^{s*}A/p \end{array} \right] \left\{ \begin{array}{c} S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ -V \end{array} \right\} \tag{7.9}$$

where SC stands for 'short-circuit' ($V = 0$), p is the length of the transducer, A is the surface of the electrodes of the equivalent homogeneous transducer and Q is the charge collected on the electrodes.

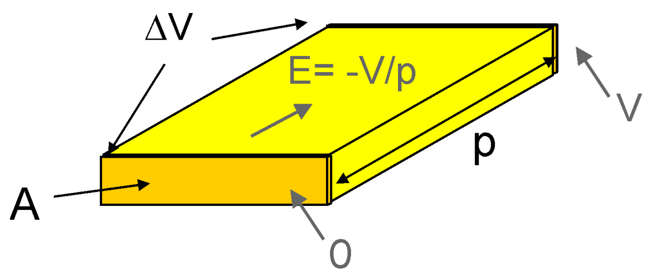


Figure 7.14: Homogeneous model of the d_{33} piezocomposite and definition of the macro variables

For d_{31}-piezocomposites, the approach is identical.

7.3.3 Definition of local problems

The RVE is made of two different materials. In order to find the homogeneous constitutive equations, Equation (7.9) is written in terms of the average values of the mechanical quantities S_i and T_i in the RVE and the electrical variables Q and V defined on the electrodes:

$$\left\{\begin{array}{c}\overline{T_2}\\\overline{T_3}\\\overline{T_4}\\\overline{T_5}\\\overline{T_6}\\Q\end{array}\right\}=\begin{bmatrix}\overline{c_{22}}^{(SC*)} & \overline{c_{23}}^{(SC*)} & 0 & 0 & 0 & -\overline{e_{32}}^*/p\\\overline{c_{32}}^{(SC*)} & \overline{c_{33}}^{(SC*)} & 0 & 0 & 0 & -\overline{e_{33}}^*/p\\0 & 0 & \overline{c_{44}}^{(SC*)} & 0 & 0 & 0\\0 & 0 & 0 & \overline{c_{55}}^{(SC*)} & 0 & 0\\0 & 0 & 0 & 0 & \overline{c_{66}}^{(SC*)} & 0\\\overline{e_{32}}^*A & \overline{e_{33}}^*A & 0 & 0 & 0 & \overline{\varepsilon_{33}}^{S*}A/p\end{bmatrix}\left\{\begin{array}{c}\overline{S_2}\\\overline{S_3}\\\overline{S_4}\\\overline{S_5}\\\overline{S_6}\\-V\end{array}\right\}\tag{7.10}$$

The different terms in Equation (7.10) can be identified by defining local problems on the RVE. The technique consists in imposing conditions on the different strain components and V and computing the average values of the stress and the charge in order to find the different coefficients. For the electric potential, two different conditions ($V = 0, 1$) are used. For the mechanical part, we assume that the displacement field is periodic in the plane of the transducer: on the boundary of the RVE, the displacement can be written :

$$u_i = \overline{S}_{ij}\, x_j + v_i \tag{7.11}$$

where u_i is the i^{th} component of displacement, $\overline{S}_{ij}$ is the average strain in the RVE (tensorial notations are used), x_j is the j^{th} spatial coordinate of the point considered on the boundary, and v_i is the periodic fluctuation on the RVE. The fluctuation v is periodic in the plane of the transducer so that between two opposite faces (noted A^-/A^+, B^-/B^+ and C^-/C^+, Figure 7.15), one can write $(v(x_j^{K+}) = v(x_j^{K-})$, $K = A, B, C)$:

$$u_i^{K+} - u_i^{K-} = \overline{S}_{ij}\,(x_j^{K+} - x_j^{K-}) \qquad K = A, B, C \tag{7.12}$$

Because we consider a plate with the plane stress hypothesis $T_1 = 0$, equation (7.12) should not be satisfied for $K = A$ and $j = 1$ (no constraint on the normal displacement on faces A^+ and A^-). For a given value of the average strain tensor ($\overline{S}_{ij}$), equation ((7.12)) defines constraints between the points on each pair of opposite faces. This is illustrated in Figure 7.16, where an average strain S_2 is imposed on the RVE and the constraints are represented for u_2 on faces B^- and B^+.

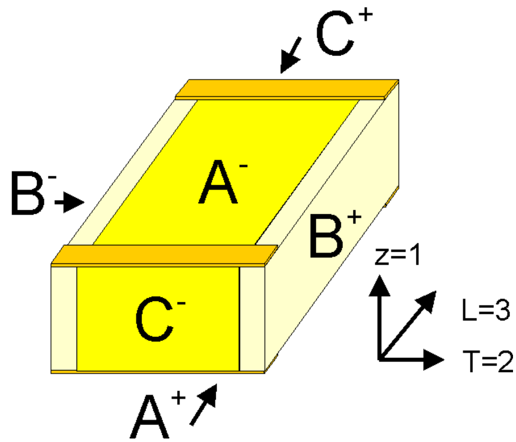


Figure 7.15: Definition of pairs of opposite faces on the RVE

Note that these constraints do not impose that the faces of the RVE remain plane, which is important for the evaluation of the shear stiffness coefficients.

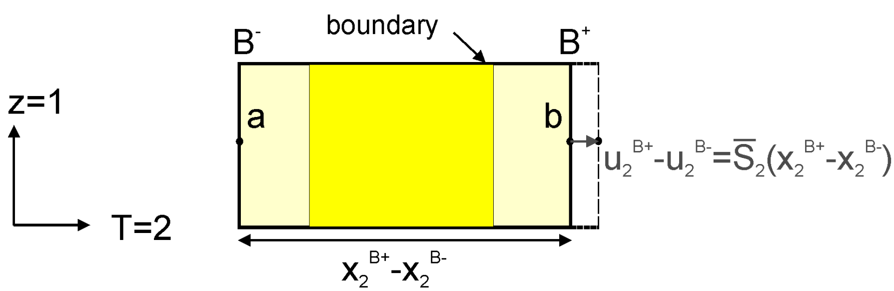


Figure 7.16: Example of an average strain S_2 imposed on the RVE and associated periodic conditions

In total, six local problems are needed to identify all the coefficients in (7.10) (Figure 7.17). The first problem consists in applying a difference of potential V to the electrodes of the RVE and imposing zero displacement on all the faces (except the top and bottom). In the next five local problems, the difference of potential is set to 0 (short-circuited condition), and five deformation mechanisms are induced. Each of the deformation mechanisms consists in a unitary strain in one of the directions (with zero strain in all the other directions). For each case, the average values of T_i and S_i, and the charge accumulated on the electrodes Q, are computed, and used to determine all the coefficients in (7.10), from which the engineering constants are determined.

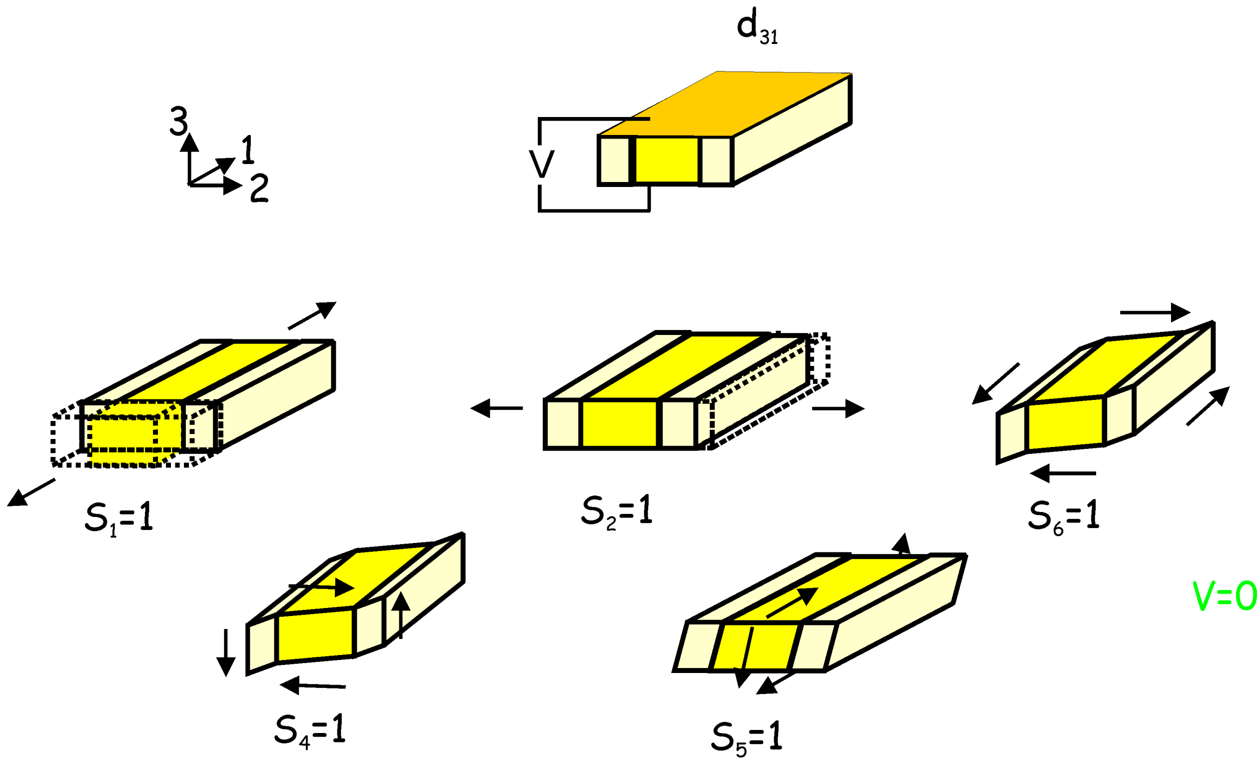


Figure 7.17: The six local problems solved by the finite element method in order to compute the homogenized properties of d_{31}-MFCs

7.3.4 Application of numerical periodic piezoelectric homogenization to $P2$-MFCs

We are going to compute the homogeneous properties of a $P2$-type MFC with varying volume fraction of piezoelectric fibers.

In `d_piezo('TutoPzMFCP2Homo-s1')` , we first define the range of volume fractions to compute the homogeneous properties and the dimensions of the RVE.

```
% See full example as MATLAB code in d_piezo('ScriptTutoPzMFCP2Homo')
d_piezo('DefineStyles');

%% Step 1 - Meshing of RVE
% Meshing script can be viewed with sdtweb d_piezo('MeshHomoMFCP2')
Range=fe_range('grid',struct('rho',[0.001 linspace(0.1,0.9,9) .999], ...
  'lx',.300,'ly',.300,'lz',.180,'dd',0.04));
```

Then in d_piezo('TutoPzMFCP2Homo-s2') for each value of the volume fraction of piezoelectric fibers ρ, we compute the solution of the six local problems, the average value of stress, strain and charge. From these values we extract the engineering mechanical properties, the piezoelectric and dielectric properties.

```
%% Step 2 - Loop on volume fraction and compute homogenize properties
for jPar=1:size(Range.val,1)

RO=fe_range('valCell',Range,jPar,struct('Table',2));% Current experiment

% Create mesh
model= ...
d_piezo(sprintf(['meshhomomfcp2 rho=%0.5g lx=%0.5g ly=%0.5g' ...
  'lz=%0.5g dd=%0.5g'],[RO.rho RO.lx RO.ly RO.lz RO.dd]));


% Define the six local problems
RB=struct('CellDir',[max(model.dx) max(model.dy) max(model.dz)],'Load', ...
       {{'e11','e22','e12','e23','e13','vIn'}});
 % Periodicity on u,v,w on x and y face
 % periodicity on u,v only on the z face
  RB.DirDofInd={[1:3 0],[1:3 0],[1 2 0 0]};
 % Voltage DOFs are always eliminated from periodic conditions

 % Compute the deformation for the six local problems
 def=fe_homo('RveSimpleLoad',model,RB);
 % Represent the deformation of the RVE
 cf=comgui('guifeplot-reset',2);cf=feplot(model,def); fecom('colordatamat')


% Compute stresses, strains and electric field
a1=p_piezo('viewstrain -curve -mean -EltSel MatId1 reset',cf); % Strain epoxy
a2=p_piezo('viewstrain -curve -mean -EltSel MatId2 reset',cf); % Strain piezo
b1=p_piezo('viewstress -curve -mean- EltSel MatId1 reset',cf); % Stress epoxy
b2=p_piezo('viewstress -curve -mean- EltSel MatId2 reset',cf); % Stress piezo

% Compute charge on electrodes
 mo1=cf.mdl.GetData;
  ind1=fe_case(mo1,'getdata','Top Actuator');ind1=fix(ind1.InputDOF);
```

```
  mo1=p_piezo('electrodesensq TopQ2',mo1,struct('MatId',2,'InNode',ind1));
  mo1=p_piezo('electrodesensq TopQ1',mo1,struct('MatId',1,'InNode',ind1));
  c1=fe_case('sensobserve',mo1,'TopQ1',cf.def); q1=c1.Y;
  c2=fe_case('sensobserve',mo1,'TopQ2',cf.def); q2=c2.Y;


% Compute average values:
a0=a1.Y(1:6,:)*(1-RO.rho)+a2.Y(1:6,:)*RO.rho;
b0=b1.Y(1:6,:)*(1-RO.rho)+b2.Y(1:6,:)*RO.rho;
q0=q1+q2; % Total charge is the sum of charges on both parts of electrode


% Compute C matrix
C11=b0(1,1)/a0(1,1); C12=b0(1,2)/a0(2,2); C22=b0(2,2)/a0(2,2);
C44=b0(4,4)/a0(4,4); C55=b0(5,5)/a0(5,5); C66=b0(6,3)/a0(6,3);
sE=inv([C11 C12; C12 C22]);

% Extract mechanical engineering constants
E1(jPar)=1/sE(1,1); E2(jPar)=1/sE(2,2); nu12(jPar)=-sE(1,2)*E1(jPar);
nu21(jPar)=-sE(1,2)*E2(jPar);G12(jPar)=C66; G23(jPar)=C44; G13(jPar)=C55;

% Extract piezoelectric properties
e31(jPar)=b0(1,6)*RO.lz; e32(jPar)=b0(2,6)*RO.lz;
d=[e31(jPar) e32(jPar)]*sE; d31(jPar)=d(1); d32(jPar)=d(2);

% Extract dielectric properties
eps33(jPar)=-q0(6)*RO.lz/(RO.lx*RO.ly);
eps33t(jPar)=eps33(jPar)+ [d31(jPar) d32(jPar)]*[e31(jPar); e32(jPar)];

end % Loop on rho0 values
```

Figure 7.18 represents the solution of the six local problems for $\rho = 0.6$.

In `d_piezo('TutoPzMFCP2Homo-s3')` , we can now plot the evolution of the homogeneous properties of the $P2$-type MFC as a function of the volume fraction ρ:

Figure 7.19 represents the evolution of the mechanical properties and Figure 7.20 represents the evolution of the piezoelectric and dielectric properties as a function of ρ. The properties of MFC transducers correspond to the value of $\rho = 0.86$.

H]

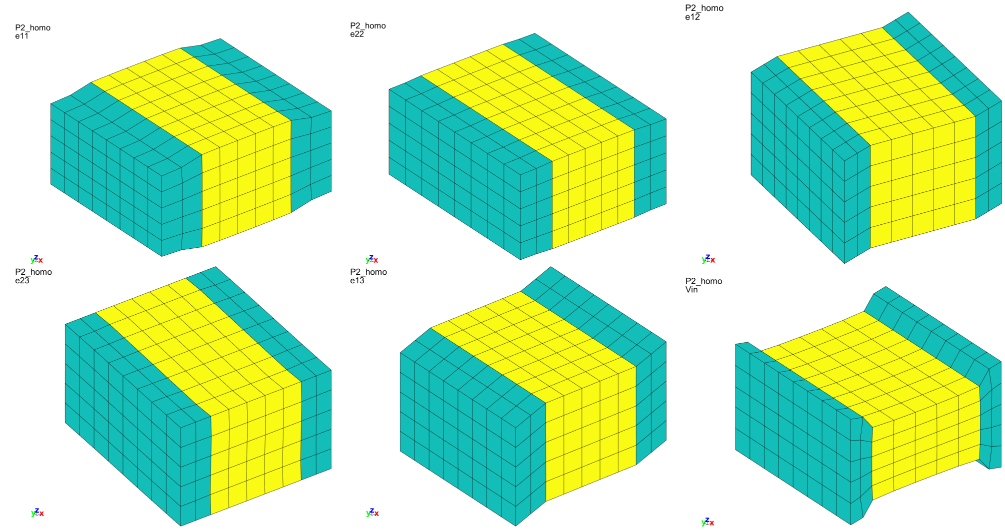


Figure 7.18: Solutions of the six local problems on the RVE for $\rho = 0.6$ for a $P2$-type MFC

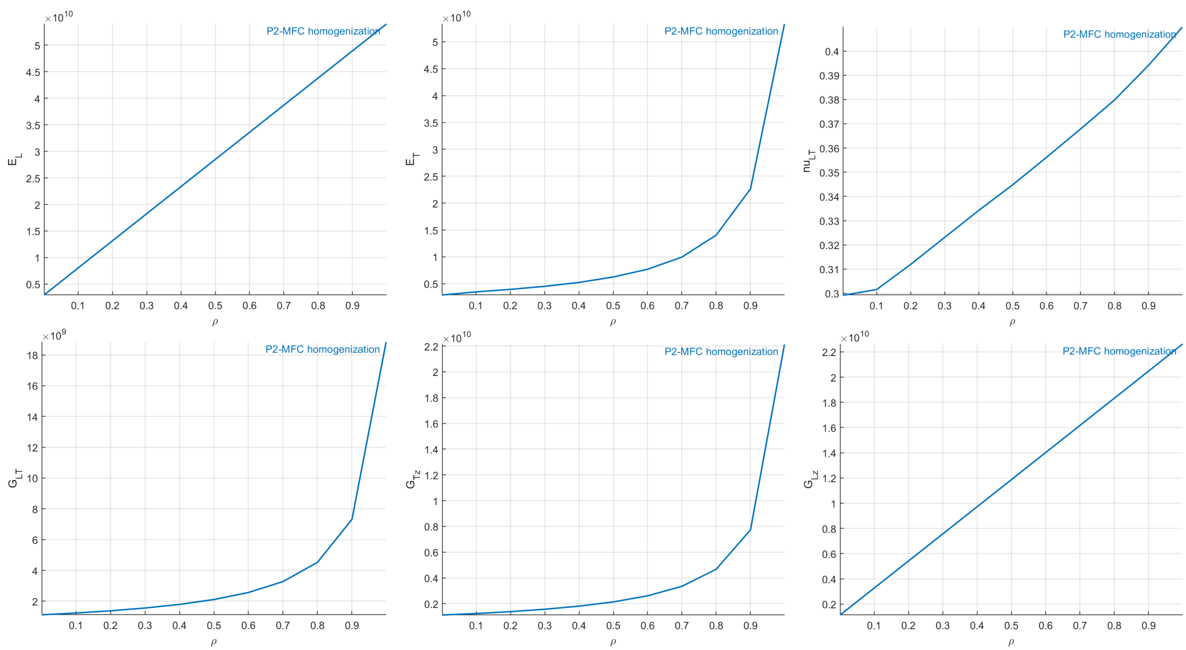


Figure 7.19: Evolution of the homogeneous mechanical properties of a $P2$-type piezocomposite as a function of ρ

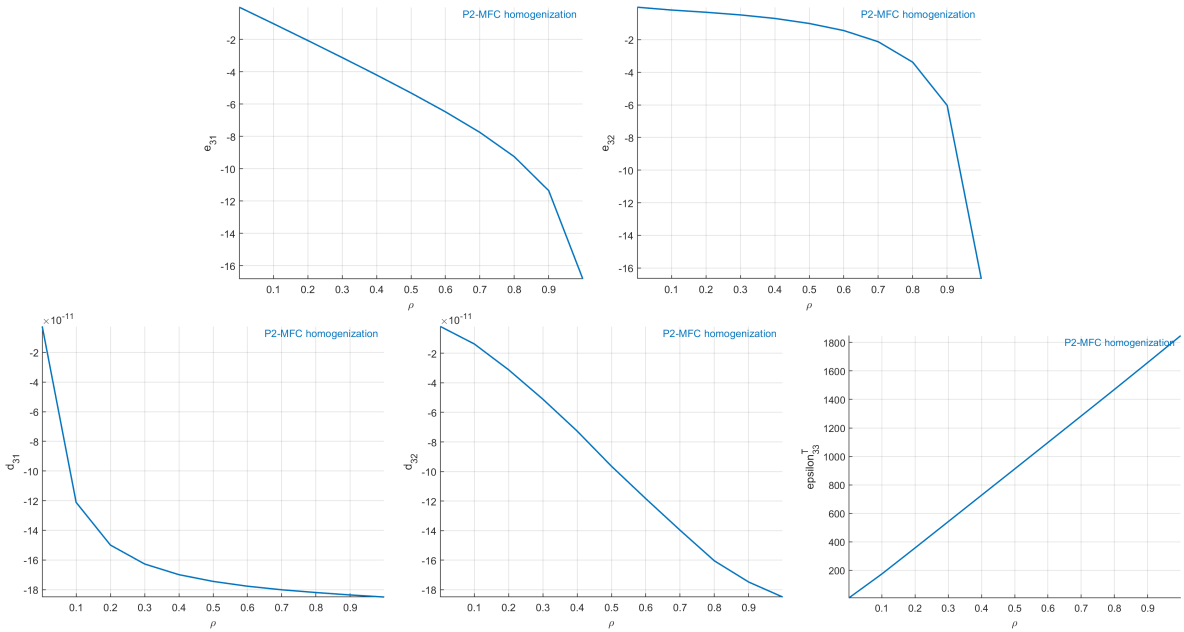


Figure 7.20: Evolution of the homogeneous piezoelectric and dielectric properties of a $P2$-type piezocomposite as a function of ρ

7.3.5 Application of numerical periodic piezoelectric homogenization to $P1$-MFCs

As we did for the $P2$-type MFC, we will now compute the homogeneous properties of a $P1$-type MFC with varying volume fraction of piezoelectric fibers.

In `d_piezo('TutoPzMFCP1Homo-s1')` , we first define the range of volume fractions to compute the homogeneous properties and the dimensions of the RVE.

Then in `d_piezo('TutoPzMFCP1Homo-s2')` for each value of the volume fraction of piezoelectric fibers ρ, we compute the solution of the six local problems, the average value of stress, strain and charge. From these values we extract the engineering mechanical properties, the piezoelectric and dielectric properties. The process is identical to the one in the previous example with $P2$-type MFCs.

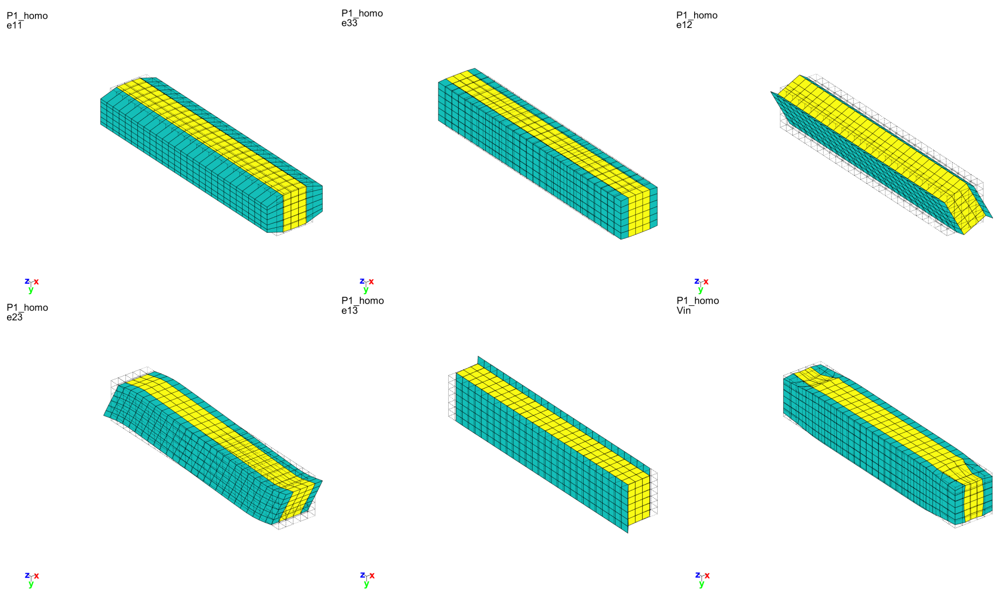


Figure 7.21: Solutions of the six local problems on the RVE for $\rho = 0.6$

Figure 7.22 represents the inhomogeneous electric field for the sixth local problem (applied voltage) and $\rho = 0.6$.

In `d_piezo('TutoPzMFCP1Homo-s3')` , we plot the evolution of the homogeneous properties of the $P1$-type MFC as a function of the volume fraction ρ

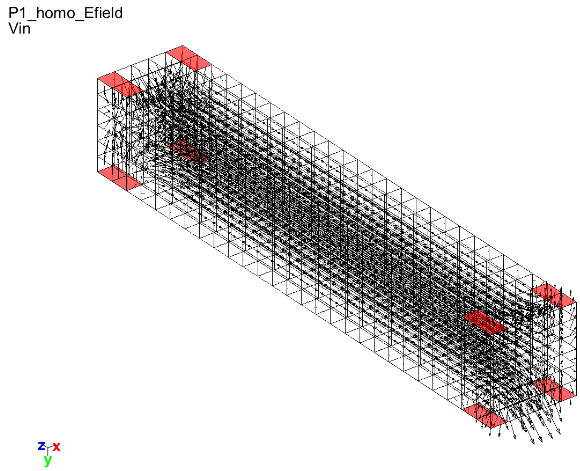


Figure 7.22: Electric field for the sixth local problem (applied voltage) on the RVE of a $P1$-type MFC for $\rho = 0.6$

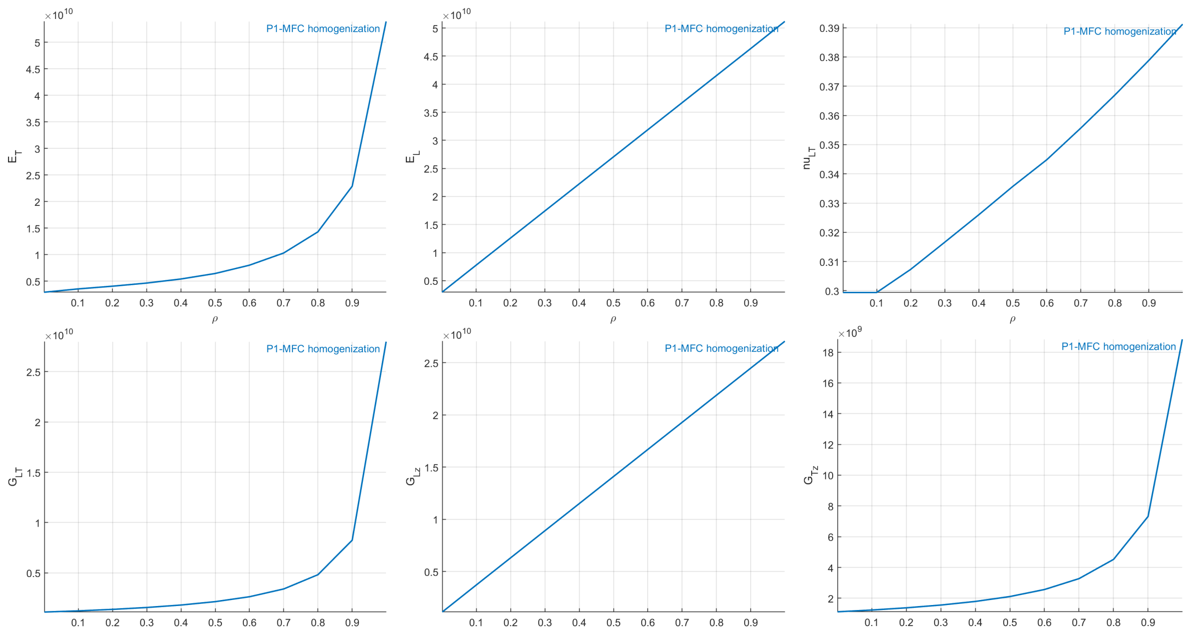


Figure 7.23: Evolution of the homogeneous mechanical properties of a $P1$-type piezocomposite as a function of ρ

Figure 7.23 represents the evolution of the mechanical properties and Figure 7.24 represents the evolution of the piezoelectric and dielectric properties as a function of ρ. The properties of MFC transducers correspond to the value of $\rho = 0.86$.

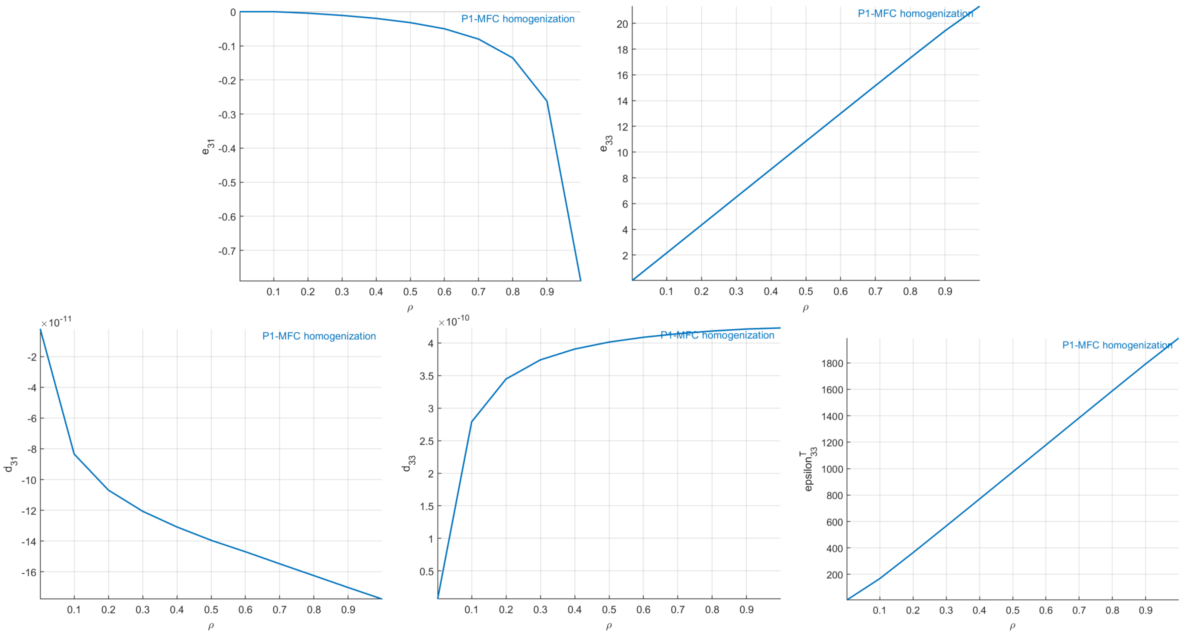


Figure 7.24: Evolution of the homogeneous piezoelectric and dielectric properties of a $P1$-type piezocomposite as a function of ρ

All the properties match well the results presented in [3]

7.4 Example of MFC transducers integrated in plate structures

This example deals with a cantilever aluminum plate with two $P1$-type MFCs (M8528-P1) attached on each side of the plate. The geometry is represented in Figure 7.25. The plate is meshed with rectangular piezoelectric elements. The main part of the beam is made of one layer (aluminum), and the part where the two MFCs are attached is made of 11 layers (5 layers for each MFC and the central aluminum layer).

In `d_piezo('TutoPzMFCPlate-s1')` , we build the mesh, define the sensors and actuators, and compute the static response. Using the two MFCs as actuators, we define two combinations in order to induce bending or traction and compute and represent the static response in (Figure 7.26).

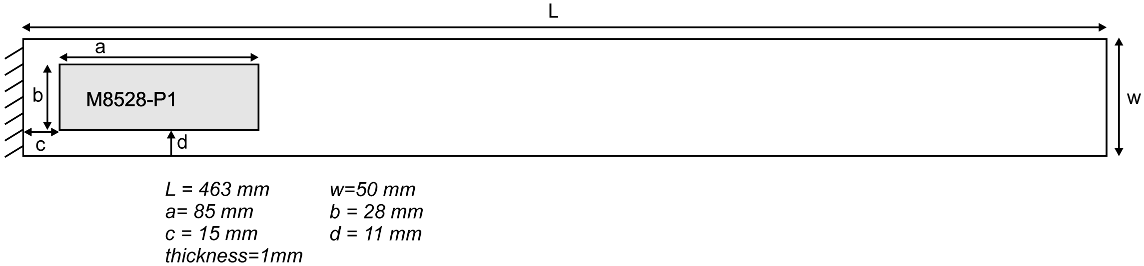
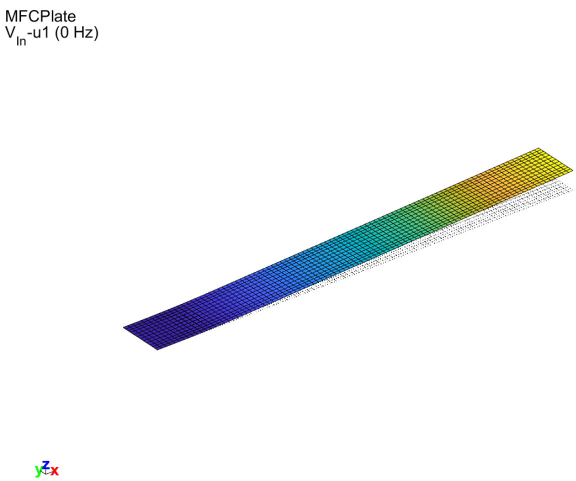


Figure 7.25: Geometric details of the aluminum plate with 2 $P1$-type MFCs

a)



b)

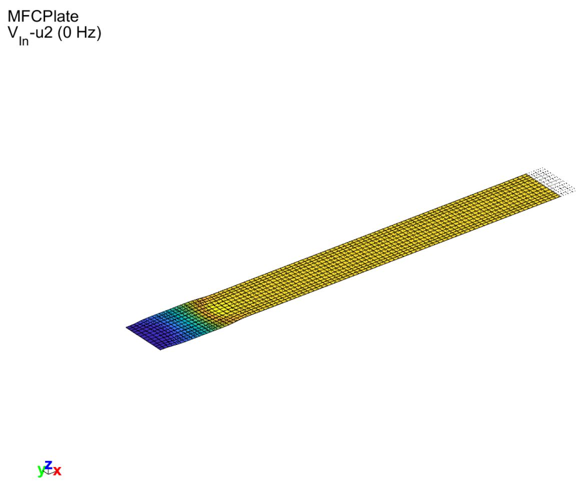


Figure 7.26: Static deformation under combined voltage actuation in a) bending, b) traction

The deformed shape under traction actuation highlights the fact that the induced strain in the lateral direction is of opposite sign with respect to the longitudinal direction, which is due to the fact that we are using a $P1$-type MFC. $F1$-type MFCs are based on the same layout as $P1$-types but the fibers are oriented with an angle of 45°. Such transducers can be easily modeled by changing the angle of the active layer in the multi-layer sequence. Assume that the bottom MFC makes an angle of 45° with respect to the axis of the beam and that the top MFC makes an angle of -45°. Each actuator induces both bending and torsion in the plate.

This is done in `d_piezo('TutoPzMFCPlate-s2')` . The torsion can be easily seen by looking at the deformed shape resulting from the combination of the two MFCs with opposite signs which cancels the bending effect, as shown in Figure 7.27.

```
%% Step 2 - Rotate fibers
model.il(2,[20 44])=[45 -45];
d1=fe_simul('dfrf',stack_set(model,'info','Freq',0)); % static response
cf.def=d1; fecom('scd 10');  C2=fe_case('SensObserve - dim 2 3 1',sens,d1);
```

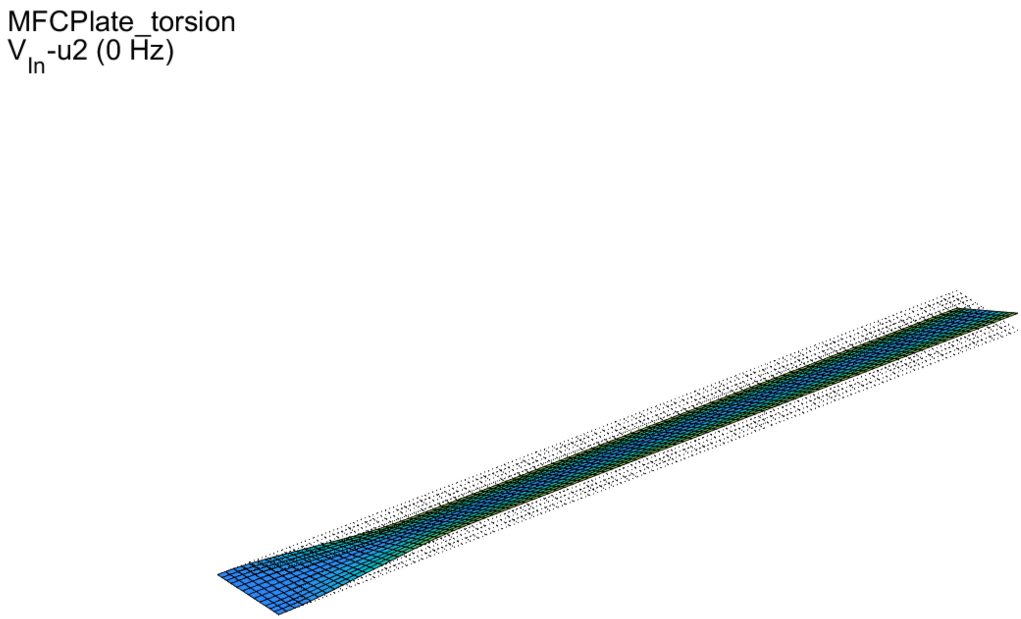


Figure 7.27: Static deformation under combined voltage with opposite sign using two $F1$-type MFCs

7.5 Using shaped orthotropic piezocomposite transducers

7.5.1 Introduction

Typical piezoelectric transducers found on the market are rectangular or circular. Different researchers have however studied the possibility to use more complex shapes. This idea was mainly driven by the active control applications. The first developments in this direction concern triangular actuators which will be used in the following example after recalling the theory behind the development of such transducers. When used as an actuator, an applied voltage on a piezoelectric patch results in a set of balanced forces on the supporting structure. The analytical expression of these so-called equivalent forces has been derived analytically in [12] in the general case of an orthotropic patch poled through the thickness with an arbitrary shape and attached to a supporting plate (assumed to follow Kirchhoff plate theory). The analytical expressions show that these forces are a function of the material properties of the piezo, as well as the expression of the normal to the contour of the patch. In the case of a triangular patch, the equivalent forces are illustrated on Figure 7.28. They consist in point forces P and $P/2$, and bending moments M_1 and M_2 whose expressions are:

$$\begin{aligned} P = & \quad -(e_{31}-e_{32})\frac{bl}{\frac{b^2}{4}+l^2} z_m V \\ M_1 = & \quad -e_{31} z_m V \\ M_2 = & \quad -\frac{\frac{b^2}{4}e_{31}+l^2 e_{32}}{\frac{b^2}{4}+l^2} z_m V \end{aligned} \tag{7.13}$$

These expressions show that the point forces are only present when the material is orthotropic ($e_{31} \neq e_{32}$). An interesting application of the triangular actuator is to design it to be a point-force actuator. This requires (i) to clamp the base of the triangle in order to cancel the bending moment M_1 and the two point forces $P/2$, (ii) to cancel M_2. The result is a single point-force P at the tip of the triangle. The cancelation of M_2 requires to have [13]:

$$M_2 = -\frac{\frac{b^2}{4}e_{31}+l^2 e_{32}}{\frac{b^2}{4}+l^2} z_m V = 0 \longrightarrow \frac{b}{l} = 2\sqrt{\frac{-e_{32}}{e_{31}}} \tag{7.14}$$

This expression shows that the point-force actuator can only be achieved when e_{31} and e_{32} are of opposite sign. With a PZT ceramic, this is possible using inter-digitated electrodes, which result in a compression in the lateral direction when the transducers elongates in the longitudinal direction. A possibility is to use a triangular transducer based on the same principle as the $P1-type$ MFC. The resulting force at the tip of the triangle is given by:

$$P = -e_{31}\frac{b}{l} z_m V \tag{7.15}$$

Given the material properties of the active layer of $P1$-MFCs in Table 7.3, the values of e_{31} and e_{32} are:

$$\begin{aligned} e_{31} &= \left(\frac{E_L}{1-\nu_{TL}\nu_{LT}}\right) d_{31} + \left(\frac{\nu_{LT}E_T}{1-\nu_{TL}\nu_{LT}}\right) d_{32} = 18.32 C/m^2 \\ e_{32} &= \left(\frac{\nu_{TL}E_L}{1-\nu_{TL}\nu_{LT}}\right) d_{31} + \left(\frac{E_T}{1-\nu_{TL}\nu_{LT}}\right) d_{32} = -0.1859 C/m^2 \end{aligned} \tag{7.16}$$

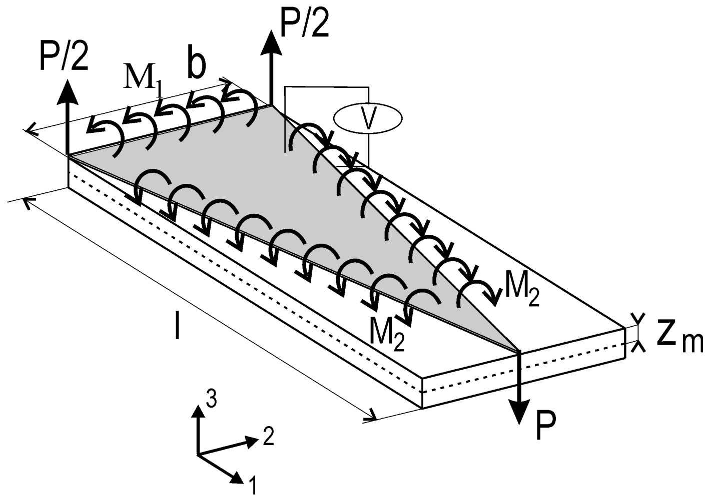


Figure 7.28: Equivalent loads for an orthotropic piezoelectric actuator

and the b/l ratio leading to a point load actuator is

$$\frac{b}{l} = 2\sqrt{\frac{-e_{32}}{e_{31}}} = 0.2015 \tag{7.17}$$

Another possibility is to use a full piezoceramic instead of a composite, which is illustrated below.

7.5.2 Example of a triangular point load actuator

We consider an example similar to the one treated in [13] which consists in a $414mm$ x $314mm$ x $1mm$ aluminum plate clamped on its edges, on which a triangular piezoelectric transducer is fixed on the top surface in order to produce a point load, as illustrated in Figure 7.29. The piezoelectric material used is *SONOX P502* whose properties are given in Table 2.2 with the exception that the value of $\nu_p = 0.4$ in order to be in accordance with the value used in [13]. With these material properties, the b/l ratio to obtain a point load actuator is $b/l = 0.336$ (this will be verified in the example script). The triangular ceramic has a thickness of $180\mu m$ and is encapsulated between two layers of epoxy (see properties in Table 7.1) which have a thickness of $60\mu m$. The basis of the triangle is $33.6mm$ in order to obtain the point load actuator (the height has a length of $100mm$). One triangle is attached to the top surface, and one to the bottom, and the transducers are driven out of phase in order to induce pure bending of the plate and no in-plane motion.

7.5.3 Numerical implementation of the triangular point load actuator

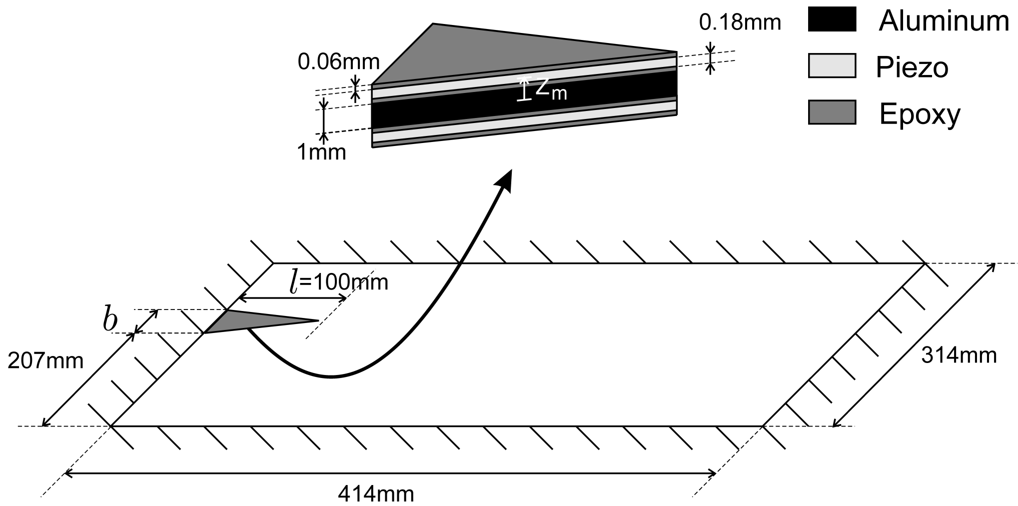


Figure 7.29: Description of the numerical case study for a point load actuator

`d_piezo('TutoPzTriangle-s1')` is used to generate the mesh shown in Figure 7.30.

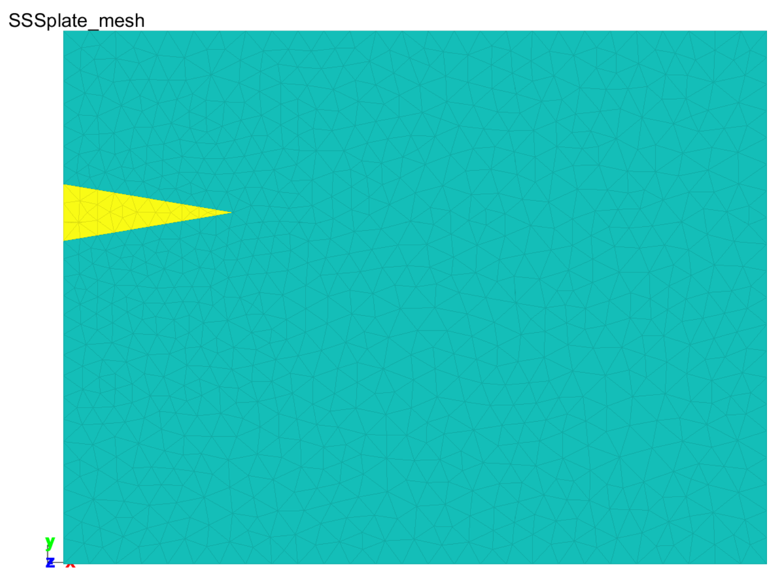


Figure 7.30: Mesh of the aluminum plate with a triangular piezoelectric actuator

Then, in `d_piezo('TutoPzTriangle-s2')` , the static response due to an applied voltage on the piezo actuators is computed and represented on Figure 7.31.

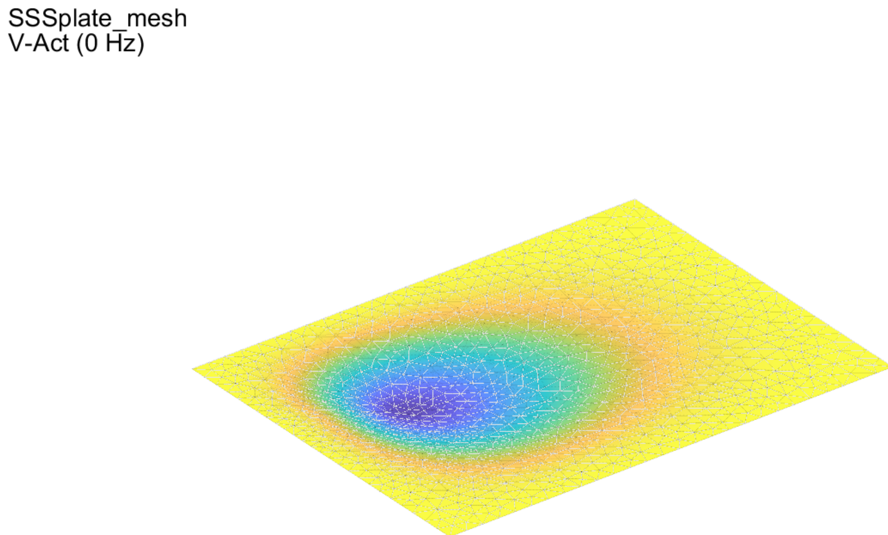


Figure 7.31: Static displacement due to voltage actuation on the triangular piezo

In `d_piezo('TutoPzTriangle-s3')` , we compute and compare the dynamic response of the plate excited with the triangular piezo and a point force whose amplitude is computed using Equation (7.15). The computation is performed with a reduced state-space model using 20 mode shapes:

The amplitude and phase of the vertical displacement at the tip of the triangle are represented for both cases in Figure 7.32, showing the excellent agreement.

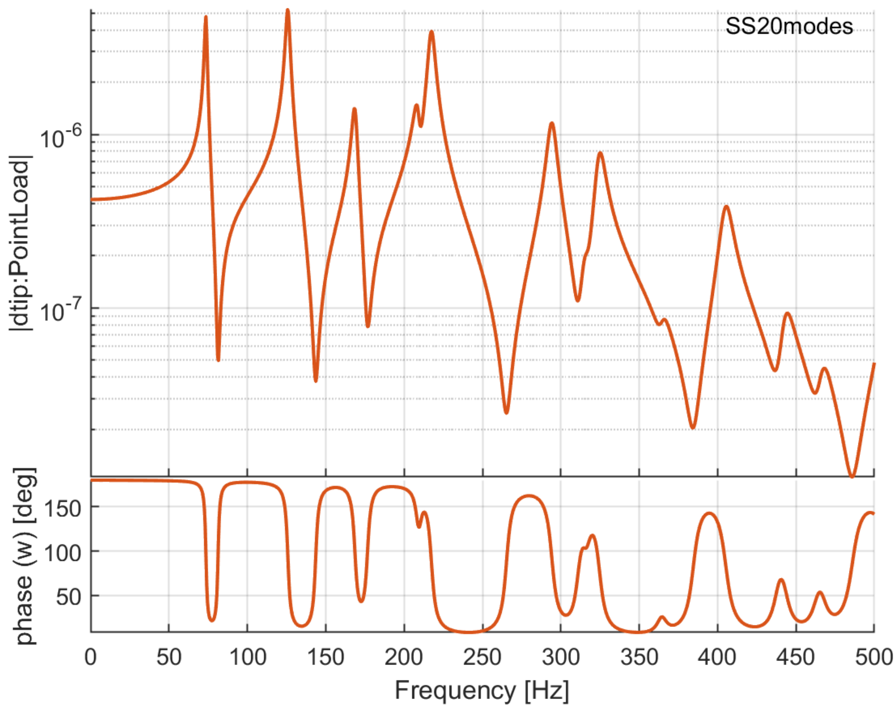


Figure 7.32: Dynamic response at the tip of the triangle due to (i) voltage actuation on the piezo and (ii) point force at the tip of the triangle

8

Other applications

Contents

8.1 Design of an accelerometer and computation of its sensitivity

This application example deals with the determination of the sensitivity of a piezoelectric sensor to base excitation.

8.1.1 Working principle of an accelerometer

By far the most common sensor for measuring vibrations is the accelerometer. The basic working principle of such a device is presented in Figure 8.1(a). It consists of a moving mass on a spring and dashpot, attached to a moving solid. The acceleration of the moving solid results in a differential movement x between the mass M and the solid. The governing equation is given by,

$$M\ddot{x} + c\dot{x} + kx = -M\ddot{x}_0 \tag{8.1}$$

In the frequency domain $x/\ddot{x}_0$ is given by,

$$\frac{x}{\ddot{x}_0} = \frac{-1}{-\omega^2 + \omega_n^2 + 2j\xi\omega\omega_n} \tag{8.2}$$

with $\omega_n = \sqrt{\frac{k}{M}}$ and $\xi = b/2\sqrt{kM}$ and for frequencies $\omega << \omega_n$, one has,

$$\frac{x}{\ddot{x}_0} \simeq \frac{-1}{\omega_n^2} \tag{8.3}$$

showing that at low frequencies compared to the natural frequency of the mass-spring system, x is proportional to the acceleration $\ddot{x}_0$. Note that since the proportionality factor is $\frac{-1}{\omega_n^2}$, the sensitivity of the sensor is increased as ω_n^2 is decreased. At the same time, the frequency band in which the accelerometer response is proportional to $\ddot{x}_0$ is reduced.

The relative displacement x can be measured in different ways among which the use of piezoelectric material, either in longitudinal or shear mode (Figure 8.2). In such configurations, the strain applied to the piezoelectric material is proportional to the relative displacement between the mass and the base. If no amplifier is used, the voltage generated between the electrodes of the piezoelectric material is directly proportional to the strain, and therefore to the relative displacement. For frequencies well below the natural frequency of the accelerometer, the voltage produced is therefore proportional to the absolute acceleration of the base.

8.1.2 Determining the sensitivity of an accelerometer to base excitation

A basic design of a piezoelectric accelerometer working in the longitudinal mode is shown in Figure 8.3. In this example, the casing of the accelerometer is not taken into account, so that the device

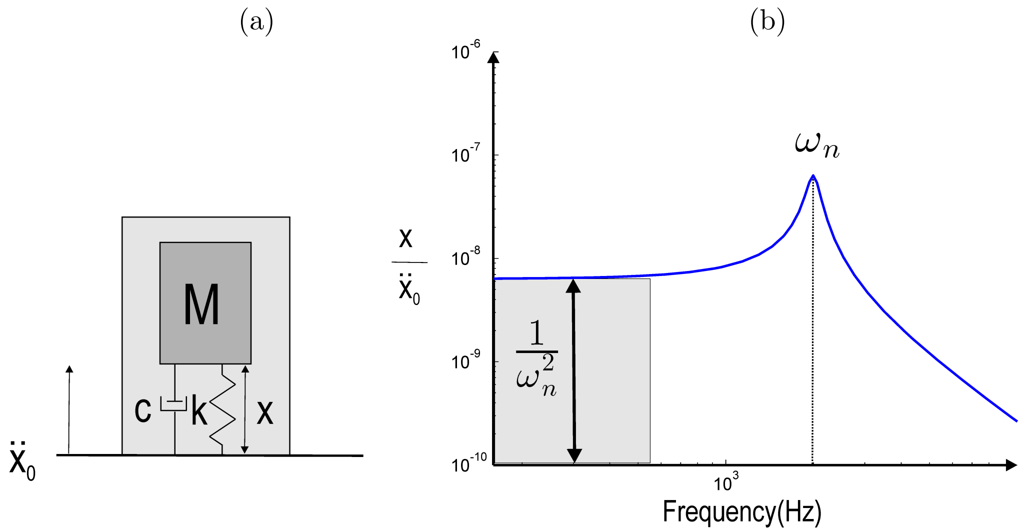


Figure 8.1: Working principle of an accelerometer

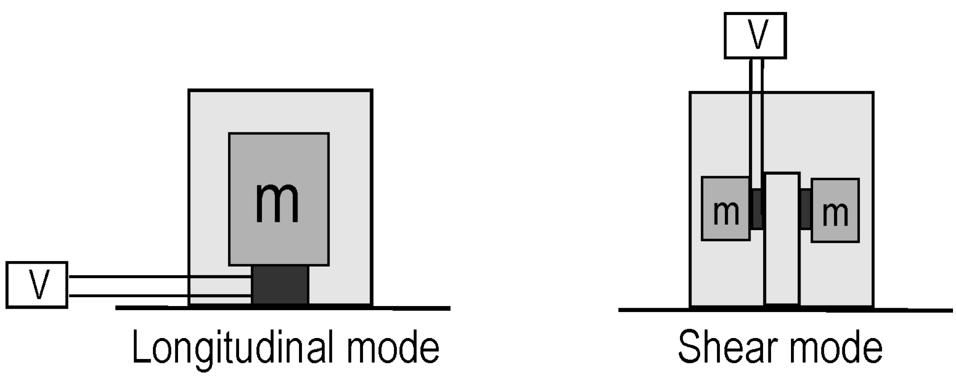


Figure 8.2: Different sensing principles for standard piezoelectric accelerometers

consists in a $3mm$ thick rigid wear plate ($10mm$ diameter), a $1mm$ thick piezoelectric element ($5mm$ diameter), and a $10mm$ thick ($10mm$ diameter) steel proof mass. The mechanical properties of the three elements are given in Table 8.1. The piezoelectric properties of the sensing element are given in Table 8.2 and correspond to *SONOX_P502_iso* property in m_piezo Database. The sensing element is poled through the thickness and the two electrodes are on the top and bottom surfaces.

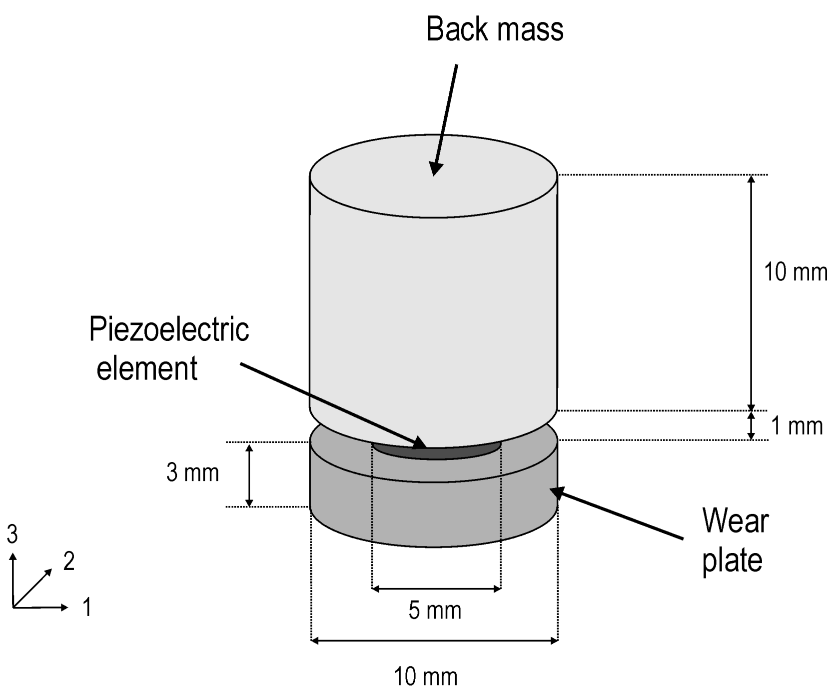


Figure 8.3: Basic design of a piezoelectric accelerometer working in the longitudinal mode

Part	Material	E (GPa)	ρ (kg/m^3)	ν
Wear plate	Al_2O_3	400	3965	0.22
Sensing element	Piezo	54	7740	0.44
Proof mass	Steel	210	7800	0.3

Table 8.1: Mechanical properties of the wear plate, sensing element and proof mass

Property	Value
$d_{31} = d_{32}$	-185 $10^{-12} pC/N$ (or m/V)
d_{33}	440 $10^{-12} pC/N$ (or m/V)
$d_{15} = d_{24}$	560 $10^{-12} pC/N$ (or m/V)
$\varepsilon_{33}^T = \varepsilon_{22}^T = \varepsilon_{11}^T$	1850 ε_0
ε_0	8.854 $10^{-12} F m^{-1}$

Table 8.2: Piezoelectric properties of the sensing element

The sensitivity curve of the accelerometer, expressed in $V/m/s^2$ is used to assess the response of the sensor to a base acceleration in the sensing direction (here vertical). In order to compute this sensitivity curve, one needs therefore to apply a uniform vertical base acceleration to the sensor and to compute the response of the sensing element as a function of the frequency.

The mesh is built and visualized in `d_piezo('TutoPzAccel-s1')` . It is shown in Figure 8.4

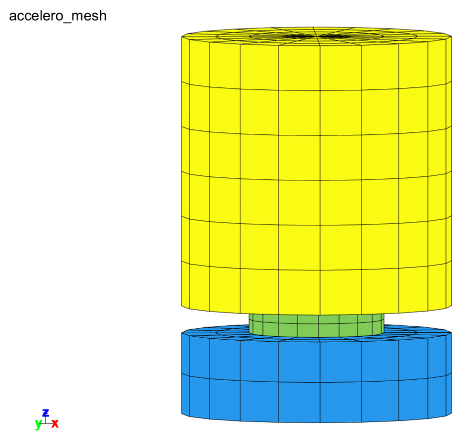


Figure 8.4: Mesh of the piezoelectric accelerometer. The different colors represent the different groups

The call includes the meshing of the accelerometer, the definition of material properties, as well as the definition of electrodes. In addition, the bottom electrode is grounded, and both a voltage and a charge sensor are defined for the top electrode.

In `d_piezo('TutoPzAccel-s2')` , a displacement sensor at the center of the base is defined in order to compute the sensitivity.

```
%% Step 2 - Define sensors and actuators
% -MatID 2 requests a charge resultant sensor
% -vout requests a voltage sensor
model=p_piezo('ElectrodeMPC Top sensor -matid 2 -vout',model,'z==0.004');
% -ground generates a v=0 FixDof case entry
model=p_piezo('ElectrodeMPC Bottom sensor -ground',model,'z==0.003');
% Add a displacement sensor for the basis
model=fe_case(model,'SensDof','Base-displ',1.03);
```

The sensitivity for the sensor used in a voltage mode is then computed using an imposed displacement on the basis. The measured displacement is transformed to acceleration in the frequency domain. It is represented in Figure 8.5 and is computed with the following script:

```
(d_piezo('TutoAccel-s3') )

%% Step 3 - Response with imposed displacement
% Remove the charge sensor (not needed)
model=fe_case(model,'remove','Q-Top sensor');

% Link dofs of base and impose unit vertical displacement
n1=feutil('getnode z==0',model);
rb=feutilb('geomrb',n1,[0 0 0],fe_c(feutil('getdof',model),n1(:,1),'dof'));
rb=fe_def('subdef',rb,3); % Keep vertical displacement
model=fe_case(model,'DofSet','Base',rb);

% Other parameters
f=linspace(1e3,2e5,200)';
model=stack_set(model,'info','Freq',f); % freq. for computation

 % Reduced ss-model
[sys,TR1]=fe2ss('free 5 10 0 -dterm',model,5e-3); %

C1=qbode(sys,f(:)*2*pi,'struct-lab');C1.name='SS-voltage';
C1.X{2}={'Sensor output(V)';'Base Acc(m/s^2)';'Sensitivity (V/m/s^2)'}; %outputs

% C1 compute accel and sensitivity
 C1.Y(:,2)=C1.Y(:,2).*(-(C1.X{1}*2*pi).^2); % Base acc
 C1.Y(:,3)=C1.Y(:,1)./C1.Y(:,2);% Sensitivity

 ci=iiplot; iicom(ci,'curveinit',{'curve',C1.name,C1});  iicom('ch3');
```

```
d_piezo('setstyle',ci)
```

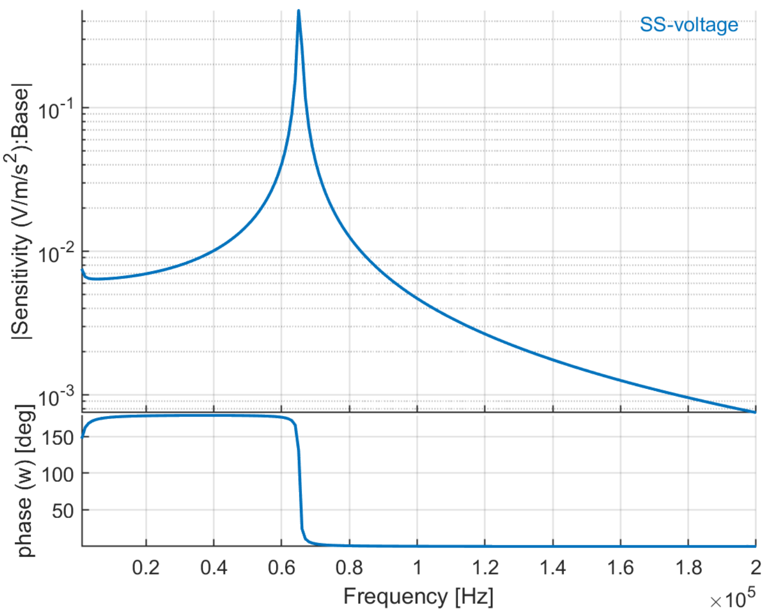


Figure 8.5: Sensitivity of the accelerometer in the voltage mode, uniform acceleration imposed at the base

The sensor can also be used in the charge mode. The following scripts compares the sensitivity of the sensor used in the voltage and charge modes. The sensitivities are normalized to the static sensitivity in order to be compared on the same graph, as the orders of magnitude are very different (Figure 8.6). The charge sensor corresponds to a short-circuit condition which results in a lower resonant frequency than the sensor used in a voltage mode where the electric field is present in section 6.3.4 for a plate. Here the difference of eigenfrequency is however higher (about 10%) due to the fact that there is more strain energy in the piezoelectric element, and that it is used in the d_{33} mode which has a higher electromechanical coupling factor than the d_{31} mode.
(d_piezo('TutoPzAccel-s4'))

```
%% Step 4 - Compare charge and voltage mode for sensing

model=d_piezo('MeshBaseAccel');
model=fe_case(model,'remove','V-Top sensor');

% Short-circuit electrodes of accelerometer
```

```
model=fe_case(model,'FixDof','V=0 on Top Sensor', ...
    p_piezo('electrodedof Top sensor',model));

% Other parameters
  model=stack_set(model,'info','Freq',f);

% Link dofs of base and impose unit vertical displacement
n1=feutil('getnode z==0',model);
rb=feutilb('geomrb',n1,[0 0 0],fe_c(feutil('getdof',model),n1(:,1),'dof'));
rb=fe_def('subdef',rb,3); % Keep vertical displacement
model=fe_case(model,'DofSet','Base',rb);

% Reduced model
[sys2,TR1]=fe2ss('free 5 10 0 -dterm',model,5e-3); %
C2=qbode(sys2,f(:)*2*pi,'struct');C2.name='SS-charge';

C2.X{2}={'Sensor output(V)';'Base Acc(m/s^2)';'Sensitivity (normalized)'}; %outputs
C1.X{2}={'Sensor output(q)';'Base Acc(m/s^2)';'Sensitivity (normalized)'};

% C2 compute accel and sensitivity
 C2.Y(:,2)=C2.Y(:,2).*(-(C2.X{1}*2*pi).^2); % Base acc
 C2.Y(:,3)=C2.Y(:,1)./C2.Y(:,2);% Sensitivity

 % Normalize sensitivity to first freq
 C1.Y(:,3)=C1.Y(:,3)/abs(C1.Y(1,3));
 C2.Y(:,3)=C2.Y(:,3)/abs(C2.Y(1,3))

 ci=iiplot; iicom(ci,'curveinit',{'curve',C1.name,C1;'curve',C2.name,C2}); ...
 iicom('ch3');  d_piezo('setstyle',ci)
```

8.1.3 Computing the sensitivity curve using a piezoelectric shaker

Experimentally, the sensitivity curve can be measured by attaching the accelerometer to a shaker in order to excite the base. Usually, this is done with an electromagnetic shaker, but we illustrate in the following example the use of a piezoelectric shaker for sensor calibration. The piezoelectric shaker consists of two steel cylindrical parts with a piezoelectric disc inserted in between. The base of the shaker is fixed and the piezoelectric element is used as an actuator: imposing a voltage difference between the electrodes results in the motion of the top surface of the shaker to which the accelerometer is attached (Figure 8.7).

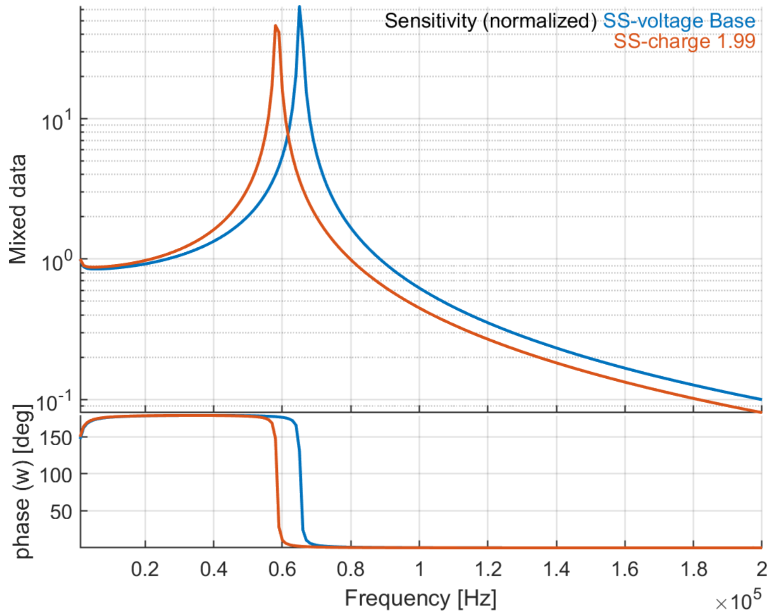


Figure 8.6: Comparison of the normalized sensitivities of the sensor used in the charge and voltage mode

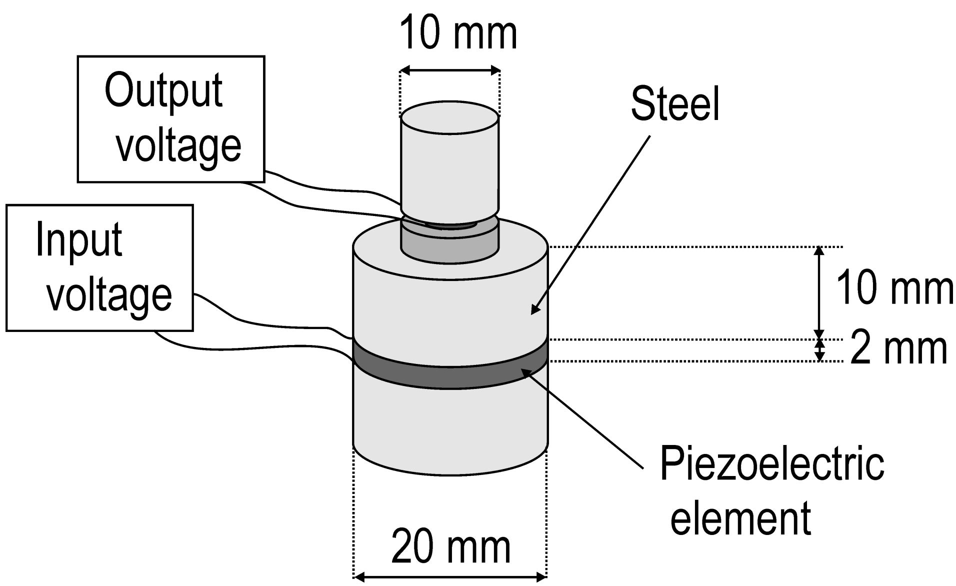


Figure 8.7: Piezoelectric accelerometer attached to a piezoelectric shaker for sensor calibration

The piezoelectric properties for the actuating element in the piezoelectric shaker are identical to the ones of the sensing element given in Table 4.3 and it is poled through the thickness.

In d_piezo('TutoPzAccShaker-s1') , the mesh is produced and visualized (Figure 8.8).

Then in d_piezo('TutoPzAccShaker-s2') , a voltage actuator is defined for the piezoelectric disk in the piezoelectric shaker by setting the bottom electrode potential to zero, and defining the top electrode potential as an input, and an acceleration sensor is defined at the base of the accelerometer.

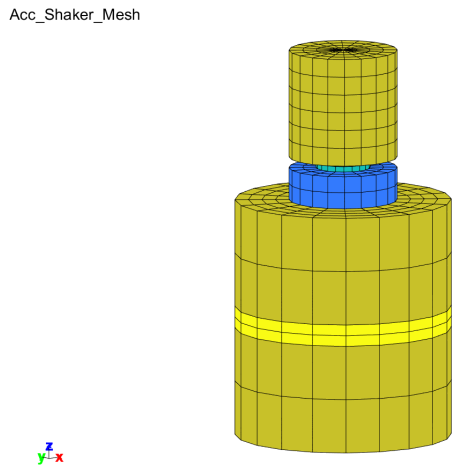


Figure 8.8: Mesh of the piezoelectric accelerometer attached to a piezoelectric shaker

d_piezo('TutoPzAccShaker-s3') is used to compute the response on the sensors (acceleration at the base of the accelerometer, and voltage on the piezo disk in the accelerometer) and to deduce the sensitivity of the accelerometer by dividing the sensor output by the base acceleration.

```
%% Step 3 - Compute response, voltage input on shaker
f=linspace(1e3,2e5,200)';
model=stack_set(model,'info','Freq',f);

% Reduced ss-model
model=fe_case(model,'pcond','Piezo','d_piezo(''Pcond'')');
[sys,TR1]=fe2ss('free 5 45 0 -dterm ',model,1e-3); %

C1=qbode(sys,f(:)*2*pi,'struct-lab');C1.name='SS-voltage';
```

```
C1.X{2}={'Sensor output(V)';'Base Acc(m/s^2)';'Sensitivity (V/m/s^2)'}; %outputs

% Compute sensitivity
C1.Y(:,3)=C1.Y(:,1)./C1.Y(:,2);% Sensitivity

ci=iiplot; iicom(ci,'curveinit',{'curve',C1.name,C1});  iicom('ch3');
d_piezo('setstyle',ci)
```

The sensitivity curve obtained is shown in Figure 8.9. It is comparable to the one computed with imposed displacement around the natural frequency of the accelerometer, but at low frequencies, the flat part is not correctly represented and a few spurious peaks appear at high frequencies. These differences are due to the fact that the piezoelectric shaker does not impose a uniform acceleration of the base of the sensor, and has several resonance frequencies in the band of interest. There are also lateral (contraction) effects due to the piezoelectric element in the shaker which induce a response of the accelerometer and modify the sensitivity curve.

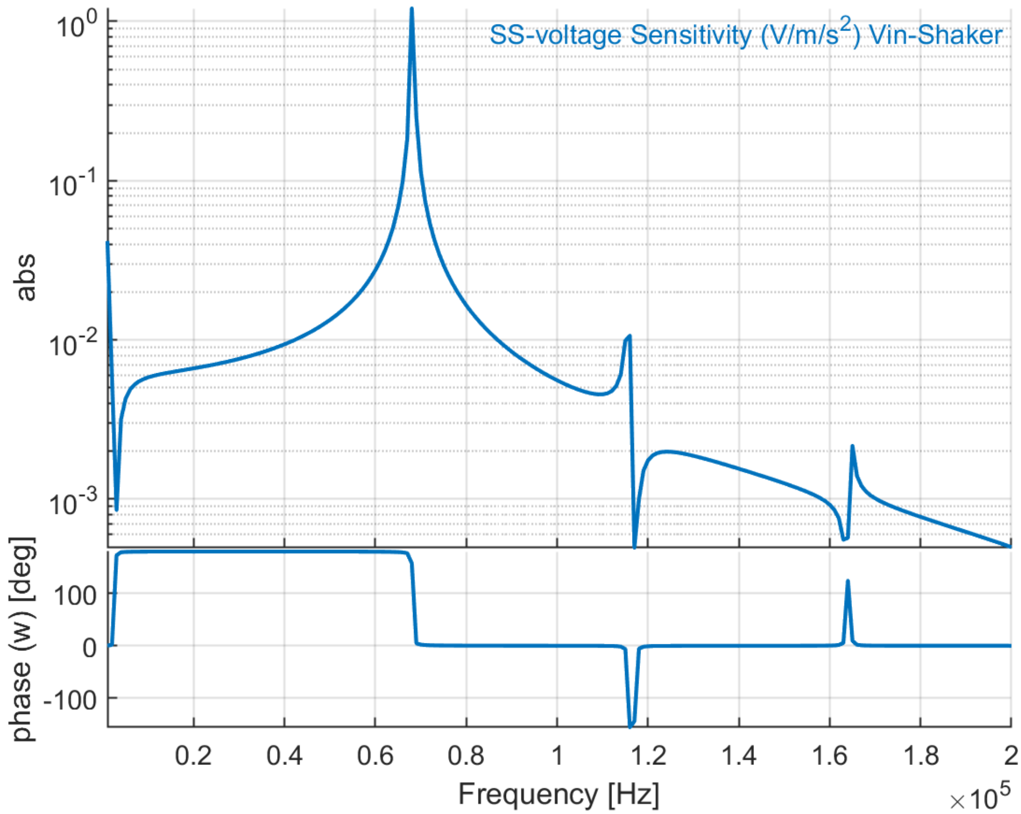


Figure 8.9: Sensitivity curve obtained with a piezoelectric shaker

8.2 Vibration damping using a tuned resonant shunt circuit

8.2.1 Introduction

The idea of damping a structure via a resonant shunt circuit is very similar to the mechanical *tuned mass damper* (TMD) concept. The mechanical TMD is replaced by a 'RL' shunt circuit which, together with the capacitance of the piezoelectric element to which it is attached, acts as a resonant 'RLC' circuit. By tuning the resonance frequency of this circuit to the open-circuit (OC) resonance frequency of the structure equipped with a piezoelectric transducer, one can achieve vibration reduction around the resonant peak of interest. The mechanism is based on the conversion of part of the mechanical energy to electrical energy which is then dissipated in the resistive component of the circuit.

The part of mechanical energy which is converted into electrical energy is given by the generalized electro-mechanical coupling coefficient α_i of the mode i of interest. In practice, this generalized coupling coefficient can be computed based on the open-circuit (OC) and short-circuit (SC) frequencies Ω_i and ω_i of the piezoelectric structure. Experimentally, these frequencies are usually obtained via the measurement of the impedance (V/I) or the capacitance (Q/V) of the piezoelectric structure, and one has:

$$\alpha_i = \frac{\Omega_i^2 - \omega_i^2}{\Omega_i^2} \tag{8.4}$$

Once α_i is known for the mode of interest, the values of R and L can be computed. Several rules exist to compute the optimal values of R and L [14]. We adopt here Yamada's tuning rules which are equivalent to Den Hartog's tuning rules for the mechanical TMD. The first rule aims at tuning the resonant circuit to the OC frequency Ω_i of the piezoelectric structure:

$$\delta = \frac{\Omega_i}{\omega_e} = 1 \tag{8.5}$$

with

$$\omega_e = \sqrt{\frac{1}{LC_{i2}}} \tag{8.6}$$

where C_{i2} is the capacitance of the piezoelectric element attached to the structure taken after the resonant frequency of interest. Note that this value is in practice difficult to measure with precision. In this example, we will take the value which is at the frequency corresponding to the mean value between the SC resonant frequencies ω_i and ω_{i+1}. This first tuning rule allows to compute the value of L. For different values of R, one can show that when plotting the response of the structure to which the resonant shunt has been added for different values of R, all curves cross at two points P and Q which are at the same height (Figure 8.10). The second tuning rule is aimed at finding the optimal value of R which minimizes the response of the structure for the range of frequencies around the natural frequency of interest and is given by:

$$R = \sqrt{\frac{3\alpha_i^2}{2 - \alpha_i^2}} \frac{1}{C_{i2}\Omega_i} \tag{8.7}$$

8.2.2 Resonant shunt circuit applied to a cantilever beam

We illustrate the use of a resonant shunt with the following example of a cantilever beam. The beam has a length of *350 mm*, a width of *25 mm* and a height of *2 mm* (Figure 8.11). Two pairs of piezoelectric *PIC 255* patches of dimensions *50 mm x 25 mm x 0.5 mm* are glued on each side of the beam starting at the cantilever side . The nodes associated to the electrical DOFs of the four patches are numbered respectively [10001 10002] for the patches next to the clamping side, and

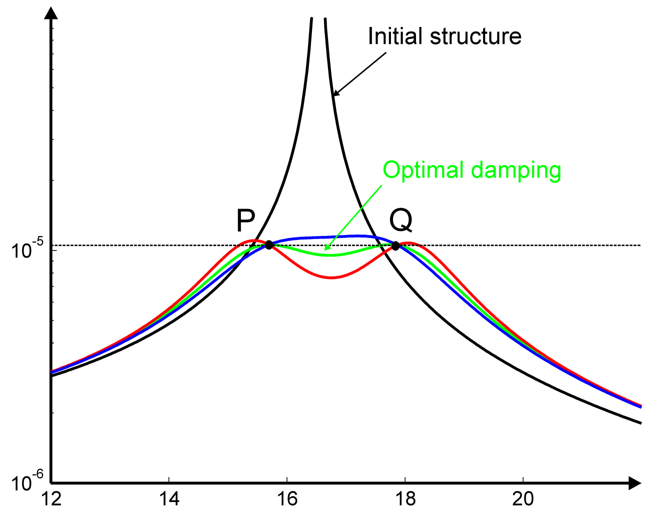


Figure 8.10: Response of the structure with and without and RL shunt : P and Q are at the same height when $\delta = 1$

[20001 20002] for the other pair situated next to it.

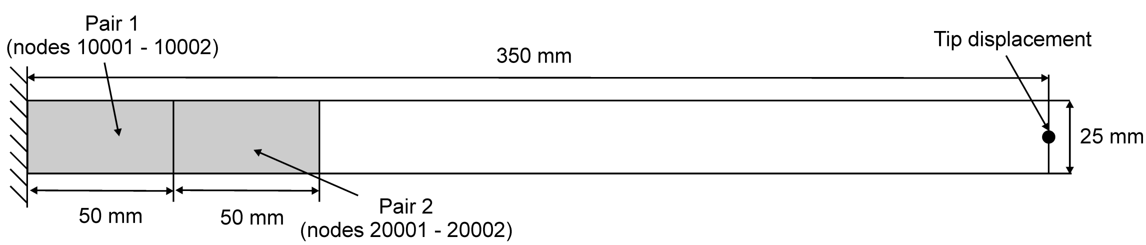


Figure 8.11: Cantilever beam with two pairs of piezoelectric patches

In `d_piezo('TutoPz_shunt-s1')` , we start by generating the mesh (Figure 8.12) and setting the damping in the model.

In `d_piezo('TutoPz_shunt-s2')` , in order to implement the shunt, we will compute the capacitance curve of the first set of patches used in phase opposition (bending) and extract the OC and SC first natural frequency of the cantilever beam. In order to do that, we define two combinations of patches for actuation (bending using the first pair in opposition of phase, and bending using the second pair in opposition of phase), one combination of charge sensors in opposition of phase (to compute the capacitance of the first pair), and one sensor for tip displacement (Figure 8.11).

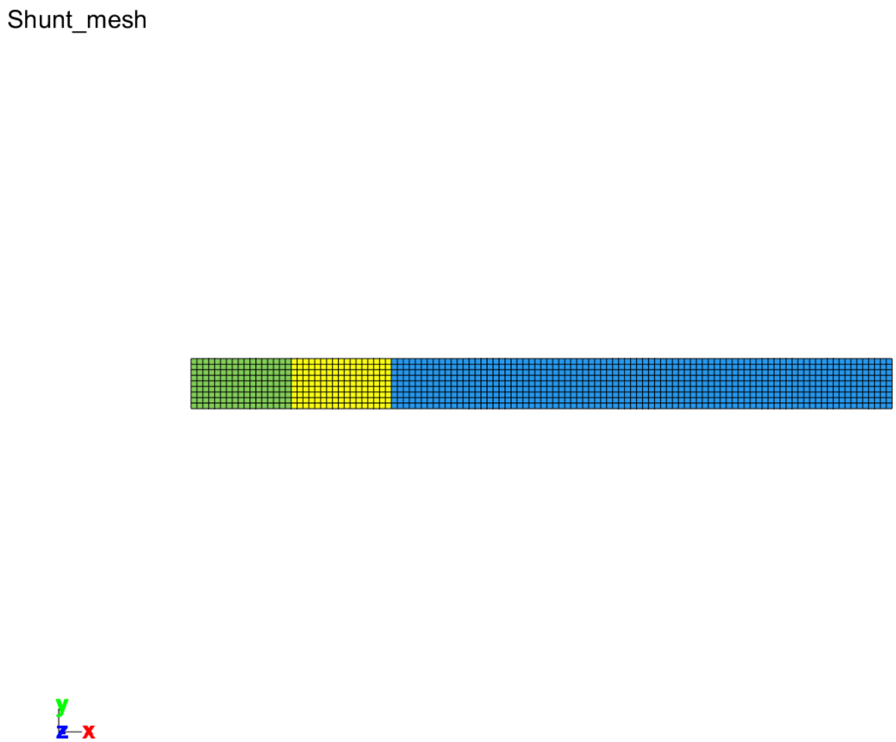


Figure 8.12: Mesh of the cantilever beam showing the two pairs of piezoelectric patches on the left, next to the clamp

In `d_piezo('TutoPz_shunt-s3')` , we can now compute the response of the structure to the two bending actuators using a reduced state-space model with 30 modes. The response of the combination of charge sensors is the capacitance curve (Figure 8.13) of the first pair of piezo patches used in opposition of phase.

In `d_piezo('TutoPz_shunt-s4')` , ω_1 and Ω_1 are extracted from this curve to compute α_1, and C_{12}. Note that because the model is in millimetres, the voltage is expressed in μV and the charge remains in Coulombs, so that the capacitance is expressed in MF (megaFarads) (see `fe_mat('convertSIMM')` for details on the unit conversion when working in millimeters).

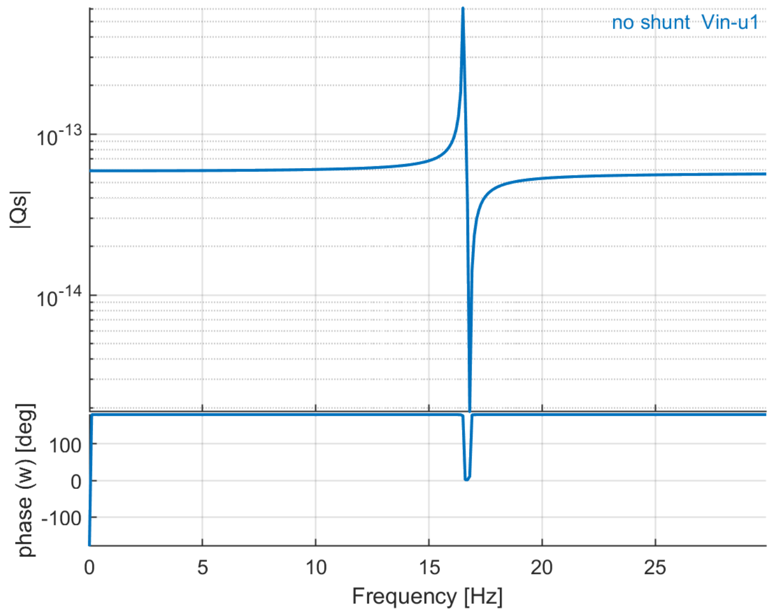


Figure 8.13: Capacitance (Q/V) of the first pair of piezo patches in bending. The resonance corresponds to ω_1 and the anti-resonance to Ω_1. C_{12} is the capacitance in the flat part after the anti-resonance

```
%% Step 4 - Determine parameters for shunt tuning
 % Extract w1 and W1 and compute alpha_1
C=C1.Y(:,1);
% Find poles and zeros of impedance (1/jwC)
if ~exist('findpeaks','file'); warning('Skipping step, signal toolbox');
return;end
[pksPoles,locsPoles]=findpeaks(abs(1./C)); Wi=w(locsPoles);
[pksZeros,locsZeros]=findpeaks(abs(C)); wi=w(locsZeros);
% concentrate on mode of interest (mode 1)
 W1=w(locsPoles(1)); w1=w(locsZeros(1));
% Compute alpha for mode of interest
a1=sqrt((W1^2-w1^2)/W1^2);

% Compute Cs2 for mode of interest
ind1=1; i2=locsZeros(1); i3=locsZeros(2);
dw2=w(i3)-w(i2); wCs2=w(i2)+dw2/2;
[y,i]=min(abs(w-wCs2)); Cs2=abs(C(i));
```

```
%% Determine shunt parameters (R and L) and apply it to damp 1st mode
% Tuning using Yamada's rule
d=1; r=sqrt((3*a1^2)/(2-a1^2));
L_Yam=1/d^2/Cs2/W1^2; R_Yam=r/Cs2/W1;
```

In (`d_piezo('TutoPz_shunt-s5')`), we represent the FRF of the tip displacement due to bending actuation on the second pair of piezos for the initial system and the system with the shunt (Figure 8.14). The shunt is implemented using the *feedback* function of the *Control toolbox*.

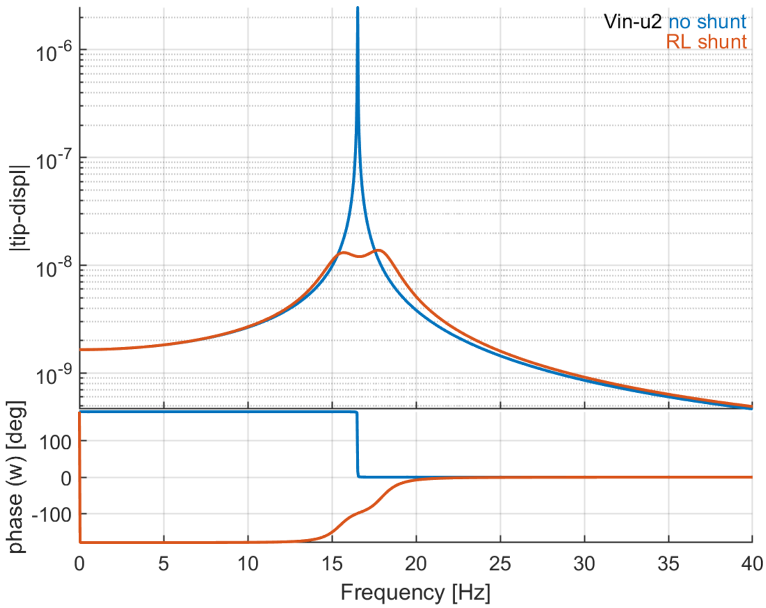


Figure 8.14: Tip displacement due to the bending actuator of the second pair of piezo patches with and without shunt

```
%% Step 5 - Compute dynamic response with optimal shunt
w=linspace(0,40,1e3)*2*pi;
C1=qbode(sys,w,'struct'); C1.name='no shunt';
C1.X{2}={'Qs';'tip-displ'}; %outputs
C1.X{3}={'Vin-u1';'Vin-u2'}; %inputs

% Implement shunt using feeback - requires control toolbox - compute FRF
if ~exist('feedback','file'); warning('Skipping step, control toolbox');
```

```
  return;
end
A=tf([L_Yam R_Yam 0],1); % RL shunt in tf form
sys2=feedback(sys,A,1,1,1);
% qbode does not work with feedback so use freqresp from control toolbox
 C=freqresp(sys2,w); a=C(:); C2=C1; C2.Y=reshape(a,4,1000)';
 C2.name='RL shunt'; C1.X=C2.X;

% Plot and compare curves
iicom(ci,'curveinit',{'curve',C1.name,C1;'curve',C2.name,C2});
iicom(ci,'ch 4'); d_piezo('setstyle',ci);
```

Function reference

Contents

Piezo related functions	
m_piezo	piezoelectric material property handling
p_piezo	piezoelectric volume and shell property handling
d_piezo	support for demonstration of piezo capabilities

m_piezo

Purpose

Material function for piezoelectric solids

Syntax

```
mat= m_piezo('database name')
pl = m_piezo('dbval MatId  -elas 12 Name');
```

See d_piezo Tuto for tutorial calls. Accepted commands are

[Database, Dbval] [-unit TY] [,MatiD]] Name

m_piezo contains a number of defaults obtained with the database and dbval commands which respectively return a structure or an element property row. You can select a particular entry of the database with using a name matching the database entries.
Piezoelectric materials are associated with two material identifiers, the main defines the piezoelectric properties and contains a reference ElasMatId to an elastic material used for the elastic properties of the material (see m_elastic for input formats).

```
m_piezo('info') % List of materials in data base
% database piezo and elastic properties
pl=m_piezo('dbval 3 -elas 12 SONOX_P502_iso')
```

Theoretical details on piezoelectric materials are given in section 2 . The m_piezo Const and BuildConstit commands support integration constant building for piezo electric volumes integrated in the standard volume elements. Element properties are given by p_solid entries, while materials formats are detailed here.

Patch

Supports the specification of a number of patches available on the market. The call uses an option structure with fields

- .name of the form ProId*val+patchName*. For example ProId1+SmartM.MFC-P1.2814.
- MatId value for the initial MatId.

m_piezo('patch') lists currently implemented geometries. In particular

- Disk.Material.Geometry is used for generic circular patches. (Legacy Noliac.Material.Geometry is used for circular patches by Noliac). The list of materials in the database is obtained with m_piezo('info'). Fields for the geometry are

  - OD outer diameter (mm). This can be replaced by RC.
  - TH Thickness (mm). To specify a milimiter fraction replace the . by and _. For example TH0_7 is used for TH=0.7 mm.
  - ID inner diameter (mm) (optional for piezo rings).
  - LC characteristic length for meshing (when mesher has hability to use the information)

- Rect.Material.Geometry is used for generic rectangular patches. SmartM.Material.Geometry is used for rectangular patches by Smart Materials. Fields for the geometry are

  - WI width (mm)
  - LE length (mm)
  - TH Thickness (mm). To specify a milimiter fraction replace the . by and _. For example TH0_7 is used for TH=0.7 mm.
  - LC characteristic length for meshing (when mesher has hability to use the information)

  Thus WI28LE14TH_2 is a 28 by 14 by 0.2 mm patch. The geometry can also be coded as 2814TH_2. See d_piezo('TutoPzMeshingAuto') for illustration.

The piezoelectric constants can be declared using the following sub-types

1:Simplified 3D piezoelectric properties

`[ProId Type ElasMatId d31 d32 d33 eps1T eps2T eps3T EDType]`
These simplified piezoelectric properties (3.5) can be used for PVDF, but also for PZT if shear mode actuation/sensing is not considered ($d_{24} = d_{15} = 0$). For EDType==0 on assumes d is given. For EDType==1, e is given. Note that the values of ε^T (permitivity at zero stress) should be given (and not ε^S).

2:General 3D piezo

`[ProId Type ElasMatId d_1:18 epsT_1:9]`
d_1:18 are the 18 constants of the $[d]$ matrix (see section 2.1), and epsT_1:9 are the 9 constants of the $\left[\varepsilon^T\right]$ matrix. One reminds that strains are stored in order xx, yy, zz, yz, zx, yx.

3 : General 3D piezo, e matrix

`[ProId Type ElasMatId e_1:18 epsT_1:9]`
e_1:18 are the 18 constants of the $[d]$ matrix, and epsT_1:9 are the 9 constants of the $\left[\varepsilon^T\right]$ matrix in the constitutive law (see section 2.1).

See also

p_piezo.

p_piezo

Purpose

Property function for piezoelectric shells and utilities associated with piezoelectric models.

Syntax

```
mat= m_piezo('database name')
pl = m_piezo('dbval MatId  -elas 12 Name');
```

See `d_piezo Tuto` for tutorial calls. Accepted commands are

ElectrodeMPC

`[model,InputDOF(end+1,1)]=p_piezo('ElectrodeMPC Name',model,'z==5e-5');` defines the isopotential constraint as a case entry `Name` associated with `FindNode` command `z==5e-5`. An illustration is given in section 8.1.2 .
Accepted command options are

- `-Ground` defines a fixed voltage constraint `FixDof,V=0 on` *Name*.
- `-Input"InName"` defines an enforced voltage `DofSet,`*InName* entry for voltage actuation.
- `MatId`*i* is used to define a resultant sensor to measure the charge associated with the electrode. Note that the electrode surface must not be inside the volume with `MatId`*i*. If that is the case, you must arbitrarily decompose your mesh in two parts with different `MatId`. You can also generate this sensor a posteriori using `ElectrodeSensQ`, which attempts to determine the `MatId`*i* based on the search of a piezoelectric material connected to the MPC.

ElectrodeSensQ

`model=p_piezo('ElectrodeSensQ 1682 Q-Base',model);` adds a charge sensor (`resultant`) called `Q-Base` on node `1682`. (See (3.10) for theory).
For **shells**, the node number is used to identify the `p_piezo` shell property and thus the associated elements. It is reminded that `p_piezo` entries must be duplicated when multiple patches are used.
For **volumes**, the `p_piezo ElectrodeMPC` should be first defined, so that it can be used to obtain the electrode surface information.
Note that the command calls `fe_case('SensMatch')` so that changes done to material properties after this call will not be reflected in the observation matrix of this sensor.
To obtain sensor combinations (add charges of multiple sensors as done with specific wiring), specify a data structure with observation `.cta` at multiple `.DOF` as illustrated below.
For a voltage sensor, you can simply use a DOF sensor
`model=fe_case(model,'SensDof','V-Base',1682.21)`.

```
model=d_piezo('meshULBPlate cantilever');  % creates the model
% If you don't remember the electrode node numbers
r2=p_piezo('ElectrodeDOF',model)
% Combined charge
r1=struct('cta',[1 1],'DOF',[r2{1:2,2}]+.21,'name','QS2+3');
model=p_piezo('ElectrodeSensQ',model,r1);
sens=fe_case(model,'sens');
% Combined voltage
r1=struct('cta',[1 1],'DOF',[r2{3:4,2}]+.21,'name','VS2+3');
model=fe_case(model,'SensDof',r1.name,r1);
sens=fe_case(model,'sens');sens.lab
```

ElectrodeDOF

`p_piezo('ElectrodeDof Bottom',model)` returns the DOF the bottom electrode. With no name for selection `p_piezo('ElectrodeDof',model)` the command returns the list of electrode DOFs based on MPC defined using the `ElectrodeMPC` command or `p_piezo` shell entries. Use `ElectrodeDof.*` to get all DOFs.

ElectrodeView ...

`p_piezo('electrodeview',cf)` outlines the electrodes in the model and prints a clear text summary of electrode information. To only get the summary, pass a model `model` rather than a pointer `cf` to a `feplot` figure.

`p_piezo('electrodeviewCharge',cf)` builds a `StressCut` selection allowing the visualization of charge density. You should be aware that only resultant charges at nodes are known. For proper visualization a transformation from charge resultant to charge density is performed, this is known to have problem in certain cases so you are welcome to report difficulties.

Electrode2Case

`Electrode2Case` uses electrode information defined in the obsolete `Electrode` stack entry to generate appropriate case entries : `V_In` for enforced voltage actuators, `V_Out` for voltage measurements, `Q_Out` for charge sensors.

ElectrodeInit

`ElectrodeInit` analyses the model to find electric master DOFs in piezo-electric shell properties or in MPC associated with volume models.

Tab

Tab commands are used to generate tabulated information about model contents. The calling format is p_piezo('TabDD',model). With no input argument, the current feplot figure is used. Currently generated tabs are

- TabDD constitutive laws
- TabPro material and element parameters shown as java tables.

View

p_piezo('ViewDD',model) displays information about piezoelectric constitutive laws in the current model.
p_piezo('ViewElec ...',model) is used to visualize the electrical field. An example is given in section 7.2 . Command options are DefLen*val* to specify the arrow length, EltSel*val* for the selection of elements to be viewed, Reset to force reinit of selection.
ViewStrain and ViewStress follow the same calling format.

Shell element properties

Piezo shell elements with electrodes are declared by a combination of a mechanical definition as a layered composite, see p_shell 2, and an electrode definition with element property rows of the form
[ProId Type MecaProId ElNodeId1 LayerId1 UNU1 ElNodeId2...]

- Type typically fe_mat('p_piezo','SI',1)
- MecaProId : ProId for mechanical properties of element p_shell 2 composite entry. The MatId*i* for piezo layers must be associated with piezo electric material properties.
- ElNodId1 : NodeId for electrode 1. This needs to be a node declared in the model but its position is not used since only the value of the electric potential (DOF 21) is used. You may use a node of the shell but this is not necessary.
- LayerId : layer number as declared in the composite entry.
- UNU1 : currently unused property (angle for polarization)

The constitutive law for a piezoelectric shell are detailed in section 3.2 . The following gives a sample declaration.

```
model=femesh('testquad4'); % Shell MatId 100 ProdId 110

% MatId 1 : steel, MatId 12 : PZT elastic prop
```

```
model.pl=m_elastic('dbval 1  Steel');
% Sonox_P502 piezo material, sdtweb m_piezo('Sonox_P502')
model.pl=m_piezo(model.pl,'dbval 3 -elas 12 SONOX_P502');

% ProId 111 : 3 layer composite (mechanical properties)
model.il=p_shell(model.il,['dbval 111 laminate ' ...
    '3 1e-3 0 ' ...  % MatID 3 (PZT),   1 mm piezo, 0
    '1 2e-3 0  ' ... % MatID 1 (Steel), 2 mm
    '3 1e-3 0']);    % MatID 3 (PZT),   1 mm piezo, 0
% ProId 110 : 3 layer piezo shell with electrodes on nodes 1682 and 1683
model.il=p_piezo(model.il,'dbval 110 shell 111 1682 1 0 1683 3 0');

p_piezo('viewdd',model) % Details about the constitutive law
p_piezo('ElectrodeInfo',model) % Details about the layers
```

d_piezo

Purpose

Support function for piezoelectric demos.

`d_piezo` contains a set of demonstration and meshing scripts. Use `d_piezo` to display tag list and see available content.

Mesh

In order to call a specific meshing script, use `d_piezo('Mesh...',options)`, for example

MeshPlate

Meshing utilities for the placement of piezoelectric patches on a supporting structure (flat plate for now).
The options are specified in a structure with fields

- `.list` : defines a list of features to be introduced with columns giving `name,LamSpec,Geo`, name laminate specification and shape options.
- `.unit` : gives the model unit (needed since patch dimensions are always given in mm).

The laminate specification string is composed of the following

- `BaseId`*i* gives the `ProId` of the base laminate which is then used to figure out the position of patches.
- `+Patch` or `-Patch` to place a patch above or below the base laminate.
- `Patch` itself is a specification of a patch material and geometry. The list of implemented patch can be obtained using `m_piezo Patch`
- `.In` to specify that the patch has an enforced voltage.

The geometry/position specification string `Geo` can be

- a specification of the patch corner and orientation such as `xc=.03 yc=.05 ang=30` if the patch geometry is specified using the laminate specification.
- `rect` shapes `[xc lx nx yc ly ny alpha MatId ProId]` where `MatId` and `ProId` are filled automatically if not provided.
- `circ` shapes `[xc yc rc lc (MatId ProId)]`

- global lx=.4 ly=.3 lc=.02 -Sens. The option -Sens generates a sensor entry corresponding to normal displacement of the initial mesh. Alternatively you can add a sensor configuration SensDOF entry see sdtweb('sensor#scell') or sdtweb('sensor#sstruct').

```
% Start by defining properties of the underlying laminate
mdl=struct('Node',[],'Elt',[], ... % empty model
    'pl', ... % composite layer property
     [1 fe_mat('m_elastic','SI',1) 42.5e9 .042 1490 3.35e9 .01], ...
     'il', ... % laminate definition (6 layers at 0,90,0,90,0,90)
     p_shell(['dbval 1 laminate    1 2.167e-4 0    1 2.167e-4 90 ' ...
         '1 2.167e-4 0 1 2.167e-4 90 1 2.167e-4 0 1 2.167e-4 90']), ...
     'unit','SI');
RG=struct;
RG.list={'Name','Lam','shape'
   'Main_plate', mdl,'global lx=.4 ly=.3 lc=.02'
   'Act1','BaseId1 +SmartM.MFC-P1.2814 -SmartM.MFC-P1.2814.in','xc=.35 yc=.25 ang=30'
   'Sen2','BaseId1 +SmartM.MFC-P1.2814','xc=.03 yc=.05 ang=30'
   'Sen3','BaseId1 +Noliac.NCE51.OD25TH1','xc=.05 yc=.25'
   };
cf=feplot;d_piezo('MeshPlate',RG);cf.mdl.name='Plate with piezo';
p_piezo('electrodeinfo',cf.mdl.GetData)
matgui('jil',cf);matgui('jpl',cf); % Display properties
```

See d_piezo('TutoPzMeshingAuto') for more illustrations.

MeshBaseAccel

Other meshing scripts do not have options, and simply produce the mesh for a specific tutorial script (see below). Run for example

```
model=d_piezo('MeshBaseAccel')}
```

without options. If no output if specified, the mesh is generated and plotted in the feplot window.

Tuto

All the tutorial scripts described in the documentation are available, and can be accessed either in command mode or through a gui interface. To have access to the gui, run d_piezo('tuto') and expand the demo script (+), the arrows allow to run the different steps separately.

To run a demo script in command line, just run d_piezo('Scriptname'), or d_piezo('Scriptname-sn') for a specific step (replace n by the step needed).

See section 1.1 for more details.

Bibliography

[1] J. Yang, *An introduction to the theory of piezoelectricity.* Springer, 2010.

[2] "IEEE standards on piezoelectricity, ans n° 176-187, IEEE," 1988.

[3] A. Deraemaeker and H. Nasser, "Numerical evaluation of the equivalent properties of Macro Fiber Composite (MFC) transducers using periodic homogenization," *International Journal of Solids and Structures*, vol. 47, pp. 3272–3285, 2010.

[4] R. J. Craig, "A review of time-domain and frequency domain component mode synthesis methods," *Int. J. Anal. and Exp. Modal Analysis*, vol. 2, no. 2, pp. 59–72, 1987.

[5] "Multiple-support seismic analysis of large structures," *Computers and Structures*, vol. 36, no. 6, pp. 1153–1158, 1990.

[6] R. Guyan, "Reduction of mass and stiffness matrices," *AIAA Journal*, vol. 3, p. 380, 1965.

[7] R. J. Craig and M. Bampton, "Coupling of substructures for dynamic analyses," *AIAA Journal*, vol. 6, no. 7, pp. 1313–1319, 1968.

[8] G. Raze, C. Dumoulin, and A. Deraemaeker, "Reduced-order state-space models of structures with imposed displacements and accelerations," *Mechanical Systems and Signal Processing*, vol. 191, p. 110156, 2023.

[9] W. Wilkie, R. Bryant, J. High, R. Fox, R. Hellbaum, A. Jalink, B. Little, and P. Mirick, "Low-cost piezocomposite actuator for structural control applications," in *Proc. SPIE 7th Annual Int. Symp. Smart. Struct. Mater.*, (Newport Beach, USA), 2000.

[10] A. Deraemaeker, H. Nasser, A. Benjeddou, and A. Preumont, "Mixing rules for the piezoelectric properties of Macro Fiber Composites," *Journal of Intelligent Material Systems and Structures*, vol. 20(12), pp. 1391–1518, 2009.

[11] N. Hagood, R. Kindel, K. Ghandi, and P. Gaudenzi, "Improving transverse actuation of piezoceramics using interdigitated surface electrodes," in *Proc. SPIE Vol. 1917, p. 341-352, Smart*

Structures and Materials 1993: Smart Structures and Intelligent Systems, Nesbitt W. Hagood; Ed. (N. W. Hagood, ed.), vol. 1917, pp. 341–352, 1993.

[12] A. Deraemaeker, G. Tondreau, and F. Bourgeois, "Equivalent loads for two-dimensional distributed anisotropic piezoelectric transducers with arbitrary shapes attached to thin plate structures," *Journal of the Acoustical Society of America*, vol. 129(2), pp. 681–690, 2011.

[13] G. Tondreau, S. Raman, and A. Deraemaeker, "Point load actuation on plate structures based on triangular piezoelectric patches," *Smart Structures and Systems*, vol. 13(4), pp. 547–565, 2014.

[14] P. Soltani, G. Kerschen, G. Tondreau, and A. Deraemaeker, "Piezoelectric vibration damping using resonant shunt circuits: an exact solution," *Smart Materials and Structures*, vol. 23, 2014. 125014.

Index