



**HAL**  
open science

# Topology Optimization of Industrial Robots: Application to a Five-bar Mechanism

Sébastien Briot, Alexandre Goldsztejn

► **To cite this version:**

Sébastien Briot, Alexandre Goldsztejn. Topology Optimization of Industrial Robots: Application to a Five-bar Mechanism. *Mechanism and Machine Theory*, 2018, 120, pp.30-56. 10.1016/j.mechmachtheory.2017.09.011 . hal-01587979

**HAL Id: hal-01587979**

**<https://hal.science/hal-01587979v1>**

Submitted on 24 Jun 2019

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

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

# Topology Optimization of Industrial Robots: Application to a Five-bar Mechanism

Sébastien Briot<sup>a,\*</sup>, Alexandre Goldsztejn<sup>a</sup>

<sup>a</sup>*Laboratoire des Sciences du Numérique de Nantes (LS2N)  
UMR CNRS 6004 – 1 rue de la Noë, 44321 Nantes, France*

---

## Abstract

Recent works introduced topology optimization in the design of robots, but the proposed methodologies led to a local optimization of the performance. Moreover, most of performance indices used are not in strong relation with easy-to-understand technological requirements.

We propose a methodology that is able to perform a topology optimization for robots, valid globally in the workspace or for a set of given trajectories, and which is based on the use of technology-oriented performance criteria. In order to enforce the chosen performance indices to be valid globally, optimal robot configurations or trajectories for which extreme performance will be attained are computed, and iteratively updated.

In order to decrease the computational time associated with these performance indices, we exploit the structure of the elastic models in order to reduce their computational complexity.

Finally, we use an optimization algorithm called the Linearization Method which gives results in a computational time equivalent to standard topology optimization algorithms, but its implementation is less complex and makes it quite easy to perform modification or improvement.

The methodology is applied for the design of a five-bar mechanism. We show that our approach led to a robust optimization of the robot performance over the whole workspace.

*Keywords:* Robot design, deformation, natural frequency, topological optimization, linearization method.

---

## 1. Introduction

Robots are expected to perform a large variety of tasks. However, it is not wise to believe that a single robot will be able to achieve all conceivable tasks. Inherent robot limitations arise from its own physical performance (accuracy, deformation, vibrations, etc.), which are a combination of the performance of the mechanical architecture and of the controller.

Good performance of the mechanical architecture can be obtained via optimal design [1]. The usual design methodology proposed by French [2] is illustrated in Fig. 1. The first step is to analyze the need in order to formulate the design problem. The second phase focuses on the preliminary design and aims to synthesize design concepts (for instance, new types of robot architectures [3, 4]) and to select the best design alternatives with respect to given criteria (e.g. complexity [5], singularity [6]). The third phase, denoted as the advanced design phase or embodiment of schemes, deals with the computation of the dimensions and shapes of the product element in order to fulfill performance criteria in terms of:

- geometric performance: e.g. workspace size and shape under joint limitations, link collisions [7, 8], accuracy under input errors [9], under clearance [10, 11] or link manufacturing errors [12],
- kinematics / kinetostatics: e.g. velocity transmission ratio [13, 14, 15], effort transmission [16, 17, 18, 19, 20, 21],

---

\*Corresponding author

*Email addresses:* [Sebastien.Briot@ircyn.ec-nantes.fr](mailto:Sebastien.Briot@ircyn.ec-nantes.fr) (Sébastien Briot),  
[Alexandre.Goldsztejn@ircyn.ec-nantes.fr](mailto:Alexandre.Goldsztejn@ircyn.ec-nantes.fr) (Alexandre Goldsztejn)

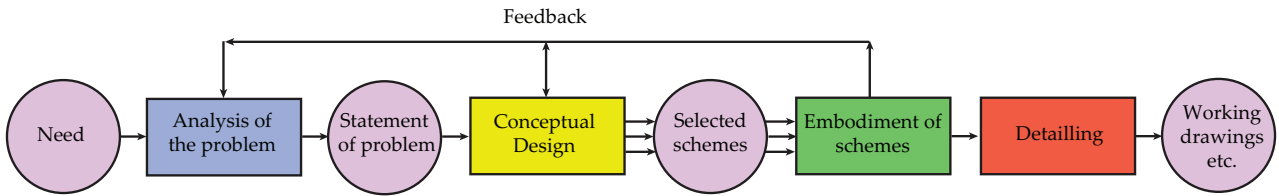


Figure 1: French design process [2]

- dynamics: e.g. moving mass reduction [22], maximal input torques [23], static [24] or dynamic balancing [25, 26] conditions, decoupled structure of the dynamics equations [27, 28],
- elasticity: static or dynamic deformations, natural frequencies and vibrations [29, 30, 31].

The fourth phase is the detailed design stage and consists in obtaining the working drawings of the product elements, in synthesizing their dimensional and geometric tolerances [32], and in manufacturing the prototypes.

The design optimization problem treated in the advanced design phase is usually formulated as a multicriteria optimization problem and it is most of the time solved in cascade in order to reduce its complexity [22]. In a first step, the multicriteria optimization problem takes only into account geometric, kinematic and kinetostatic constraints and objectives and allows for fixing the primary geometric parameters of the robot (lengths of links, angles between the joint axes etc.) [33]. In a second step, the secondary geometric parameters are found (size of the link cross-sections, link mass distribution, or more generally link shapes) taken into account dynamic, elastostatic and elastodynamic aspects [23].

The link shape optimization of robots is probably the most time-consuming step of the optimal design process. This is due to the complexity of the model involved, especially the elastic models, which must be computed thousands of times (and even more) in order to calculate the robot elastic performance in many robot configurations for a given set of design variables [22]. This is necessary in order to ensure that the performance can be guaranteed in a wide range of robot configurations [15]. As a result, in order to decrease the time of computation, a common approach is to reduce the number of design variables. It can be easily reduced by doing a parametric optimization [22, 31], i.e. by modeling links using beam theory [34] and by considering that the geometry of the beam cross-sections is fixed (for instance, circle, square, rectangle, I-shape) but parameterized by a limited number of variables (radius for circles, edge lengths for squares, rectangles, I-shapes).

This approach is known not to be the more accurate for finding the optimal design of links, contrary to topology optimization [35]. Topology optimization was for instance used for the design of compliant mechanisms [36, 37, 38]. This latter technique aims at optimizing the material distribution in a link in order to satisfy performance criteria: a classical problem met in the literature is to minimize the link mass under compliance constraints [39]. The link shape is meshed, and deformation and vibration models are computed using Finite Element Methods (FEM) [40]. The presence of one element of material is parameterized by a design variable varying from 0 to 1, 1 means that there is material while 0 represents a void. As a result, in order to have a refined prediction of the link behavior and a refined visualization of the link shape, this method usually leads to a vector of design variables in the optimization process containing dozens of thousands of components. Topology optimization is thus most of the time computationally expensive due to both the complexity of the models involved and the high number of design variables. Therefore, it is few used in robot design.

However, recent works introduced this technique in the design of robots. First attempts optimized the robot topology for a single loading case. For instance, [41] optimized the torso of a humanoid robot with an objective of minimal mass under compliance constraints while [42, 43] optimized the pelvis of a walking robot. [44] employed topology optimization method to develop stiff and light frames of the lower body for stable walking of their humanoid robot. [45, 46] both optimized the shape of the upper-arm for a 6-degrees-of-freedom (dof) industrial robot. [47] improves the shape of a chassis for a mobile robot.

Optimizing the robot link shapes for a single loading case does not take into account the intrinsic nature of a robot whose performance varies with the configuration, the loading and with the time. Therefore other

works proposed alternative approaches. For instance, [48, 49] considered a given set of reference control signals for the motion of the robot arm and look at some robot performance, such as overshoot, controller settling time, number of oscillations, final deviation or also actuator energy consumption. [50] optimized the link shapes of 3-dof robots under varying configurations while addressing the problem of the reduction of the computational time by dividing the optimization problem into subproblems with lower computational complexity. They minimized the strain energy while constraining the robot mass. [51] optimized the shape of the pelvis of a humanoid robot by using equivalent static loads with the objective to minimize the strain energy.

All these works introduced the variation of the robot configuration but they constrained the robot to move on a limited set of trajectories. This is because the computation of their performance indices is extremely time-consuming. As a result, the optimization is still local, and it is not possible to ensure that the performance will be kept along another set of trajectories. Moreover, most of authors used criteria whose physical sense is not straightforward: For instance, it is difficult to say what is a good value for the strain energy or for the compliance while it seems simpler to understand what are good performance in terms of (static or dynamic) deformations, natural frequencies, input torques or actuator energy consumption. These indices are said to be “technology oriented” as we can relate their value to some technological constraints, as it will be detailed later in the paper.

In the present paper, we propose a methodology in order to perform a topology optimization for robots, valid globally in the workspace or for a set of given trajectories, and which is based on the use of technology-oriented performance criteria. In order to enforce the chosen performance indices to be globally valid, optimal robot configurations or trajectories for which extreme performance will be attained are computed, and iteratively updated.

In order to decrease the computational time associated with the computation of performance indices in numerous configurations, we exploit the structure of the elastic models (which have the highest computational costs) in order to reduce their computational complexity. Indeed, we show that it is possible to use configuration-independent model reduction techniques in order to considerably decrease the size of the stiffness and mass matrices of each link, expressed in their local frame. Then, these reduced matrices are used in order to build the configuration-dependent robot elastic models which are shown to be computationally efficient.

Finally, we use an optimization algorithm called the Linearization Method (LM) [52] which has proven, for our class of problems, to give results in a computational time equivalent to standard topology optimization algorithms (e.g. Method of Moving Asymptotes (MMA) [53], Optimality Criteria (OC) (for instance, see [54]), Projected Gradient [55], Convex Linearization method (CONLIN) [56]) but its implementation is less complex and makes it quite easy to perform modification or improvement.

The paper is written as follows. First, in Section 2, we introduce the LM algorithm. Then, in Section 3, we discuss the several technology-oriented performance indices that could be used for our optimization problem and explain our choice. We also provide a computationally efficient method for their calculation. Section 4 introduces the procedure to choose the optimal configurations and trajectories that will be used in order to compute the robot performance. In Section 5, the topology optimization of a five-bar mechanism is performed. Finally, in Section 6, our conclusions are drawn.

## 2. Topology optimization formulation and resolution

### 2.1. Problem formulation

The topology optimization problems have the standard form of a nonlinear program:

$$\min_{\substack{\boldsymbol{\rho} \in [0,1]^n \\ g(\boldsymbol{\rho}) \leq 0 \\ h(\boldsymbol{\rho}) = 0}} f(\boldsymbol{\rho}) \quad (1)$$

Each component of the vector of decision variables  $\boldsymbol{\rho} \in [0, 1]^n \subseteq R^n$  represents the density of a given element, defined in Section 3. The objective function and the constraints usually model some performance indices

(described in the next sections) or constraints on the structure (e.g., some constant mass for easing the solving process).

The main characteristics of these optimization problems is the high number of variables, usually several thousands, and the availability of the symbolic expression of the gradient. Therefore, the aim of the solving method is basically to use as well as possible the linear models of the constraints to prevent too many function and gradient evaluations.

The most used method for solving this class of problems is MMA [53]. It turns out to be more efficient than other methods, and is available under the form of a Matlab code, which is the only one that handles multiple inequality and equality constraints. However, it presents several drawbacks that lead us to consider alternative methods. First, MMA shows some unexplained surprising behaviors. In particular, it fails on badly conditioned problems, e.g., minimizing  $\rho_1 + \rho_2$  subject to  $\rho_1^2 + 1000\rho_2^2 \leq 1$ <sup>1</sup>. Second, MMA and its corresponding code are very complex, making very difficult any modification or improvement. For these reasons, we considered using a more standard optimization method, namely the Linearization Method.

## 2.2. The Linearization Method

The projected gradient method is the most used method for large scale convex optimization. It consists in choosing the search direction by projecting the cost function opposite gradient on the constraints. This projection is well defined only for convex constraints. A straightforward extension to non-convex problems consists in projecting the cost function opposite gradient on the constraint linearization instead of the constraint, and is called the Linearization Method. Up to our knowledge, this method seems to be first introduced in [57] (see [58, 59] for more comprehensive studies, including convergence properties of the method), but has not been applied to topology optimization so far. The basic idea of the method is a standard iteration  $\boldsymbol{\rho}_{k+1} = \boldsymbol{\rho}_k + \alpha_k \mathbf{d}_k$ , where  $\alpha_k$  is a step size and  $\mathbf{d}_k$  is chosen as illustrated on Figure 2. The feasible set of the constraint  $g(\boldsymbol{\rho}) = \rho_1^2 + \rho_2^2 - 1 \leq 0$  is the disk, and the linearization of the constraint at  $\boldsymbol{\rho}_k$  is the half-plane  $g(\boldsymbol{\rho}_k) + \nabla g(\boldsymbol{\rho}_k)(\boldsymbol{\rho} - \boldsymbol{\rho}_k) \leq 0$ . The vector  $\mathbf{u}_k = \boldsymbol{\rho}_k - s_k \nabla f(\boldsymbol{\rho}_k)$  is projected on the linearization of the constraint to obtain  $\mathbf{v}_k$ . Then,  $\mathbf{d}_k$  is the vector  $\mathbf{v}_k - \boldsymbol{\rho}_k$ , depicted as a thick arrow in Figure 2. We see that the projection process modifies the cost function gradient (the thin arrow) so that the new direction (the thick arrow) accounts the constraint. The scalar  $s_k$  is used as a second step size for the method. The right hand side diagram of Figure 2 shows the vectors  $\mathbf{d}_k$  obtained for different vectors  $\boldsymbol{\rho}_k$ , and clearly illustrates that the method converges to the minimizer of the problem for sufficiently small step sizes. Formally, the Linearization Method is defined as follows:

$$\mathbf{v}_k = \text{proj}_{L_k}(\boldsymbol{\rho}_k - s_k \nabla f(\boldsymbol{\rho}_k)) \quad (2)$$

$$\boldsymbol{\rho}_{k+1} = \boldsymbol{\rho}_k + \alpha_k (\mathbf{v}_k - \boldsymbol{\rho}_k). \quad (3)$$

with

$$L_k = \{\boldsymbol{\rho} \in R^n : g(\boldsymbol{\rho}_k) + \nabla g(\boldsymbol{\rho}_k)^T(\boldsymbol{\rho} - \boldsymbol{\rho}_k) \leq 0, h(\boldsymbol{\rho}_k) + \nabla h(\boldsymbol{\rho}_k)^T(\boldsymbol{\rho} - \boldsymbol{\rho}_k) = 0\}, \quad (4)$$

where the inequality stands component-wise. The projection operation  $\text{proj}_{L_k}(\mathbf{u})$  projects the vector  $\mathbf{u}$  onto the set  $L_k$ , which is here defined as linear inequality and equality constraints. It is therefore convex, and the projection is well defined.

*Implementation details.* The projection onto the convex set  $L_k$ , although well defined, is difficult in practice due to the high number of variables. We used an approximation computed by successive projections [60]. The high number of variables although prevents any line search, which requires too many function evaluations. Therefore, we implemented a constant step size strategy  $\alpha_k = 1$  and  $s_k = s_0$  [61]. However, several studies have shown that enforcing a maximal norm for the step increases the overall efficiency. Therefore, we implemented the following heuristic step size rule:

$$\alpha_k = \begin{cases} 1 & \text{if } \|\mathbf{v}_k - \boldsymbol{\rho}_k\|_\infty \leq \bar{\alpha} \\ \frac{\bar{\alpha}}{\|\mathbf{v}_k - \boldsymbol{\rho}_k\|_\infty} & \text{otherwise.} \end{cases} \quad (5)$$

<sup>1</sup>Note that providing the diagonal of the Hessian to MMA allows solving this problem, but this second order information is not available in the problems solved in the present paper.

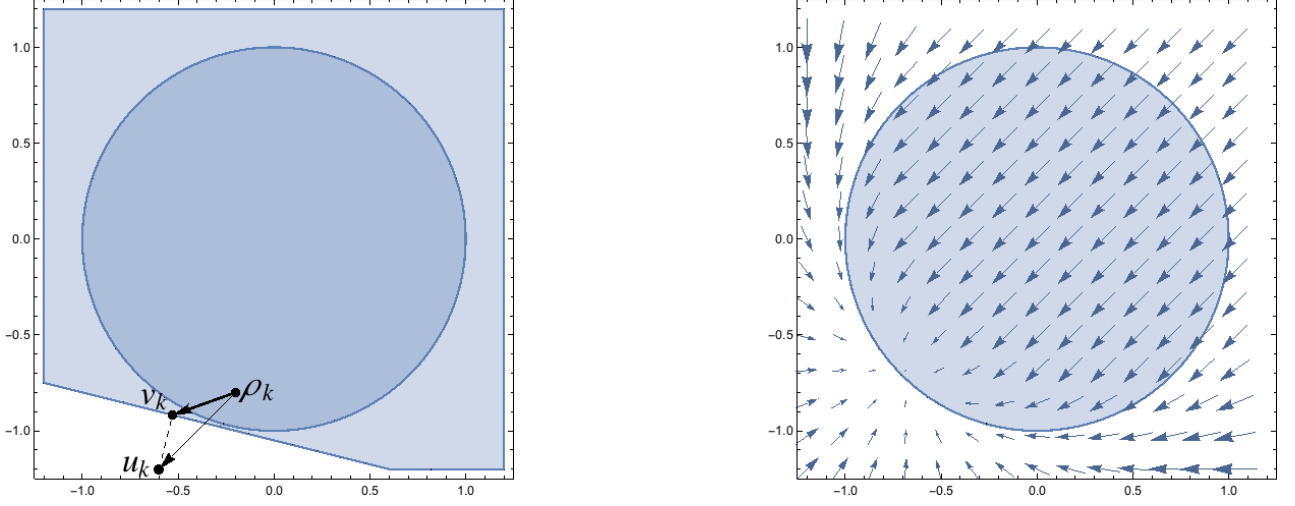


Figure 2: Left: Projected gradient. Right: Linearization method (vectors are scaled for the sake of lisibility).

147 The constants  $s_0$  and  $\bar{\alpha}$  need to be fixed for each problem with the usual drawbacks of constant step size  
 148 methods. Too large initial steps cause divergence, while too small initial steps slow down the convergence.  
 149 However, this behavior is easily identified on the first iterations of the algorithm, and a correct step size is easy  
 150 to find. Another heuristic dedicated to topology optimization has been implemented. Preliminary experiments  
 151 showed that the Linearization Method did not remove enough material during the first iterations. We fixed  
 152 this issue by considering a descent direction with a larger infinite norm, having the effect of removing more  
 153 material. Equation (2) is replaced by  $\mathbf{v}_k = \text{proj}_{L_k}(\boldsymbol{\rho}_k - s_k \mathbf{d}_k)$  with

$$154 \quad d_{ki} = \begin{cases} \|\nabla f(\boldsymbol{\rho}_k)\|_\infty & \text{if } \nabla f(\boldsymbol{\rho}_k)_i > 0 \\ -\|\nabla f(\boldsymbol{\rho}_k)\|_\infty & \text{if } \nabla f(\boldsymbol{\rho}_k)_i < 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

155 This is indeed a descent direction since  $\nabla f(\boldsymbol{\rho}_k)^T(-\mathbf{d}_k) < 0$ . Finally, a simplified heavy-ball process [59] is  
 156 added to the method to speedup its convergence. An implementation of this Linearization Method is available  
 157 on the website of the second author.

### 158 3. Performance indices

#### 159 3.1. Selection of the performance indices

160 In our work, we focus on the selection on technology-oriented performance indices whose computational  
 161 cost can be acceptable for an optimization process.

162 The selection of the performance indices for the optimization of a mechanical system is not a simple task  
 163 and is highly dependent on the sensitivity of the man-or-the-art. For instance, as mentioned in the introduction,  
 164 past works [50, 51] used as criteria the strain energy or the compliance of the robot, but the physical sense of  
 165 these indices is not straightforward. Indeed, engineers would wonder what is an acceptable value for the strain  
 166 energy or for the compliance of the robot. Other works [48, 49] considered performance such as overshoot,  
 167 controller settling time, number of oscillations, final deviation or also actuator energy consumption. All these  
 168 indices are closer from technological constraints, however the authors constrain the robot to move on a limited  
 169 set of trajectories in order to decrease the computational cost of the problem. Therefore, the optimization of  
 170 the robot stays local.

171 In order to solve these issues, we prefer to use for the optimization process the following indices (the list  
 172 is not necessary exhaustive), which can be strongly related to the technological constraints:

- 173 • *robot inertial parameters*: the authors of [50, 51] used the robot mass as a criterion performance, which  
 174 is one of the most obvious performance index, as the mass plays an important role on the value of the  
 175 input efforts, but also can be linked to the robot cost (lower mass leads to the use of less material for the

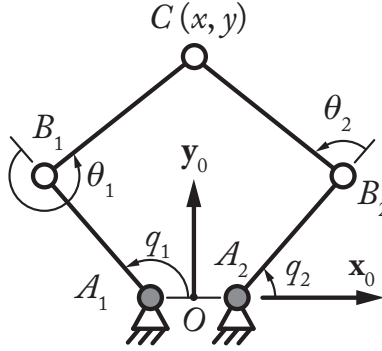


Figure 3: A five-bar mechanism (the gray pairs denote the actuated joints).

176 robot design). However, the robot mass is not necessarily the only inertial parameter to be taken into  
 177 account. For instance, as shown in [62], the most influential term in the dynamic model of the five-bar  
 178 mechanism (Fig. 3) is a grouped inertial parameter equal to  $zz_R = zz_{i1} + \ell_{i1}^2 m_{i2}$ , where  $zz_{i1}$  is the moment  
 179 of inertia around  $z_0$  of the link between the points  $A_i$  and  $B_i$  computed at  $A_i$ ,  $\ell_{A_i B_i}$  is the distance between  
 180 points  $A_i$  and  $B_i$  and  $m_{i2}$  is the mass of the link between the points  $B_i$  and  $C$ . Therefore, minimizing  $zz_R$   
 181 is more likely to minimize the robot input torques than minimizing the robot mass.

- 182 • *robot input efforts or energy consumption*: the cost of an actuator is related to its power or to its continu-  
 183 ous or peak torques. We know that, for a usual robot, more than 90 % of the dynamic effects come from  
 184 the “rigid behavior” of the mechanical architecture (remaining 10 % being due to elasticity, impacts  
 185 due to clearance in joints, and unmodelled friction effects) [63]. Therefore, criteria based on the input  
 186 efforts, energy or power consumption can be computed with a very good accuracy neglecting the elastic  
 187 behavior of the links.
- 188 • *robot static deformations*: static deformations under external (potentially variable) loading are classical  
 189 performance indices when designing a mechanical system.
- 190 • *robot natural frequencies*: the first natural frequencies are associated with the highest level of energy  
 191 due to vibrations, and the first modes represent the displacements with the highest amplitude. Moreover,  
 192 the first natural frequencies are also used in order to set the bandwidth of controllers: it is usually said  
 193 that the cutoff frequency for a robot controller must be set around the half of the first natural frequency of  
 194 the mechanical architecture [64]. If not, the controller can become unstable due to unmodeled dynamics.  
 195 We believe that natural frequencies are more important indices than the amplitude of the robot dynamic  
 196 deformations used [48, 49, 50, 51]: The lack of accuracy due to oscillation can be compensated thanks  
 197 to advanced controller (e.g. [65, 66]) while low robot natural frequencies cannot be increased through  
 198 control.

199 The computation of the robot inertial parameters and input efforts presents no difficulty, and this is the  
 200 reason why they are given in Appendix A and Appendix B. We show in these appendices that the expression  
 201 of all rigid inertial parameters and of the rigid dynamic model is linear with respect to the decision variables.

202 Below, we prefer to stress the issue of the calculation of the robot deformations and natural frequencies,  
 203 whose cost of computation is huge, and which must be computed for numerous robot configurations. As  
 204 mentioned in introduction, in order to considerably decrease the computation cost of these performance in-  
 205 dices, we exploit the structure of the elastic models (which have the highest computational costs) in order  
 206 to reduce their computational complexity. Indeed, we show below that it is possible to use configuration-  
 207 independent model reduction techniques in order to considerably decrease the size of the stiffness and mass  
 208 matrices of each link, expressed in their local frame. Then, these reduced matrices are used in order to build  
 209 the configuration-dependent robot elastic models which are shown to be computationally efficient.

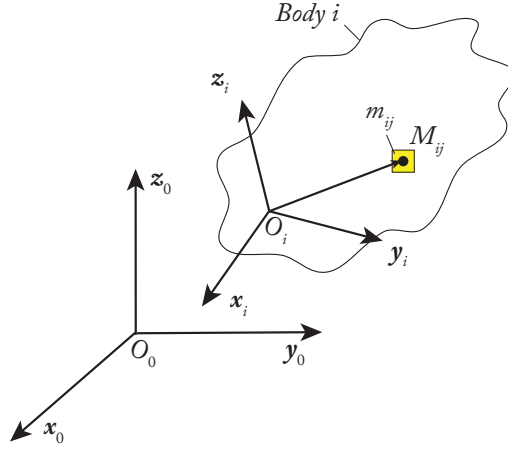


Figure 4: A body in space meshed with finite elements.

### 210 3.2. Static deformations

211 In what follows, we consider a robot made of  $n$  bodies. The body  $i$  is meshed with  $m$  elements. The  
 212 element  $j$  of the body  $i$  is denoted as the element  $ij$  (Fig. 4).

#### 213 3.2.1. Material interpolation scheme and stiffness matrix of a link

214 Topology optimization uses the same physical model as in the FEM for modelling of a link, except that we  
 215 use an interpolation scheme in order to define an artificial material. This method is called the Solid Isotropic  
 216 Material with Penalization (SIMP, [67]) and is known to be the most effective and the most widely used  
 217 material interpolation scheme. This material interpolation scheme is adopted in order to avoid having a result  
 218 of optimization with too much intermediate material density, i.e. by denoting as  $\rho_{ij}$  the density associated with  
 219 the element  $ij$ , in order to have a black ( $\rho_{ij} = 1$ ) and white ( $\rho_{ij} = 0$ ) solution without too many grey elements  
 220 ( $0 < \rho_{ij} < 1$ ).

221 The SIMP scheme is defined as follows:

$$222 \quad E_{ij} = E_{\min} + \rho_{ij}^p (E_0 - E_{\min}), \quad \text{with } \rho_{ij} \in [0, 1] \quad (7)$$

223 where  $E_0$  is the Young's modulus of the material,  $E_{\min}$  is a very small stiffness value assigned to void regions  
 224 in order to prevent the stiffness matrix from becoming singular,  $p$  (typically  $p = 3$ ) is the penalization factor,  
 225 and  $E_{ij}$  is the Young's modulus of element  $j$  of the body  $i$  corresponding to the density variable  $\rho_{ij}$ .

226 Then the stiffness matrix associated with the element  $ij$  is thus given by:

$$227 \quad \mathbf{K}_{ij} = E_{ij} \mathbf{K}_{ij}^{(0)} = (E_{\min} + \rho_{ij}^p (E_0 - E_{\min})) \mathbf{K}_{ij}^{(0)} \quad (8)$$

228 where  $\mathbf{K}_{ij}^{(0)}$  is the stiffness matrix of a single element computed for a Young's modulus equal to 1, and the  
 229 potential elastic energy of the element is:

$$230 \quad U_{e_{ij}} = \frac{1}{2} \mathbf{u}_{ij}^T \mathbf{K}_{ij} \mathbf{u}_{ij} \quad (9)$$

231 in which  $\mathbf{u}_{ij}$  is the vector of the element  $ij$  nodal displacements. The total elastic energy of the body  $i$  is thus  
 232 equal to

$$233 \quad U_{e_i} = \sum_{j=1}^m U_{e_{ij}} = \frac{1}{2} \sum_{j=1}^m \mathbf{u}_{ij}^T \mathbf{K}_{ij} \mathbf{u}_{ij} = \frac{1}{2} \mathbf{u}_{i\text{tot}}^T \mathbf{K}_{i\text{tot}} \mathbf{u}_{i\text{tot}} \quad (10)$$

234 where

- 235 •  $\mathbf{u}_{i\text{tot}} = [\mathbf{u}_{i1}^T \dots \mathbf{u}_{im}^T]^T$  is the vector stacking all nodal displacements for all  $m$  elements of the body  $i$



236 •  $\mathbf{K}_{i\text{tot}}$  is a block-diagonal matrix stacking on its diagonal all elementary stiffness matrices as follows:

$$237 \quad \mathbf{K}_{i\text{tot}} = \begin{bmatrix} \mathbf{K}_{i1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_{im} \end{bmatrix} \quad (11)$$

238 Finally, the link stiffness matrix  $\mathbf{K}_i$  of the body  $i$  can be obtained by taking into account the fact that the  
239 nodal displacements of the element  $ij$  are equal to the nodal displacements of its adjacent elements. As a  
240 result, the expression of the vector  $\mathbf{u}_{i\text{tot}}$  can be obtained from a reduced set of independent coordinates  $\mathbf{u}_i$  [68]  
241 as follows:

$$242 \quad \mathbf{u}_{i\text{tot}} = \mathbf{J}_i \mathbf{u}_i \quad (12)$$

243 where, for a single body,  $\mathbf{J}_i$  is constant. Introducing (12) into (10), we get

$$244 \quad U_{e_i} = \frac{1}{2} \mathbf{u}_i^T \mathbf{K}_i \mathbf{u}_i \quad (13)$$

245 where  $\mathbf{K}_i = \mathbf{J}_i^T \mathbf{K}_{i\text{tot}} \mathbf{J}_i$  is the body stiffness matrix which relates the nodal displacements  $\mathbf{u}_i$  to the forces  $\mathbf{f}_i$   
246 exerted on the nodes by the relation [34]:

$$247 \quad \mathbf{f}_i = \frac{\partial U_{e_i}}{\partial \mathbf{u}_i} = \mathbf{K}_i \mathbf{u}_i \quad (14)$$

248 In the following the paper, we consider that the components of the vector  $\mathbf{u}_i$  are ordered and split into two  
249 parts such as

$$250 \quad \mathbf{u}_i = \begin{bmatrix} \mathbf{u}_{i f} \\ \mathbf{u}_{i l} \end{bmatrix} \quad (15)$$

251 where  $\mathbf{u}_{i f}$  is the vector of displacements of the nodes on which no force or displacements are imposed, while  
252  $\mathbf{u}_{i l}$  is the vector of displacements for the nodes which are constrained with forces and/or displacements. The  
253 nodes whose displacements are represented by the vector  $\mathbf{u}_{i l}$  are named the interface nodes.

### 254 3.2.2. Efficient computation of the elastostatic model

255 The usual computation of the robot elastostatic model is shown in Appendix C. If this approach is used  
256 for the computation of the robot stiffness matrix  $\mathbf{K}$ , its final dimension is huge (typically, in our problems, the  
257 dimension of  $\mathbf{K}$  is greater than  $(10^5 \times 10^5)$ ), thus leading to a huge computational cost for solving the defor-  
258 mation problem given at equation (C.5), problem which must be solved for any tested robot configurations.

259 It is possible to considerably reduce the computational cost for solving the elastostatic model by using a  
260 model reduction technique for the robot links as follows.

261 In order to perform the reduction of the model size, we use the partitioning of the vector  $\mathbf{u}_i$  shown in (15).  
262 We denote as  $a$  the size of the vector  $\mathbf{u}_{i f}$  and as  $b$  the size of the vector  $\mathbf{u}_{i l}$ . From the definitions of  $\mathbf{u}_{i f}$  and  
263  $\mathbf{u}_{i l}$ , we also partition the vector  $\mathbf{f}_i$  in (14) such that

$$264 \quad \mathbf{f}_i = \begin{bmatrix} \mathbf{0}_{(a \times 1)} \\ \mathbf{f}_{i l} \end{bmatrix} \quad (16)$$

265 in which  $\mathbf{0}_{(a \times 1)}$  is a zero vector of size  $a$  which represents the absence of interaction between the environ-  
266 ment and the nodes whose displacements are parameterized by  $\mathbf{u}_{i f}$ . Taking into account the separation of  
267 components in the vectors  $\mathbf{u}_i$  and  $\mathbf{f}_i$ , the equation (14) can be rewritten as:

$$268 \quad \begin{bmatrix} \mathbf{0}_{(a \times 1)} \\ \mathbf{f}_{i l} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_i^{(11)} & \mathbf{K}_i^{(12)} \\ \mathbf{K}_i^{(21)} & \mathbf{K}_i^{(22)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{i f} \\ \mathbf{u}_{i l} \end{bmatrix} \quad (17)$$

269 where  $\mathbf{K}_i^{(11)}$  is a  $(a \times a)$  matrix,  $\mathbf{K}_i^{(12)} = \mathbf{K}_i^{(21)T}$  is a  $(a \times b)$  matrix and  $\mathbf{K}_i^{(22)}$  is a  $(b \times b)$  matrix.

270 (17) can be split into the two following equations:

$$271 \quad \mathbf{0}_{(a \times 1)} = \mathbf{K}_i^{(11)} \mathbf{u}_{if} + \mathbf{K}_i^{(12)} \mathbf{u}_{il} \quad (18)$$

$$272 \quad \mathbf{f}_{il} = \mathbf{K}_i^{(21)} \mathbf{u}_{if} + \mathbf{K}_i^{(22)} \mathbf{u}_{il} \quad (19)$$

273 By using (18), it is possible to express  $\mathbf{u}_{if}$  as a function of the displacements of the interface nodes  $\mathbf{u}_{il}$  as:

$$274 \quad \mathbf{u}_{if} = \mathbf{\Phi}_{si} \mathbf{u}_{il}, \text{ in which } \mathbf{\Phi}_{si} = -(\mathbf{K}_i^{(11)})^{-1} \mathbf{K}_i^{(12)} \quad (20)$$

275  $\mathbf{\Phi}_{si}$  is the matrix of the static modes [69].

276 Introducing (20) into (19), we get

$$277 \quad \mathbf{f}_{il} = \mathbf{K}_i^{red} \mathbf{u}_{il}, \text{ in which } \mathbf{K}_i^{red} = \mathbf{K}_i^{(21)} \mathbf{\Phi}_{si} + \mathbf{K}_i^{(22)} \quad (21)$$

278 while the potential elastic energy (13) can be rewritten as

$$279 \quad U_{e_i} = \frac{1}{2} \mathbf{u}_{il}^T \mathbf{K}_i^{red} \mathbf{u}_{il} \quad (22)$$

280  $\mathbf{K}_i^{red}$  is the reduced stiffness matrix associated with the displacements of the interface nodes of the body  $i$ .  
 281 This matrix is usually of small dimension (typically, in our 2D problems, of dimension  $(6 \times 6)$ , while in 3D  
 282 it is of dimension  $(12 \times 12)$ ) and it does not depend on the robot configuration, which is a great advantage  
 283 because it must be computed only once at each step of the optimization algorithm, whatever the number of  
 284 tested configurations for the robot. Indeed, most of the computational cost of its computation is due to the  
 285 computation of the matrix  $\mathbf{\Phi}_{si}$  in (20). However, for two bodies  $i$  and  $k$ , the computation of the matrices  $\mathbf{\Phi}_{si}$   
 286 and  $\mathbf{\Phi}_{sk}$  is independent. Thus the computation can be made in parallel on multi-core computers in order to  
 287 save computational time.

288  
 289 Considering now the robot made of  $n$  bodies, the full potential elastic of the system is given by:

$$290 \quad U_e = \sum_{i=1}^n U_{e_i} = \frac{1}{2} \sum_{i=1}^n \mathbf{u}_{il}^T \mathbf{K}_i^{red} \mathbf{u}_{il} = \frac{1}{2} \mathbf{u}_{tot}^{red T} \mathbf{K}_{tot}^{red} \mathbf{u}_{tot}^{red} \quad (23)$$

291 where

- 292 •  $\mathbf{u}_{tot}^{red} = [\mathbf{u}_{1l}^T \dots \mathbf{u}_{nl}^T]^T$  is the vector stacking all interface node displacements for all  $n$  robot bodies
- 293 •  $\mathbf{K}_l$  is a block-diagonal matrix stacking on its diagonal all bodies stiffness matrices as follows:

$$294 \quad \mathbf{K}_{tot}^{red} = \begin{bmatrix} \mathbf{K}_1^{red} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_n^{red} \end{bmatrix} \quad (24)$$

295 The reduced robot stiffness matrix  $\mathbf{K}_r$  can be obtained by taking into account the fact the robot bodies are  
 296 connected altogether through the interface nodes. As a result, the expression of the vector  $\mathbf{u}_{tot}^{red}$  can be obtained  
 297 from a reduced set of independent coordinates  $\mathbf{u}_r$  [34, 68] as follows:

$$298 \quad \mathbf{u}_{tot}^{red} = \mathbf{J}_r(\mathbf{q}) \mathbf{u}_r \quad (25)$$

299 where  $\mathbf{J}_r(\mathbf{q})$  depends on the robot configuration  $\mathbf{q}$  but not on the nodal displacements  $\mathbf{u}_r$  (assumption of small  
 300 perturbations).

301 Introducing (25) into (23), we get

$$302 \quad U_e = \frac{1}{2} \mathbf{u}_r^T \mathbf{K}_r \mathbf{u}_r \quad (26)$$

303 where  $\mathbf{K}_r = \mathbf{J}_r^T(\mathbf{q})\mathbf{K}_{tot}^{red}\mathbf{J}_r(\mathbf{q})$  is the robot stiffness matrix which relates the nodal displacements  $\mathbf{u}_r$  to external  
 304 forces  $\mathbf{f}_r$  exerted on the considered nodes by the relation:

$$305 \quad \mathbf{f}_r = \frac{\partial U_e}{\partial \mathbf{u}_r} = \mathbf{K}_r \mathbf{u}_r \quad (27)$$

306 For a 2D problem, the typical size of the matrix  $\mathbf{K}_r$  is lower than  $(6n \times 6n)$ ,  $n$  being the number of robot bodies  
 307 ( $(12n \times 12n)$  for 3D problems). Therefore, even if the problems (C.5) and (27) will give exactly the same  
 308 results of computation, the resolution of the equation (27) is much more efficient than the resolution of the  
 309 equation (C.5) due to the considerably reduced size of the problem.

### 310 3.2.3. Stiffness performance index

311 Usually, it is not necessary to constrain all deformations  $\mathbf{u}_r$ , but a set of them. Let us define a row vector  
 312  $\mathbf{e}$  able to extract  $c$  components  $\mathbf{u}_c$  of  $\mathbf{u}_r$  such as  $\mathbf{u}_c = \mathbf{e} \mathbf{u}_r$ . It is then quite usual to constrain the two-norm of  
 313 this vector  $\mathbf{u}_c$  or its square (in order to avoid the use of square roots in the definition of the performance index)  
 314 given by:

$$315 \quad c_1 = \mathbf{u}_c^T \mathbf{u}_c = \mathbf{u}_r^T \mathbf{e}^T \mathbf{e} \mathbf{u}_r \quad (28)$$

316 The derivative of this criterion with respect to the decision variable  $\rho_{ij}$  is given in Appendix E.

## 317 3.3. Natural frequencies

### 318 3.3.1. Mass matrix of a link

319 The mass matrix associated with the element  $ij$  is given by:

$$320 \quad \mathbf{M}_{ij} = \rho_{ij} \mathbf{M}_{ij}^{(0)} \quad (29)$$

321 where  $\mathbf{M}_{ij}^{(0)}$  is the mass matrix of a single element computed for a density equal to 1. Accordingly, the kinetic  
 322 energy of the element due to elastic oscillations is:

$$323 \quad T_{e_{ij}} = \frac{1}{2} \dot{\mathbf{u}}_{ij}^T \mathbf{M}_{ij} \dot{\mathbf{u}}_{ij} \quad (30)$$

324 in which  $\dot{\mathbf{u}}_{ij}$  is the vector of the element  $ij$  nodal velocities. The total kinetic energy of the body  $i$  is thus equal  
 325 to

$$326 \quad T_{e_i} = \sum_{j=1}^m T_{e_{ij}} = \frac{1}{2} \sum_{j=1}^m \dot{\mathbf{u}}_{ij}^T \mathbf{M}_{ij} \dot{\mathbf{u}}_{ij} = \frac{1}{2} \dot{\mathbf{u}}_{i\,tot}^T \mathbf{M}_{i\,tot} \dot{\mathbf{u}}_{i\,tot} \quad (31)$$

327 where  $\mathbf{M}_{i\,tot}$  is a block-diagonal matrix stacking on its diagonal all elementary mass matrices as follows:

$$328 \quad \mathbf{M}_{i\,tot} = \begin{bmatrix} \mathbf{M}_{i1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{im} \end{bmatrix} \quad (32)$$

329 Finally, differentiating (12) with respect to time and recalling that the matrix  $\mathbf{J}_i$  is constant, we obtain:

$$330 \quad \dot{\mathbf{u}}_{i\,tot} = \mathbf{J}_i \dot{\mathbf{u}}_i \quad (33)$$

331 Introducing (33) into (31), we get

$$332 \quad T_{e_i} = \frac{1}{2} \dot{\mathbf{u}}_i^T \mathbf{M}_i \dot{\mathbf{u}}_i \quad (34)$$

333 where  $\mathbf{M}_i = \mathbf{J}_i^T \mathbf{M}_{i\,tot} \mathbf{J}_i$  is the body mass matrix.

### 3.3.2. Efficient computation of the elastodynamic model

The usual computation of the robot elastodynamic model is shown in Appendix D. If this approach is used for the computation of the natural frequencies, as previously met for the computation of the elastostatic model, the cost of their evaluation will be prohibitive due to the huge dimension of the matrices to evaluate for any tested robot configurations.

It is possible to considerably reduce the computational cost for computing the robot natural frequencies by using a Craig-Bampton model reduction technique [69] applied to each robot body as follows.

The Craig-Bampton model reduction technique for a body is based on the assumption that the body nodal coordinates  $\mathbf{u}_{if}$  (recall that  $\mathbf{u}_{if}$  is the vector of displacements of the nodes on which no force or displacements are imposed) can be expressed as a function of the nodal coordinates  $\mathbf{u}_{il}$  corresponding to the interface nodes plus a term characterizing their vibratory free behavior, as follows:

$$\mathbf{u}_{if} = \Phi_{si}\mathbf{u}_{il} + \Phi_{di}\mathbf{q}_i \quad (35)$$

where the term  $\Phi_{si}\mathbf{u}_{il}$  comes from the equation (20) and characterizes the node static displacements and the term  $\Phi_{di}\mathbf{q}_i$  is an additional term characterizing the body oscillatory behavior. Classically, the matrix  $\Phi_{di}$  has the following form

$$\Phi_{di} = \begin{bmatrix} \mathbf{u}_{if}^{(1)} & \dots & \mathbf{u}_{if}^{(s)} \end{bmatrix} \quad (36)$$

where the vector  $\mathbf{u}_{if}^{(k)}$  is the  $k$ th eigenmode associated with the equation:

$$\mathbf{M}_i^{(11)}\ddot{\mathbf{u}}_{if} + \mathbf{K}_i^{(11)}\mathbf{u}_{if} = \mathbf{0} \quad (37)$$

in which the matrix  $\mathbf{K}_i^{(11)}$  is defined in (17) and the matrix  $\mathbf{M}_i^{(11)}$  is the part of the matrix  $\mathbf{M}_i$  corresponding to the variables  $\ddot{\mathbf{u}}_{if}$ . Thus, the dimension of the matrix  $\mathbf{M}_i^{(11)}$  is  $(a \times a)$ . In  $\Phi_{di}$ ,  $s$  vectors are chosen among the  $a$  eigenmodes associated with (37). Usually,  $s \ll a$  (typically, in our examples,  $s$  is lower than 10).

Now, using (35), the nodal displacement vector  $\mathbf{u}_i$  is thus given by

$$\mathbf{u}_i = \begin{bmatrix} \mathbf{u}_{if} \\ \mathbf{u}_{il} \end{bmatrix} = \mathbf{B}_i \mathbf{u}_{ir}, \text{ where } \mathbf{B}_i = \begin{bmatrix} \Phi_{si} & \Phi_{di} \\ \mathbf{1}_b & \mathbf{0}_{(b \times s)} \end{bmatrix} \text{ and } \mathbf{u}_{ir} = \begin{bmatrix} \mathbf{u}_{il} \\ \mathbf{q}_i \end{bmatrix} \quad (38)$$

in which  $\mathbf{1}_b$  is the identity matrix of dimension  $b$ . Taking into account once again that the matrix  $\mathbf{B}$  does not depend on the robot configuration or of the link deformation (hypothesis of small perturbations), the derivative of (38) with respect to time leads to

$$\dot{\mathbf{u}}_i = \mathbf{B}_i \dot{\mathbf{u}}_{ir} \quad (39)$$

By using these transformations of coordinates, the body potential and kinetic elastic energies given at (13) and (31) becomes

$$U_{e_i} = \frac{1}{2} \mathbf{u}_{ir}^T \mathbf{K}_{ir} \mathbf{u}_{ir} \quad (40)$$

$$T_{e_i} = \frac{1}{2} \dot{\mathbf{u}}_{ir}^T \mathbf{M}_{ir} \dot{\mathbf{u}}_{ir} \quad (41)$$

where  $\mathbf{K}_{ir} = \mathbf{B}_i^T \mathbf{K}_{i\text{tot}} \mathbf{B}_i$  and  $\mathbf{M}_{ir} = \mathbf{B}_i^T \mathbf{M}_{i\text{tot}} \mathbf{B}_i$  are the reduced body stiffness and mass matrices. These matrices are usually of small dimension (typically, in our 2D problems, of dimension  $(12 \times 12)$ , while in 3D it is of dimension  $(18 \times 18)$ ) and they not depend on the robot configuration, which is a great advantage because they must be computed only once at each step of the optimization algorithm, whatever the number of tested configurations for the robot. Indeed, similarly as for the elastostatic model, most of the computational cost is due to the computation of the matrix  $\Phi_{di}$  in (36). However, for two bodies  $i$  and  $k$ , the computation of the matrices  $\Phi_{di}$  and  $\Phi_{dk}$  is independent. Thus the computation can be made in parallel on multi-core computers in order to save computational time.

373

374 Considering now the robot made of  $n$  bodies, the full potential and kinetic elastic energies of the system  
 375 are given by:

$$376 \quad U_e = \sum_{i=1}^n U_{e_i} = \frac{1}{2} \sum_{i=1}^n \mathbf{u}_{i_r}^T \mathbf{K}_i^{red} \mathbf{u}_{i_r} = \frac{1}{2} \mathbf{u}_{tot}^{dyn T} \mathbf{K}_{tot}^{dyn} \mathbf{u}_{tot}^{dyn} \quad (42)$$

$$377 \quad T_e = \sum_{i=1}^n T_{e_i} = \frac{1}{2} \sum_{i=1}^n \dot{\mathbf{u}}_{i_r}^T \mathbf{M}_i^{red} \dot{\mathbf{u}}_{i_r} = \frac{1}{2} \dot{\mathbf{u}}_{tot}^{dyn T} \mathbf{M}_{tot}^{dyn} \dot{\mathbf{u}}_{tot}^{dyn} \quad (43)$$

378 where

- 379 •  $\mathbf{u}_{tot}^{dyn} = [\mathbf{u}_{1_r}^T \dots \mathbf{u}_{n_r}^T]^T$  is the vector stacking all vectors  $\mathbf{u}_{i_r}$  for all  $n$  robot bodies
- 380 •  $\mathbf{K}_{tot}^{dyn}$  and  $\mathbf{M}_{tot}^{dyn}$  are block-diagonal matrices stacking on their diagonal all bodies stiffness and mass  
 381 matrices as follows:

$$382 \quad \mathbf{K}_{tot}^{dyn} = \begin{bmatrix} \mathbf{K}_{1_r} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_{n_r} \end{bmatrix}, \quad \mathbf{M}_{tot}^{dyn} = \begin{bmatrix} \mathbf{M}_{1_r} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{n_r} \end{bmatrix} \quad (44)$$

383 The reduced robot elastodynamic model can be obtained by taking into account the fact the robot bodies  
 384 are connected altogether through the interface nodes. As a result, the expression of the vector  $\mathbf{u}_{tot}^{dyn}$  can be  
 385 obtained from a reduced set of independent coordinates  $\mathbf{u}_d$  [34, 68] as follows:

$$386 \quad \mathbf{u}_{tot}^{dyn} = \mathbf{J}_d(\mathbf{q}) \mathbf{u}_d \quad (45)$$

387 where  $\mathbf{J}_d(\mathbf{q})$  depends on the robot configuration  $\mathbf{q}$  but not on the nodal displacements  $\mathbf{u}_d$  (assumption of small  
 388 perturbations). Moreover, for the analysis of the oscillatory free behavior, the configuration  $\mathbf{q}$  is considered  
 389 constant, thus

$$390 \quad \dot{\mathbf{u}}_{tot}^{dyn} = \mathbf{J}_d(\mathbf{q}) \dot{\mathbf{u}}_d \quad (46)$$

391 Introducing (45) and (46) into (42) and (43), we get

$$392 \quad U_e = \frac{1}{2} \mathbf{u}_d^T \mathbf{K}_d \mathbf{u}_d \quad (47)$$

$$393 \quad T_e = \frac{1}{2} \dot{\mathbf{u}}_d^T \mathbf{M}_d \dot{\mathbf{u}}_d \quad (48)$$

394 where  $\mathbf{K}_d = \mathbf{J}_d^T(\mathbf{q}) \mathbf{K}_{tot}^{dyn} \mathbf{J}_d(\mathbf{q})$  and  $\mathbf{M}_d = \mathbf{J}_d^T(\mathbf{q}) \mathbf{M}_{tot}^{dyn} \mathbf{J}_d(\mathbf{q})$  are the reduced robot stiffness and mass matrices.

395 Using the Lagrange equations, which state that, in absence of external efforts, we have

$$396 \quad \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{u}}} \right) - \frac{\partial L}{\partial \mathbf{u}} = \mathbf{0} \quad (49)$$

397 where  $L = T_e - U_e$ , the reduced dynamic equation characterizing the robot free oscillations is

$$398 \quad \mathbf{M}_d \ddot{\mathbf{u}}_d + \mathbf{K}_d \mathbf{u}_d = \mathbf{0} \quad (50)$$

399 A solution  $\mathbf{u}_{dk}$  of this equation satisfies:

$$400 \quad (\omega_k^2 \mathbf{M}_d - \mathbf{K}_d) \mathbf{u}_{dk} = \mathbf{0} \quad (51)$$

401 where  $\omega_k = 2\pi f_k$ ,  $f_k$  is the natural frequency associated with the  $k$ th natural mode of vibrations and  $\mathbf{u}_{dk}$  is its  
 402 associated eigenvector.

403 For a 2D problem, the typical size of matrices  $\mathbf{K}_d$  and  $\mathbf{M}_d$  is lower than  $(12n \times 12n)$ ,  $n$  being the number of  
 404 robot bodies ( $(18n \times 18n)$  for 3D problems). Therefore, even if the problems (F.1) and (51) will give exactly  
 405 the same results of computation for the first frequencies, the resolution of the equation (51) is much more  
 406 efficient than the resolution of the equation (F.1) due to the considerably reduced size of the problem.

### 407 3.3.3. Vibration performance index

408 The first robot natural frequency is used as a performance index:

$$409 \quad c_2 = f_1 \quad (52)$$

410 Its derivative with respect to the decision variables is given in Appendix F.

## 411 4. Selection of the robot configurations and trajectories for the computation of the performance criteria

412 All indices presented in Section 3 are configuration dependent (except for the case of the inertial param-  
413 eters) and are thus local by nature. In the present section, we propose a methodology that can be used in  
414 order to enforce the chosen performance indices to be valid globally in the workspace or for a set of given  
415 trajectories. This methodology is based on the choice of optimal robot configurations or trajectories for which  
416 extreme performance will be attained. This procedure was inspired from [70, 71].

### 417 4.1. Selection of the robot trajectories for the computation of the input efforts

418 As presented in the introduction, the main drawback of analyzing the dynamic performances with the  
419 manipulator input efforts is that they depend on the trajectory. In order to avoid this problem, one idea could  
420 be to make the manipulator moves on selecting “exciting” trajectories that make it exhibiting the highest  
421 values of the input efforts. To generate these exciting trajectories, it is necessary to define an optimization  
422 procedure.

423 Several methods for exciting trajectory generation exist [72], the large majority of them being developed  
424 for dynamic parameter identification purpose. However, these methods have been developed in order to min-  
425 imize the uncertainty of estimation of the parameters to be identified, which is a different goal from what  
426 we target, i.e. finding trajectories on which the designed robot will be near to attain its maximal dynamic  
427 capacities.

428 Therefore, we propose to adopt the following strategy, which is based on the assumption that we have  
429 an *a priori* estimation of the near-optimal design for the robot link that will be denoted as  $\hat{\rho}$ . The choice of  
430 the variables  $\hat{\rho}$  is discussed in Section 4.3. From Appendix B, we know that the input efforts are linear with  
431 respect to the decision parameters  $\rho$  (see Eq. (B.4)) and that, for our estimated optimal design  $\hat{\rho}$ , they will be  
432 given by:

$$433 \quad \boldsymbol{\tau}(t) = \boldsymbol{\Gamma}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t)) \hat{\rho} \quad (53)$$

434 where  $t$  is the current time instant,  $\boldsymbol{\tau}$  is the vector of robot input efforts,  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$  are the robot active joint  
435 position, velocity and acceleration, respectively, and the matrix  $\boldsymbol{\Gamma}$  is the Jacobian matrix of  $\boldsymbol{\tau}$  with respect to  
436  $\hat{\rho}$ .

437 Let us now denote as  $\mathcal{T}$  a trajectory described by  $N$  samples of time, leading to  $N$  samples of the joint  
438 positions, velocities and accelerations  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ ,  $(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k)$  being the value of  $(\mathbf{q}(t_k), \dot{\mathbf{q}}(t_k), \ddot{\mathbf{q}}(t_k))$  at the time  
439 instant  $t_k$ . Considering all samples, we rewrite (53) as

$$440 \quad \mathbf{y} = \mathbf{W} \hat{\rho}, \text{ where } \mathbf{W} \text{ stacks all samples of } \boldsymbol{\Gamma} \text{ such as } \mathbf{W} = \begin{bmatrix} \boldsymbol{\Gamma}(\mathbf{q}_1, \dot{\mathbf{q}}_1, \ddot{\mathbf{q}}_1) \\ \vdots \\ \boldsymbol{\Gamma}(\mathbf{q}_N, \dot{\mathbf{q}}_N, \ddot{\mathbf{q}}_N) \end{bmatrix} \quad (54)$$

441 and  $\mathbf{y}$  stacks all samples of  $\boldsymbol{\tau}$  such as  $\mathbf{y} = [\boldsymbol{\tau}(t_1)^T \dots \boldsymbol{\tau}(t_N)^T]^T$ . Knowing *a priori* values for the actuator  
442 performance (such as the maximal velocity and motion ranges), the “exciting” trajectory will be found as the  
443 solution of the following optimization problem:

$$444 \quad \begin{aligned} \max_{\mathcal{T}} \quad & c_j(\hat{\rho}) \text{ (for } j = 4 \text{ or } j = 5) \\ \text{under} \quad & |\dot{q}_i(t_k)| \leq \dot{q}_i^{\max} \\ & \mathbf{q} \in [\mathbf{q}] \text{ and } \mathbf{x} \in [\mathbf{x}] \end{aligned} \quad (55)$$

445 where the functions  $c_4$  and  $c_5$  are defined in (B.10) and (B.11) and represent the squared values of the infinite  
446 norms of the input efforts and of their root-mean-square along the trajectory  $\mathcal{T}$ . Usually, we know that the  
447 infinite norm of the effort should be lower than the actuator peak effort specification, while the root-mean-  
448 square should be lower than the actuator continuous effort. Moreover,

449 •  $\dot{q}_i(t_k)$  is the value of the actuator  $i$  velocity at the sample time  $t_k$  while  $\dot{q}_i^{max}$  is the maximal admissible  
 450 velocity for the actuator  $i$ .

451 •  $[\mathbf{q}]$  represents the range of admissible joint configurations while  $[\mathbf{x}]$  is the range of admission end-  
 452 effector configurations.

453 In order to reduce the complexity of the design problem, we parameterize the trajectory  $\mathcal{T}$  as a succession  
 454 of portions of motions, each portion being parameterized by a typical motion profile (e.g. 5th order polynomial  
 455 laws, bang-bang profiles, etc). Therefore, the decision variables of the problem (55) are restricted to

- 456 • the initial and final robot configurations for each portion of the trajectory  $\mathcal{T}$
- 457 • plus the duration of the motion profile corresponding to each portion of the trajectory  $\mathcal{T}$ .

#### 458 4.2. Selection of the robot configurations for the computation of elastic performance

459 Similarly as in the previous section, it is considered here that we have an *a priori* estimation of the near-  
 460 optimal design  $\hat{\rho}$ . Therefore, the selection of the poses for the computation of the elastic performance will be  
 461 found by solving the following optimization problems

- 462 • in the case of the static deformations, for a given loading  $\mathbf{f}$ :

$$463 \begin{aligned} & \max_{\mathbf{q}} && c_1(\hat{\rho}, \mathbf{f}) \\ & \text{under} && \mathbf{q} \in [\mathbf{q}] \text{ and } \mathbf{x} \in [\mathbf{x}] \end{aligned} \quad (56)$$

464 Note that several loadings  $\mathbf{f}$  can be considered, resulting in the resolution of the problem (56) each time  
 465 that the loading is changed.

- 466 • in the case of the natural frequencies:

$$467 \begin{aligned} & \min_{\mathbf{q}} && c_2(\hat{\rho}) \\ & \text{under} && \mathbf{q} \in [\mathbf{q}] \text{ and } \mathbf{x} \in [\mathbf{x}] \end{aligned} \quad (57)$$

468 where the functions  $c_1$  and  $c_2$  are defined in (28) and (52), and represent the deformations under a loading  $\mathbf{f}$   
 469 and the first robot natural frequency, respectively.

#### 470 4.3. Discussion on the selection of near-optimal design parameters $\hat{\rho}$

471 The selection of the optimal robot configurations and trajectories defined above depends on an *a priori*  
 472 value of some near-optimal design parameters  $\hat{\rho}$ . This is an issue as we should not have an idea of them before  
 473 solving the design optimization problem.

474 A first approach would be to set  $\hat{\rho}$  as the starting point of the optimal design process, i.e. to set all  
 475 components in  $\hat{\rho}$  at 1. This is of course not the best. The approach can be refined by using an iterative  
 476 procedure:

- 477 1. First, set all components in  $\hat{\rho}$  at 1. Let us denote this vector as  $\hat{\rho}_0$ .
- 478 2. Then, solve the problems (55), (56) and (57) in order to find the exciting configurations  $\mathbf{q}_0$  and trajec-  
 479 tories  $\mathcal{T}_0$  corresponding to the design  $\hat{\rho}_0$ .
- 480 3. Run the design optimization problem (1) during  $\nu$  iterations (in our case study, 100 to 150 iterations  
 481 where enough in our case). Save the final value for the design parameters into the variable  $\hat{\rho}_{new}$ .
- 482 4. Start again the steps 2 and 3 until the change in the exciting configurations  $\mathbf{q}_{new}$  and trajectories  $\mathcal{T}_{new}$  is  
 483 “too big” (based on a criterion to be defined by the user).

484 We used this iterative procedure in what follows. In practice, we ran the previous algorithm during no  
 485 longer than three iterations and it was enough in order to obtain a robust solution.

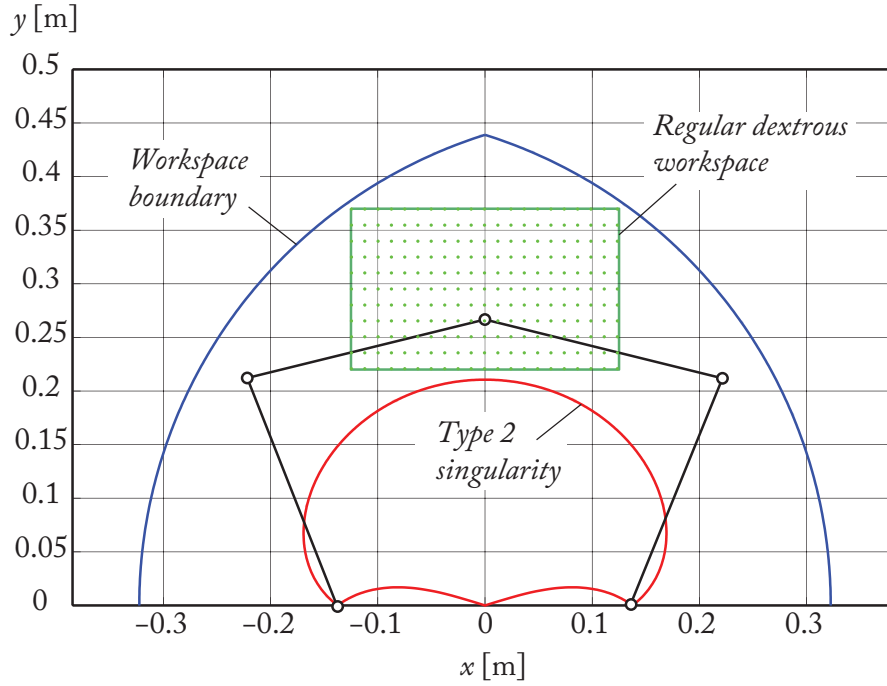


Figure 5: The five-bar mechanism under consideration and its workspace (to scale).

## 486 5. Case study: topology optimization of a five-bar mechanism

### 487 5.1. Definition of the optimal design problem in 2D

488 In this section, we propose as a case study the topology optimization of a five-bar mechanism (Fig. 3)  
 489 which is a robot composed of five revolute joints, two of them being active (at points  $A_1$  and  $A_2$ ) while the  
 490 other are passive. This mechanism with two dof is able to position in a plane its end-effector located at point  
 491  $C$  with coordinates  $(x, y)$ . The bodies between points  $A_i$  and  $B_i$  are called the proximal links and the bodies  
 492 between points  $B_i$  and  $C$  are called the distal links. We denote as “leg  $i$ ” ( $i = 1, 2$ ) the leg made with the joints  
 493 located at  $A_i$ ,  $B_i$  and  $C$ .

494 In what follows, we chose as link lengths the design parameters of the DexTAR robot [73], which is a  
 495 five-bar mechanism designed for high-speed pick-and-place operations. The link lengths are thus:  $\ell_{OA_1} =$   
 496  $\ell_{OA_2} = 0.1375$  m,  $\ell_{A_1B_1} = \ell_{A_2B_2} = 0.23$  m and  $\ell_{B_1C} = \ell_{B_2C} = 0.23$  m, where  $\ell_{PQ}$  is the distance between any  
 497 point  $P$  and  $Q$  (Fig. 3).

498 The workspace of the DexTAR robot under consideration is plotted in Fig. 5 with the Type 2 singularity  
 499 loci [74] corresponding to the robot working mode represented on the picture (branch index  $-1$  for leg 1,  $+1$   
 500 for leg 2 [75]).

501 In this workspace, we define a rectangular region, associated with the aforementioned branch indices for  
 502 the robot legs, in which we decide to guarantee the robot performance performance as was done in [76]. This  
 503 region is called the regular dextrous workspace [77] and is defined here as a rectangle centred in  $(0, 0.295)$  m  
 504 of width equal to 0.25 m and of height equal to 0.15 m (Fig. 5). This region fixes the boundaries of the  
 505 space search for finding the optimal robot configurations and trajectories for the computation of the robot  
 506 performance.

507 We impose that both proximal (distal, resp.) links have the same shape for two main reasons:

- 508 • this will allow a symmetrical distribution of the mechanism performance in the dextrous workspace  
 509 with respect to the  $y$ -axis
- 510 • the number of design variables is divided by two in this case, which decreases the computational time  
 511 for the robot elastic performance.

512 Finally, we decide to solve the following optimization problem:



- 513 • the objective is to minimize the RMS of the actuator torques for any trajectory defined as follows:
  - 514 – the trajectories are parameterized by fifth-order polynomial motion profiles [64],
  - 515 – the regular dextrous workspace is discretized with a grid made of  $21 \times 11 = 231$  points, as shown
  - 516 in Fig. 5, and the initial and final points for the trajectories are selected among these 231 points
  - 517 – the duration  $t_f$  of the trajectory between the initial and final points is parameterized by the function
  - 518  $t_f = 0.75^p$  for  $p$  an integer,  $p \in \{0, 1, 2, \dots, 8\}$  (thus  $t_f \in [0.1001, 1]$ )
  - 519 – the exciting trajectory  $\mathcal{T}^*$  selected for being used in the optimal design problem are found by solving
  - 520 the optimization problem (55) when considering near-optimal design parameters  $\hat{\rho}$  in which
  - 521 all components are equal to 1 for the initial optimization round, or equal to the optimized design
  - 522 parameters at the previous design optimization round.

523 Due to the symmetry imposed in the leg design and in the location and size of the dextrous operational  
 524 workspace, we consider only the RMS of the actuator of leg 1.

- 525 • the constraints are:
  - 526 – to ensure that the elastic translational displacement  $\delta_C$  of the end-effector under both external
  - 527 loadings  $\mathbf{f}_1 = [0 \text{ N} \ 100 \text{ N} \ 1 \text{ Nm}]^T$  and  $\mathbf{f}_2 = [100 \text{ N} \ 0 \text{ N} \ -1 \text{ Nm}]^T$  (where the components of
  - 528  $\mathbf{f}_i$  represent the force applied at  $C$  along the  $x$  and  $y$ -axes and the moment around  $z$ , respectively)
  - 529 is lower than  $\delta_{\max} = 0.2$  mm (in terms of norm) and that the elastic rotational displacement  $\theta_C$  of
  - 530 the distal link of leg 1 at  $C$  is lower than  $\theta_{\max} = 0.25$  mrad
  - 531 – to ensure that the first natural frequency  $f_1$  is bigger than  $f_{\min} = 180$  Hz

532 wherever in the dextrous regular workspace. The selection of the exciting configurations  $\mathbf{q}_i^\delta, \mathbf{q}_i^\theta$  ( $i = 1, 2$ )  
 533 and  $\mathbf{q}^f$  for the computation of these performance indices is based on the methodology explained in  
 534 Section 4.2 (where  $\mathbf{q}_i^\delta$  is the selected exciting configuration for the computation of the translational  
 535 deformation when the loading  $\mathbf{f}_i$  is applied, while  $\mathbf{q}_i^\theta$  is the selected configuration for the computation of  
 536 the rotational deformation for the same loading;  $\mathbf{q}^f$  is the configuration for the computation of the first  
 537 natural frequency).

538 The design optimization problem can thus be formalized as

$$\begin{aligned}
 & \min_{\rho} \quad F = \bar{\tau}_1^2(\mathcal{T}^*) \\
 & \text{subject to} \quad g_1 = \delta_C^2(\mathbf{f}_1, \mathbf{q}_1^\delta) - \delta_{\max}^2 \leq 0 \\
 & \quad \quad \quad g_2 = \theta_C^2(\mathbf{f}_1, \mathbf{q}_1^\theta) - \theta_{\max}^2 \leq 0 \\
 & \quad \quad \quad g_3 = \delta_C^2(\mathbf{f}_2, \mathbf{q}_2^\delta) - \delta_{\max}^2 \leq 0 \\
 & \quad \quad \quad g_4 = \theta_C^2(\mathbf{f}_2, \mathbf{q}_2^\theta) - \theta_{\max}^2 \leq 0 \\
 & \quad \quad \quad g_5 = f_{\min} - f_1(\mathbf{q}^f) \leq 0
 \end{aligned} \tag{58}$$

540 In the next section, results obtained when applying this methodology are presented.

## 541 5.2. Results and discussion for the 2D model

### 542 5.2.1. Results

543 The initial design domain for the proximal and distal links is represented in Fig. 6. Each link is made  
 544 of two holes (voids) at its extremity in which the joints will be inserted (axis of the motor at points  $A_i$ , and  
 545 passive revolute joints for all other holes). For the meshing of the links, QUA4 finite elements (i.e. four-nodes  
 546 rectangular planar elements) of size  $1 \times 1$  mm and thickness of 2 cm are used. The planar stress assumption is  
 547 used. This assumption is valid because all links are considered coplanar and the efforts are applied in the links  
 548 plane. If these hypotheses were not verified, other assumptions should have been taken into account [78].

549 Links are considered to be made of steel with Young's modulus  $E_0 = 210$  GPa, Poisson's ratio  $\nu = 0.3$   
 550 and density of  $7800$  kg/m<sup>3</sup>. As a result, 20932 elements are used for meshing the proximal links while the  
 551 distal links are made of 14208 elements (Figs. 7(a) and 7(b)).

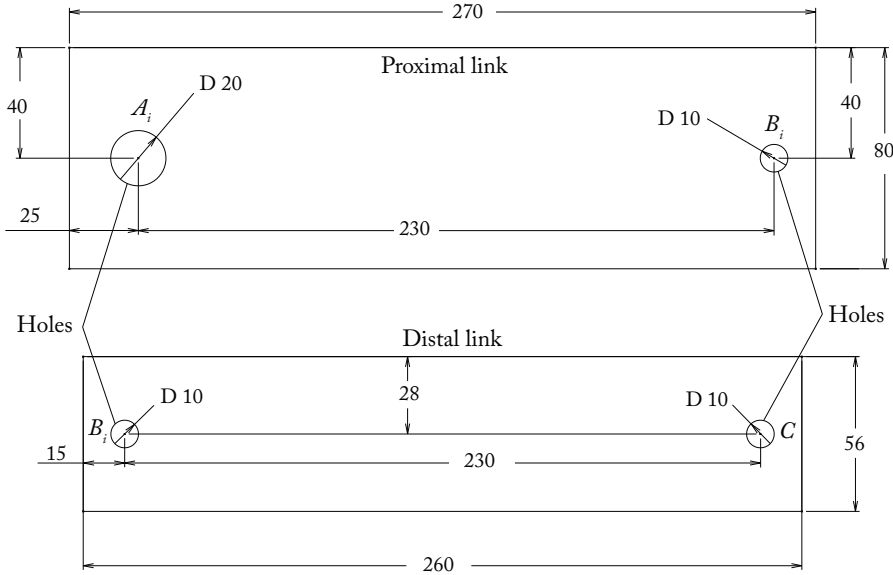


Figure 6: Initial design domain for the design of the robot link (to scale). All dimensions are in millimeters.

552 For all optimization rounds, in order to obtain a smoother layout without checkerboards problem, a filter-  
 553 ing procedure is used that consists in the modification of the density variables assigned to the elements with  
 554 the information of its neighborhoods as was proposed in [79].

555 Then, we applied our approach in order to obtain the optimal design of our mechanism. All models and  
 556 optimization algorithms have been encoded with Matlab in the Windows 7 environment.

557 We started with the selection of the configurations and trajectories for the computation of the robot perfor-  
 558 mance. The selected configurations for the elastic performance computation are given in Tabs. 1 and 2 (row  
 559 corresponding to round 1) while the trajectory of the computation of the torques is given in Fig. 8. For this  
 560 first round, we use links made of the full initial design domain for the computation of the elastic and dynamic  
 561 performance (Figs. 7(a) and 7(b)).

562 We then ran the optimization algorithm during 100 iterations. The results in terms of design for the two  
 563 links is shown in Figs. 7(c) and 7(d).

564 We then selected new configurations and trajectories based on the use of the near optimal design of the  
 565 links obtained at the end of the first round. The selected configurations for the elastic performance computation  
 566 are given in Tabs. 1 and 2 (row corresponding to round 2) while the trajectory of the computation of the torques  
 567 is the same as the one presented in Fig. 8. We can see that the configuration  $\mathbf{q}_2^0$  for the computation of the  
 568 rotational deformations when the wrench  $\mathbf{f}_2$  is applied changed. This is of critical importance for the global  
 569 robustness of the final solution.

570 We then ran the optimization algorithm during 150 iterations and we stopped the computation. The results  
 571 in terms of design for the two links is shown in Figs. 7(e) and 7(f).

572 Once again, we selected new configurations and trajectories based on the use of the near optimal design  
 573 of the links obtained at the end of the first round. The new selected configurations for the elastic performance  
 574 computation are given in Tabs. 1 and 2 (row corresponding to round 3) while the trajectory of the computation  
 575 of the torques is still the same as the one presented in Fig. 8. We can see now that all configurations and  
 576 trajectories are unchanged. Therefore we went for the final round of optimization.

577 The final results in terms of design for the two links is shown in Figs. 7(g) and 7(h). For this last round,  
 578 there was no constraint on the number of maximal iteration, we waited for the convergence of the algorithm.  
 579 We consider that the algorithm converged when the maximal change between two sequential iterations for any  
 580 component of the density vector  $\rho$  is lower than 0.01. The algorithm stopped after 597 iterations, with a max-  
 581 imal constraint violation of  $8 \cdot 10^{-3} \%$  (the maximal deformation in rotation in the whole workspace is equal  
 582 to 0.250008 mrad, instead of 0.25 mrad which is negligible). In totality, the full procedure of optimization  
 583 (including the selections of the trajectories and configurations) took 98 minutes with a Pentium 4 2.70 GHz,

Table 1: Configurations (positions of the end-effector) used for the computation of the elastostatic performance

	Const. for translational displacements				Const. for rotational displacements			
	$\mathbf{q}_1^\delta$		$\mathbf{q}_2^\delta$		$\mathbf{q}_1^\theta$		$\mathbf{q}_2^\theta$	
	$x$ [m]	$y$ [m]	$x$ [m]	$y$ [m]	$x$ [m]	$y$ [m]	$x$ [m]	$y$ [m]
Round 1	0	0.25	-0.0625	0.25	0	0.25	-0.0625	0.25
Round 2	0	0.25	-0.0625	0.25	0	0.25	0.125	0.37
Round 3	0	0.25	-0.0625	0.25	0	0.25	0.125	0.37

Table 2: Configurations (positions of the end-effector) used for the computation of the elastodynamic performance

	$\mathbf{q}^f$	
	$x$ [m]	$y$ [m]
Round 1	0	0.25
Round 2	0	0.25
Round 3	0	0.25

584 16 GB of RAM.

585 For this optimal design, the RMS of the torques along the trajectory shown in Fig. 8 is of 98 Nm while it  
586 was of 27 Nm for the final design, i.e. the torque RMS was divided by almost 4.

587 The values of the deformations and natural frequencies within the dextrous workspace are shown in Fig. 9.  
588 We can see that the deformations (first natural frequency, resp) are lower (bigger, resp.) than the defined  
589 acceptable values, i.e. that our approach led to a robust optimization of the robot performance over the  
590 whole workspace.

591 In order to show the importance of the update of the value of  $\mathbf{q}_2^\theta$  between Round 1 and Round 2, we  
592 optimized the robot for constraints calculated for the configurations and trajectories selected at the initial  
593 round only (row corresponding to round 1 in Tabs. 1 and 2 for the configurations, Fig. 8 for the trajectory).  
594 The algorithm stopped after 824 iterations, and 94 minutes of computation on the same computer as previously.  
595 The final optimized links are shown in Fig. 10 and the values of the deformations and natural frequencies are  
596 depicted in Fig. 11. It can be shown that the desired boundaries on the deformations are not respected (the  
597 maximal deformation in rotation in the whole workspace is equal to 0.35 mrad, which is far beyond the limit  
598 of 0.25 mrad – see Fig. 11(c)). This shows that it is critical to update the selected worst case configuration /  
599 trajectory during the solving process.

### 600 5.2.2. Discussions

601 *Comparison of LM and MMA in terms of convergence time.* In order to show the efficiency of the Linearization  
602 Method (LM) for this class of problem, we decided to solve the optimization problem (58) with another  
603 optimization method known for its computational efficiency: the Method of Moving Asymptotes (MMA) [53].  
604 We used the Matlab code kindly provided by Prof. K. Svanberg and ran the optimization problem (58) by  
605 defining the same stopping conditions as previously. The MMA algorithm stopped after 1785 iterations. In  
606 totality, the full procedure of optimization (including the selections of the trajectories and configurations) took  
607 143 minutes on the mentioned computer, so almost 45 % of additional time with respect to LM, however, the  
608 final torque RMS was of 25.9 Nm, i.e. MMA proposes a robot whose torque RMS on the exciting trajectory is  
609 lower than with LM, but the difference is very low (1 Nm, i.e. almost 1 % of the final objective). The results  
610 in terms of robot link design with MMA are shown in Fig. 12. Surprisingly, they are sensibly different, which  
611 seems to indicate that several local minima exist.

612 The evolution of the torque RMS as a function of the computational time in the last optimization round  
613 is shown in Fig. 13. It can be seen than both functions reach the same values of the objective in almost the  
614 same time, even if MMA goes a bit lower (but in a more longer time) than LM. Thus, LM is a valuable and  
615 computationally-efficient method for topology optimization.

616 *Computational efficiency of the reduction techniques.* We compared the time requested for the computation  
617 of the constraints when:

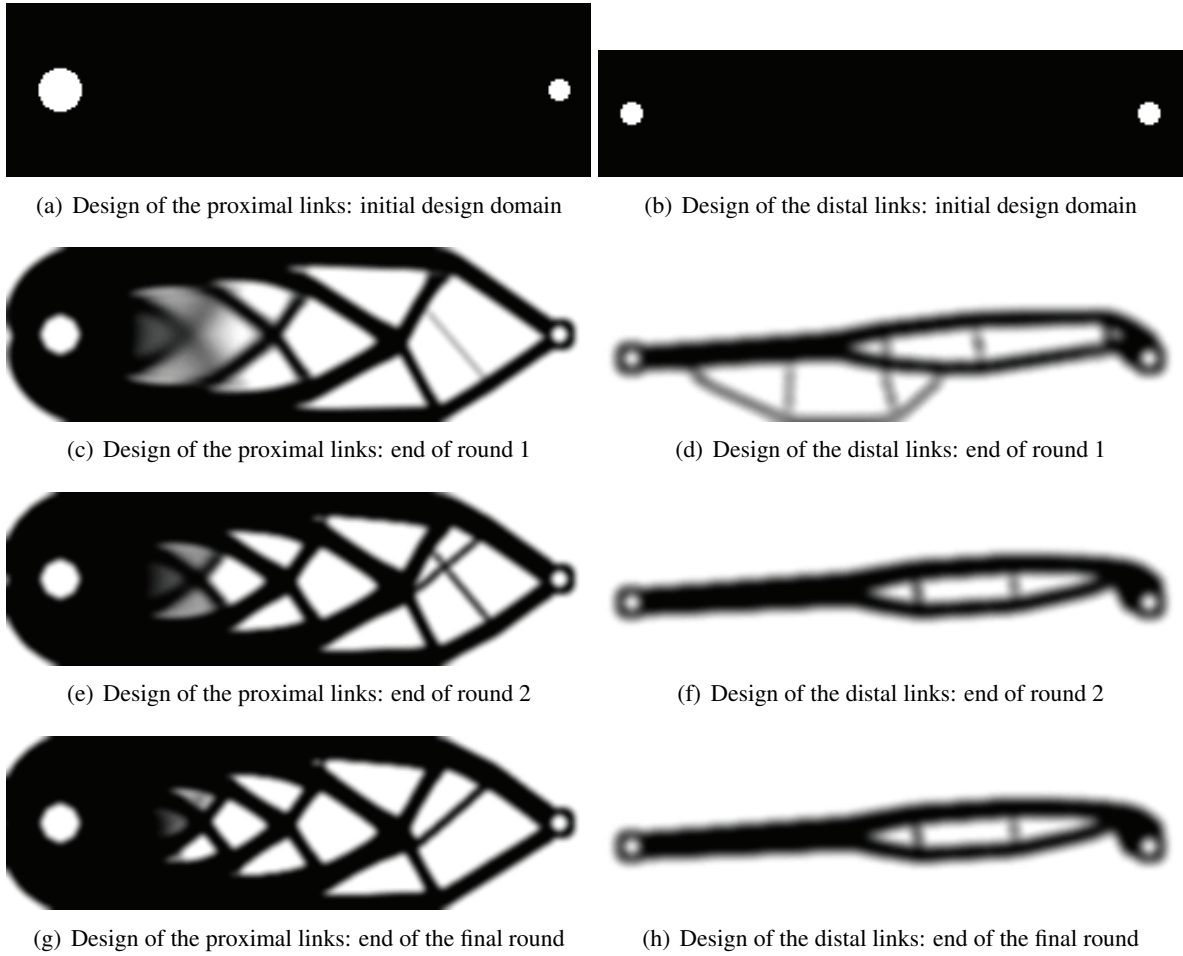


Figure 7: Evolution of the design of the five-bar links at the end of each optimization round: the links are shown in gray-scale (black elements correspond to  $\rho_{ij} = 1$ , white elements to  $\rho_{ij} = 0$ , and gray elements to  $0 < \rho_{ij} < 1$ )

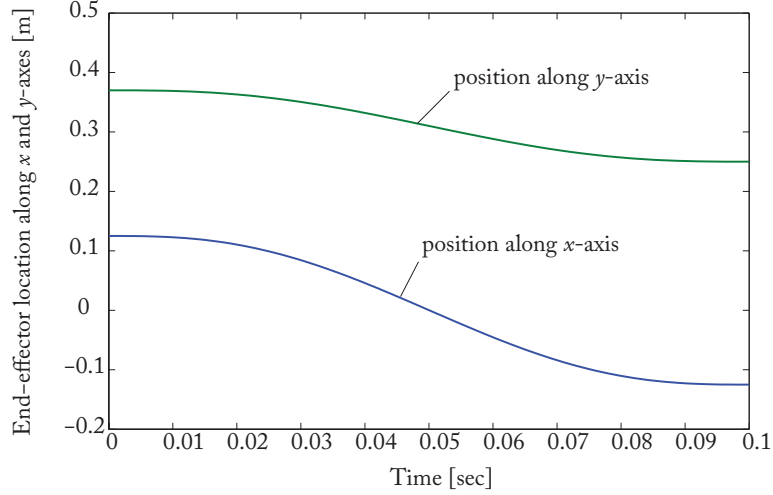


Figure 8: Trajectory used for the computation of the input torques.

- 618 • they are calculated with the model reduction techniques introduced in Section 3
- 619 • they are calculated without model reduction techniques, as shown in Appendix C and Appendix D.

620 On the aforementioned computer, the computation of the constraints when using model reduction tech-  
621 niques took 4.19 sec. When using the full models, it took 8.22 sec. In order to better understand the benefits of  
622 the model reduction technique, we plotted in Fig. 14 the evolution of computational cost for the computation  
623 of the first natural frequency as a function of the number of tested configurations. For the computation of  
624 a single configuration, the computational costs are equivalent (4.41 sec for the model reduction technique,  
625 4.19 sec without it). The benefits appear when we compute the natural frequency for two configurations: in  
626 this case, the total computation time with the model reduction technique is of 4.52 sec, to be compared with  
627 8.37 sec in order to compute the frequency without reduction technique. For twenty configurations, the com-  
628 putation with model reduction techniques took 6.44 sec, instead of 85 sec without it. Therefore, using model  
629 reduction techniques allows the considerable decrease of the computational cost in robot design optimization.

### 630 5.3. Solving the problem in 3D

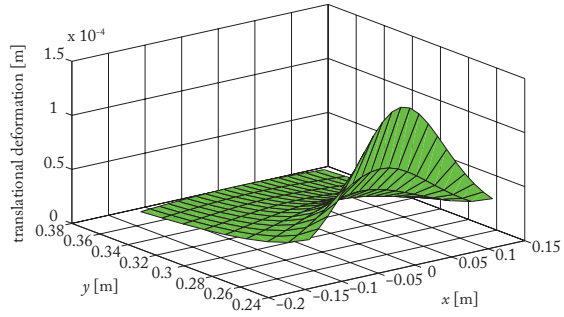
631 The optimization problem can also be solved in 3D, even if in such a case the computational time is  
632 increase. In Fig. 15, we show the proximal and distal links of a five-bar mechanism which have been obtained  
633 when solving the following optimization problem:

- 634 • the objective was to minimize the robot mass
- 635 • the constraints are to ensure that the elastic translational displacement  $\delta_C$  of the end-effector under the  
636 external loading  $\mathbf{f} = [0 \text{ N} \ 50 \text{ N} \ 10 \text{ N} \ 0 \text{ Nm} \ 0 \text{ Nm} \ 1 \text{ Nm}]^T$  (where the components of  $\mathbf{f}$  represent  
637 the force applied at  $C$  along the  $x$ ,  $y$  and  $z$ -axes and the moment around the  $x$ ,  $y$  and  $z$ -axes, respectively)  
638 is lower than  $\delta_{\max} = 0.025$  mm (in terms of norm) and that the elastic rotational displacement  $\theta_C$  of the  
639 distal link of leg 1 at  $C$  is lower than  $\theta_{\max} = 0.25$  mrad wherever in the dextrous regular workspace. The  
640 selection of the exciting configurations  $\mathbf{q}^\delta$  and  $\mathbf{q}^\theta$  for the computation of these performance indices was  
641 the based in the methodology explained in Section 4.2.

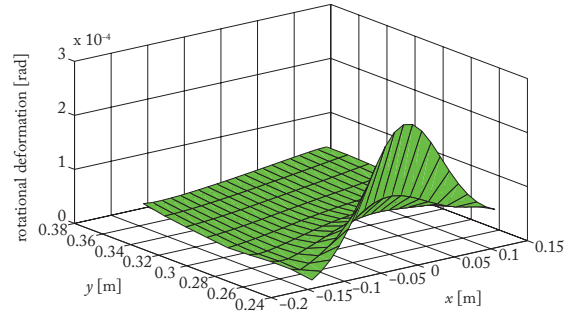
642 The design optimization problem can thus be formalized as

$$\begin{aligned}
& \min_{\rho} \quad F = m_{robot} \\
& \text{under} \quad g_1 = \delta_C^2(\mathbf{f}, \mathbf{q}^\delta) - \delta_{\max}^2 \leq 0 \\
& \quad \quad g_2 = \theta_C^2(\mathbf{f}, \mathbf{q}^\theta) - \theta_{\max}^2 \leq 0
\end{aligned} \tag{59}$$

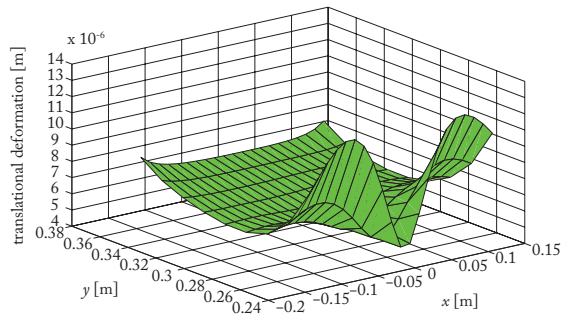
644 Here, we chose as link lengths the design parameters of the DexTAR robot [73] from the company Meca-  
645 demic. The link lengths are thus:  $\ell_{OA_1} = \ell_{OA_2} = 0.056$  m,  $\ell_{A_1B_1} = \ell_{A_2B_2} = 0.09$  m and  $\ell_{B_1C} = \ell_{B_2C} = 0.09$  m



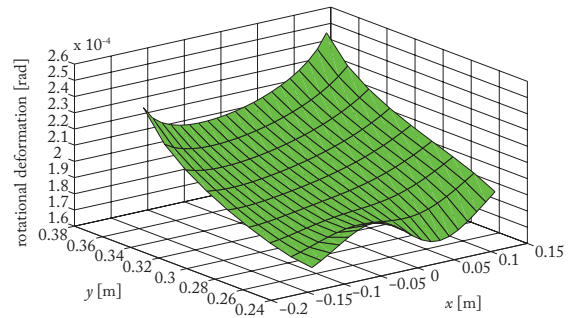
(a) Norm of the translational deformations under the application of the loading  $\mathbf{f}_1$



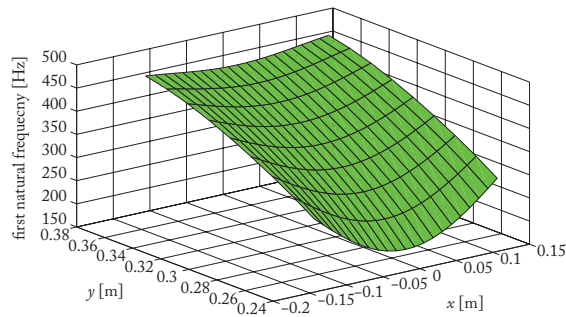
(b) Absolute value of the rotational deformations under the application of the loading  $\mathbf{f}_1$



(c) Norm of the translational deformations under the application of the loading  $\mathbf{f}_2$



(d) Absolute value of the rotational deformations under the application of the loading  $\mathbf{f}_2$



(e) First natural frequency

Figure 9: Evolution of the deformations and of the first natural frequency in the workspace for the optimized design

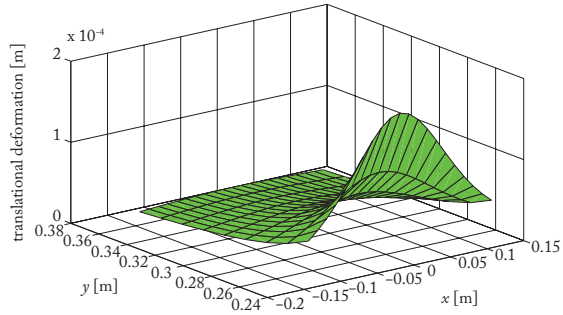


(a) Design of the proximal links

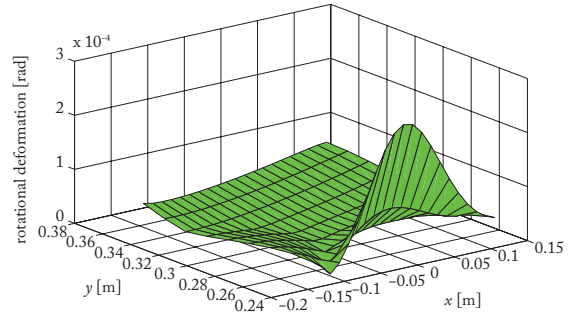


(b) Design of the distal links

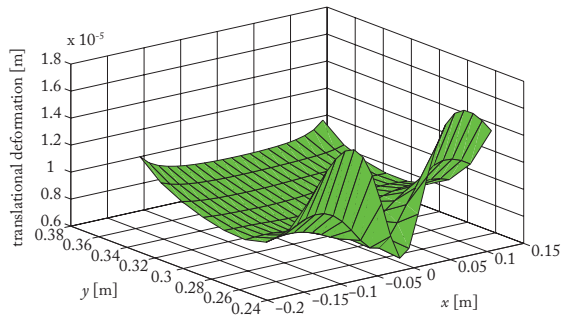
Figure 10: Optimal designs of the five-bar links using the configurations and trajectories selected at the initial round for the computation of the objective and constraints: the links are shown in gray-scale (black elements correspond to  $\rho_{ij} = 1$ , white elements to  $\rho_{ij} = 0$ , and gray elements to  $0 < \rho_{ij} < 1$ )



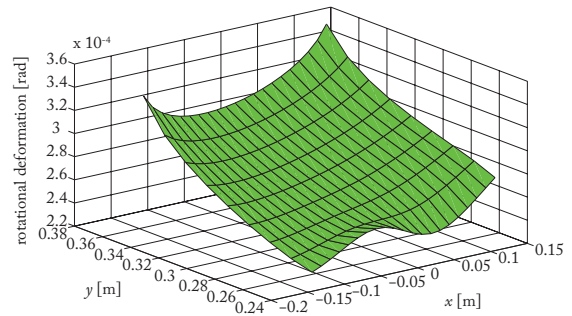
(a) Norm of the translational deformations under the application of the loading  $\mathbf{f}_1$



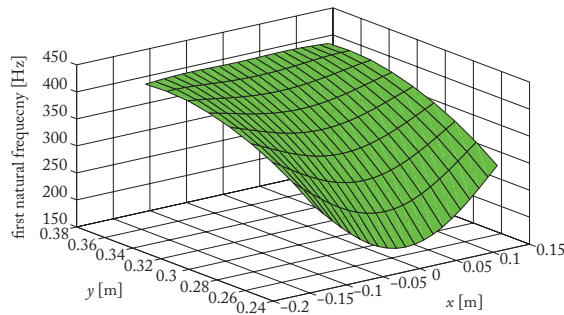
(b) Absolute value of the rotational deformations under the application of the loading  $\mathbf{f}_1$



(c) Norm of the translational deformations under the application of the loading  $\mathbf{f}_2$



(d) Absolute value of the rotational deformations under the application of the loading  $\mathbf{f}_2$



(e) First natural frequency

Figure 11: Evolution of the deformations and of the first natural frequency in the workspace for the optimized design using the configurations and trajectories selected for the initial round.



(a) Design of the proximal links



(b) Design of the distal links

Figure 12: Optimal designs of the five-bar links using MMA algorithm: the links are shown in gray-scale (black elements correspond to  $\rho_{ij} = 1$ , white elements to  $\rho_{ij} = 0$ , and gray elements to  $0 < \rho_{ij} < 1$ )

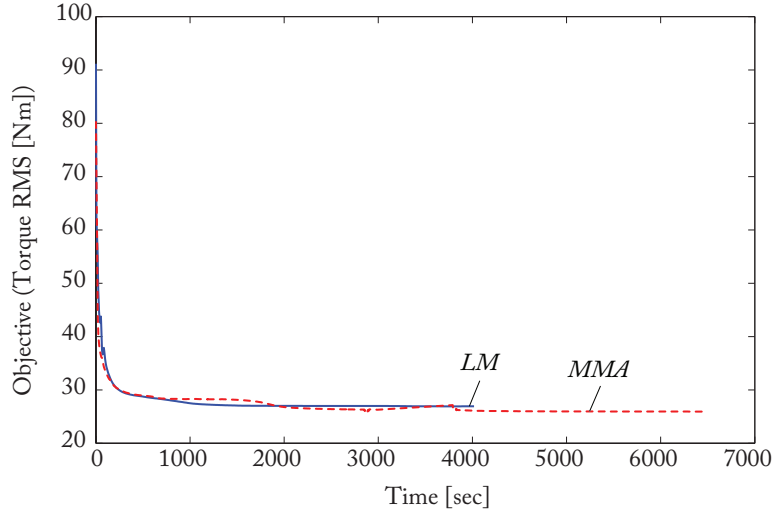


Figure 13: Torque RMS as a function of the computational time in the last optimization round for LM (full line) and MMA (dotted line) procedures.

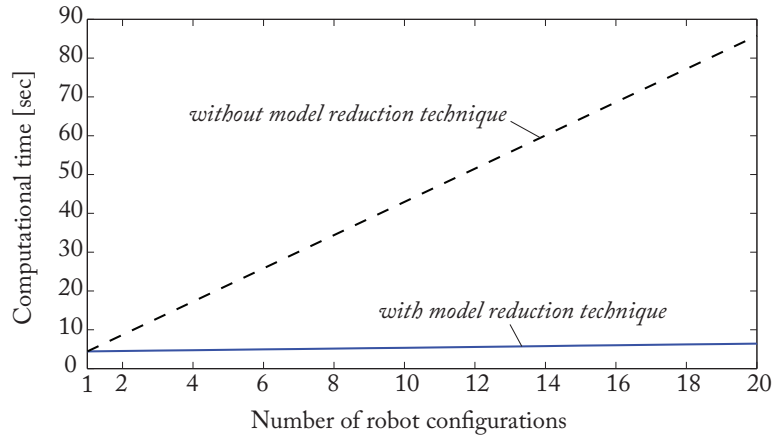


Figure 14: Evolution of computational cost for the computation of the first natural frequency as a function of the number of tested configurations.

646 (Fig. 3). The selected dextrous regular workspace is a rectangle of dimension  $12 \times 4$  cm centred in  $x = 0$  m  
 647 and  $y = 0.11$  m.

648 The initial design domain for the proximal and distal links is represented in Fig. 16 (planar view, the initial  
 649 width of both links is of 20 mm). For the meshing of the links, HEXA8 finite elements (i.e. eight-nodes cubic  
 650 planar elements) of size  $1.3 \times 1.3 \times 1.3$  mm are used. Links are considered to be made of steel with Young's  
 651 modulus  $E_0 = 210$  GPa, Poisson's ratio  $\nu = 0.3$  and density of  $7800$  kg/m<sup>3</sup>. As a result, 75200 elements are  
 652 used for meshing the proximal links while the distal links are made of 21760 elements.

653 We consider that the algorithm converged when the maximal change between two sequential iterations for  
 654 any component of the density vector  $\rho$  is lower than 0.05. The algorithm stopped after 4 hours and 42 minutes  
 655 of computation with a Pentium 4 2.70 GHz, 24 GB of RAM.

656 In order to accelerate the computation of the elastostatic properties by avoiding the inversion of the ma-  
 657 trices  $\mathbf{K}_i^{(11)}$  in (20) for both links, we computed the matrices  $\Phi_{si}$  using a “pcg” (Preconditioned Conjugate  
 658 Gradient) algorithm [80]. Indeed, we computed each column of the matrix  $\Phi_{si}$  by solving the system of  
 659 equations with the “pcg” algorithm:

$$660 \quad \mathbf{K}_i^{(11)} \Phi_{sij} = -\mathbf{K}_{ij}^{(12)} \quad (60)$$

661 in which  $\Phi_{sij}$  ( $\mathbf{K}_{ij}^{(12)}$ , resp.) is the  $j$ th column of  $\Phi_{si}$  ( $\mathbf{K}_i^{(12)}$ , resp.). The computation can be accelerated by



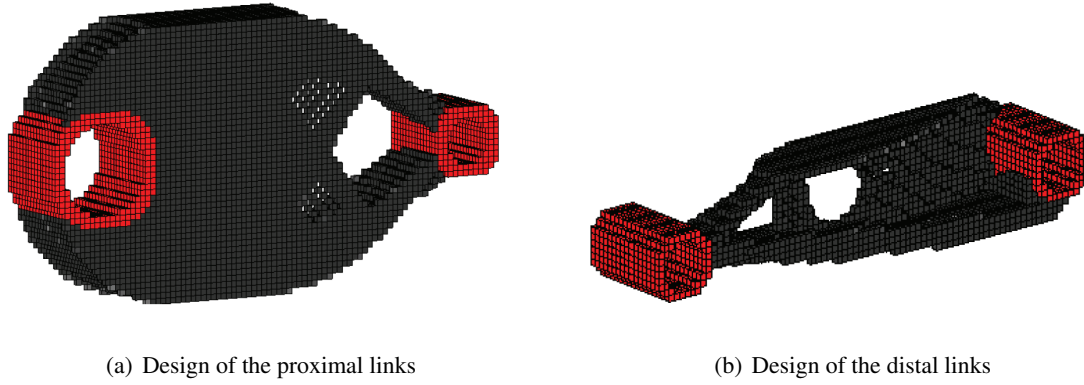


Figure 15: Optimal designs of the five-bar links using 3D models

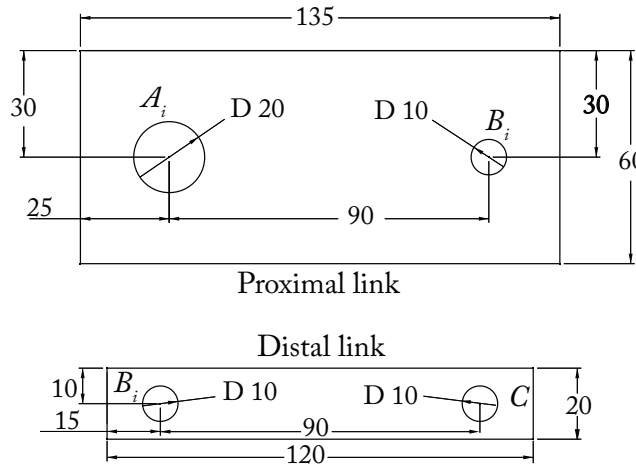


Figure 16: Initial design domain for the design of the robot link (planar view, to scale). The initial width of both links is of 20 mm. All dimensions are in millimeters.

662 doing a computation in parallel of each column of the matrix  $\Phi_{s_i}$  (indeed, we used 6 CPUs in parallel on our  
663 computer to do so).

664 We also introduced in the optimization problem some constraints on the limits of the first natural fre-  
665 quency. The results obtained were similar to the ones shown in Fig. 15, but with much longer computational  
666 time (around 6 days and 8 hours). This considerable increase of the computational cost was due to the calcu-  
667 lation of the matrices  $\Phi_{d_i}$  in (36) for both links which is obtained by solving an eigenvalue problem of large  
668 dimension (matrices of size  $(256581 \times 256581)$  for the proximal links). Furthermore, it should be mentioned  
669 that this eigenvalue problem cannot be solved in Matlab with the classical “eig” or “eigs” functions which  
670 requests too much memory allocation (our computer went out of memory), but with a “lobpcg” (Locally Op-  
671 timal Block Preconditioned Conjugate Gradient) algorithm [81] which is known to be efficient for computing  
672 eigenvalues problems of very large dimensions.

## 673 6. Conclusion

674 Topology optimization aims at optimizing the material distribution in a link or a set of links in order to sat-  
675 isfy performance criteria related to the link elastic behavior. Recent works introduced topology optimization  
676 in the design of robots, but the proposed methodology led to a local optimization of the robot performance (for  
677 a few trajectories or configurations). Moreover, most of performance indices used are not in strong relation  
678 with easy-to-understand technological requirements.

679 In the present paper, our primary contribution was the proposition of a methodology that was able to per-  
680 form a topology optimization for robots, valid globally in the workspace or for a set of given trajectories, which

681 was based on the use of technology-oriented performance criteria. In order to enforce the chosen performance  
682 indices to be globally valid, optimal robot configurations or trajectories for which extreme performance will  
683 be attained are computed, and iteratively updated.

684 In order to decrease the computational time associated with the computation of performance indices in  
685 numerous configurations, we exploited the structure of the elastic models in order to reduce their computa-  
686 tional complexity. Indeed, we showed that it was possible to use configuration-independent model reduction  
687 techniques in order to considerably decrease the size of the stiffness and mass matrices of each link, expressed  
688 in their local frame. Then, these reduced matrices are used in order to build the configuration-dependent robot  
689 elastic models which are shown to be computationally efficient.

690 We used an optimization algorithm called the Linearization Method which has proven, for our class of  
691 problems, to give results in a computational time equivalent to standard topology optimization algorithms but  
692 its implementation was less complex and made it quite easy to perform modification or improvement.

693 The methodology was successfully applied for the design of a five-bar mechanism and we showed that  
694 our approach led to a robust optimization of the robot performance over the whole workspace. We also  
695 showed that the Linearization Method algorithm was as competitive as the standard algorithms for topology  
696 optimization in terms of convergence time, such as the Method of Moving Asymptotes.

697 Finally, we would like to mention that the proposed approach is not software dependent: we used Mat-  
698 lab for finite element modeling for reasons of simplicity, but co-simulation with any FEA software able to  
699 communicate with Matlab (or another software allowing the encoding of the optimization procedure) could  
700 be used.

## 701 Appendix A. Inertial parameters

702 Inertial parameters are performance indices which are global by nature, as they do not depend on the robot  
703 configuration. As known from [64], each robot body can be characterized by ten (rigid) inertial parameters: its  
704 mass, three first moments of inertia (also called static moments), three moments of inertia and three products  
705 of inertia.

706 Let us consider the body  $i$  meshed with  $m$  elements. For the element  $j$  of the body  $i$ , denoted as the element  
707  $ij$ , we define as:

- 708 •  $x_{ij}$ ,  $y_{ij}$  and  $z_{ij}$  the position of the origin  $M_{ij}$  of the element  $ij$  in the local frame  $(O_i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$  attached  
709 to the body  $i$  (Fig. 4).
- 710 •  $\rho_{ij}$  the density associated with the element  $ij$ ,
- 711 •  $m_{ij}$  the mass of this element,

712 As a result, and from [64], we have the following formulas for the computation of the inertial parameters:

- 713 • body's mass:  $m_i = \sum_{j=1}^m m_{ij}\rho_{ij}$ ,
- 714 • body's first moments of inertia:  $mx_i = \sum_{j=1}^m m_{ij}x_{ij}\rho_{ij}$ ,  $my_i = \sum_{j=1}^m m_{ij}y_{ij}\rho_{ij}$  and  $mz_i = \sum_{j=1}^m m_{ij}z_{ij}\rho_{ij}$ ,
- 715 • body's moments of inertia:  $xx_i = \sum_{j=1}^m m_{ij}(y_{ij}^2 + z_{ij}^2)\rho_{ij}$ ,  $yy_i = \sum_{j=1}^m m_{ij}(x_{ij}^2 + z_{ij}^2)\rho_{ij}$ ,  $zz_i = \sum_{j=1}^m m_{ij}(x_{ij}^2 +$   
716  $y_{ij}^2)\rho_{ij}$ ,
- 717 • body's products of inertia:  $xy_i = \sum_{j=1}^m m_{ij}x_{ij}y_{ij}\rho_{ij}$ ,  $xz_i = \sum_{j=1}^m m_{ij}x_{ij}z_{ij}\rho_{ij}$ ,  $yz_i = \sum_{j=1}^m m_{ij}y_{ij}z_{ij}\rho_{ij}$ .

718 Grouping all inertial terms for the body  $i$  in the vector  $\chi_i = [m_i \ mx_i \ my_i \ mz_i \ xx_i \ yy_i \ zz_i \ xy_i \ xz_i \ yz_i]^T$ , we

719 finally have the following expression:

$$720 \quad \chi_i = \Psi_i \rho_i, \text{ where } \Psi_i = m_{ij} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_{i1} & x_{i2} & \dots & x_{im} \\ y_{i1} & y_{i2} & \dots & y_{im} \\ z_{i1} & z_{i2} & \dots & z_{im} \\ y_{i1}^2 + z_{i1}^2 & y_{i2}^2 + z_{i2}^2 & \dots & y_{im}^2 + z_{im}^2 \\ x_{i1}^2 + z_{i1}^2 & x_{i2}^2 + z_{i2}^2 & \dots & x_{im}^2 + z_{im}^2 \\ x_{i1}^2 + y_{i1}^2 & x_{i2}^2 + y_{i2}^2 & \dots & x_{im}^2 + y_{im}^2 \\ x_{i1}y_{i1} & x_{i2}y_{i2} & \dots & x_{im}y_{im} \\ x_{i1}z_{i1} & x_{i2}z_{i2} & \dots & x_{im}z_{im} \\ y_{i1}z_{i1} & y_{i2}z_{i2} & \dots & y_{im}z_{im} \end{bmatrix} \text{ and } \rho_i = \begin{bmatrix} \rho_{i1} \\ \rho_{i2} \\ \vdots \\ \rho_{im} \end{bmatrix} \quad (\text{A.1})$$

721 in which  $m_{ij}$  is a constant mass as we consider here that the body is meshed with equal size elements. The  
722 matrix  $\Psi_i$  is the Jacobian of  $\chi_i$  with respect to  $\rho_i$ .

723 Grouping all inertial terms for all bodies in the vector  $\chi = [\chi_1^T \dots \chi_n^T]^T$  ( $n$  being the number of robot  
724 bodies), we finally get the expression:

$$725 \quad \chi = \Psi \rho, \text{ where } \Psi = \begin{bmatrix} \Psi_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \Psi_n \end{bmatrix} \text{ and } \rho = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_n \end{bmatrix} \quad (\text{A.2})$$

726 Any linear combination of the inertial parameters can be then obtained by multiplying the vector  $\chi$  by a row  
727 vector  $\mathbf{a}$  leading to a performance criterion  $c_1$  equal to

$$728 \quad c_3 = \mathbf{a}\chi = \mathbf{a}\Psi\rho \quad (\text{A.3})$$

729 The performance criterion  $c_3$  is linear with the decision variables  $\rho$  and the computation of its derivative with  
730 respect to  $\rho$  is straightforward.

731 Note that the linear combinations allowing the computation of the grouped inertial parameters with the  
732 most effects on the dynamics of the robot can be found using the methods [82, 83].

## 733 Appendix B. Input efforts and energy consumption

734 From [64], we know that the input efforts, power and energy functions of any robot (assuming that the  
735 robot is rigid) are linear with respect to the inertial parameters, i.e. we have:

$$736 \quad \tau = \Phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\chi \quad (\text{B.1})$$

$$737 \quad P = \dot{\mathbf{q}}^T \tau = \dot{\mathbf{q}}^T \Phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\chi \quad (\text{B.2})$$

$$738 \quad H = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})\chi \quad (\text{B.3})$$

739 where  $\tau$  is the vector of robot input efforts,  $P$  is the power delivered by the actuators,  $H$  is the total energy of  
740 the system,  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$  are the robot active joint position, velocity and acceleration, respectively, and the matrix  
741  $\Phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  is the Jacobian matrix of  $\tau$  with respect to  $\chi$  (and is also called the observation matrix) while  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$   
742 is the Jacobian matrix of  $H$  with respect to  $\chi$ . By introducing (A.2) into (B.1) and (B.3), we obtain

$$743 \quad \tau = \Gamma(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\rho, \text{ where } \Gamma(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \Phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\Psi \quad (\text{B.4})$$

$$744 \quad P = \alpha(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\rho, \text{ where } \alpha = \dot{\mathbf{q}}^T \Gamma(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \dot{\mathbf{q}}^T \Phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\Psi \quad (\text{B.5})$$

$$745 \quad H = \beta(\mathbf{q}, \dot{\mathbf{q}})\rho, \text{ where } \beta(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})\Psi \quad (\text{B.6})$$

746 The power  $P$  and energy  $H$  could be used directly as performance criteria. However, two different (but  
747 complementary) performance indices could be defined from the expression of the input efforts:

- 748 • the infinite norm  $\|\tau_i\|_\infty$  of the input effort for the motor  $i$  ( $\tau_i$  being the  $i$ th component of the vector  $\tau$ )  
749 along the trajectory  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  that should be lower than the actuator peak effort specification,
- 750 • the root-mean-square  $\bar{\tau}_i$  of the input effort for the motor  $i$  along the trajectory  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  than should be  
751 lower than the actuator continuous effort specification.

752 In order to compute these indices, let us sample the trajectory  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  at a given frequency, leading to  
753  $N$  samples for  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ . We denote as  $\mathbf{y}_i$  the vector containing all values of  $\tau_i$  computed along the sampled  
754 trajectory  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ : the  $k$ th component of  $\mathbf{y}_i$  is equal to  $\tau_i^{(k)} = \Gamma_i(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k) \rho$ , where  $(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k)$  is the value of  
755  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  at the sample  $k$  and  $\Gamma_i$  is the  $i$ th row of  $\Gamma$ . As a result,  $\mathbf{y}_i$  takes the form:

$$756 \quad \mathbf{y}_i = \mathbf{W}_i \rho, \text{ where } \mathbf{W}_i = \begin{bmatrix} \Gamma_i(\mathbf{q}_1, \dot{\mathbf{q}}_1, \ddot{\mathbf{q}}_1) \\ \vdots \\ \Gamma_i(\mathbf{q}_N, \dot{\mathbf{q}}_N, \ddot{\mathbf{q}}_N) \end{bmatrix} \quad (\text{B.7})$$

757 By definition, we have

$$758 \quad \|\tau_i\|_\infty = \max \sqrt{\mathbf{y}_i * \mathbf{y}_i} \quad (\text{B.8})$$

759 in which the operator “\*” represents the term-by-term product and

$$760 \quad \bar{\tau}_i = \sqrt{\frac{\mathbf{y}_i^T \mathbf{y}_i}{N}} \quad (\text{B.9})$$

761 In order to avoid conserving square roots in the expressions of the performance criteria, it is better to use  
762 as performance index the square of the infinite norm  $\|\tau_i\|_\infty$  and of the root-mean-square  $\bar{\tau}_i$ . As a result, by  
763 using (B.7), the selected performance criteria based on the input efforts are:

$$764 \quad c_4 = \|\tau_i\|_\infty^2 = \max(\mathbf{y}_i * \mathbf{y}_i) = \max(\mathbf{W}_i \rho * \mathbf{W}_i \rho) \quad (\text{B.10})$$

$$765 \quad c_5 = \bar{\tau}_i^2 = \frac{1}{N} \mathbf{y}_i^T \mathbf{y}_i = \frac{1}{N} \rho^T \mathbf{W}_i^T \mathbf{W}_i \rho \quad (\text{B.11})$$

766 The derivatives of these quadratic forms with respect to the decision variables is still straightforward. Obvi-  
767 ously, all these criteria will differ depending on the chosen trajectory  $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ . Discussion on the selection of  
768 an optimal trajectory is the aim of the Section 4.

### 769 Appendix C. Usual computation of the robot elastostatic model

770 Considering the robot made of  $n$  bodies, and by using the expression of the body potential energy (13),  
771 the full potential elastic of the system is given by:

$$772 \quad U_e = \sum_{i=1}^n U_{e_i} = \frac{1}{2} \sum_{i=1}^n \mathbf{u}_i^T \mathbf{K}_i \mathbf{u}_i = \frac{1}{2} \mathbf{u}_{tot}^T \mathbf{K}_{tot} \mathbf{u}_{tot} \quad (\text{C.1})$$

773 where

- 774 •  $\mathbf{u}_{tot} = [\mathbf{u}_1^T \dots \mathbf{u}_n^T]^T$  is the vector stacking all nodal displacements for all  $n$  robot bodies
- 775 •  $\mathbf{K}_{tot}$  is a block-diagonal matrix stacking on its diagonal all bodies stiffness matrices as follows:

$$776 \quad \mathbf{K}_{tot} = \begin{bmatrix} \mathbf{K}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_n \end{bmatrix} \quad (\text{C.2})$$

777 The robot stiffness matrix  $\mathbf{K}$  can be obtained by taking into account the fact the robot bodies are connected  
 778 altogether through the interface nodes. As a result, the expression of the vector  $\mathbf{u}_{tot}$  can be obtained from a  
 779 reduced set of independent coordinates  $\mathbf{u}$  as follows [34, 68]:

$$780 \quad \mathbf{u}_{tot} = \mathbf{J}(\mathbf{q})\mathbf{u} \quad (\text{C.3})$$

781 where  $\mathbf{J}(\mathbf{q})$  depends on the robot configuration  $\mathbf{q}$  but not on the nodal displacements  $\mathbf{u}$  (assumption of small  
 782 perturbations).

783 Introducing (C.3) into (C.1), we get

$$784 \quad U_e = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} \quad (\text{C.4})$$

785 where  $\mathbf{K} = \mathbf{J}^T(\mathbf{q})\mathbf{K}_{tot}\mathbf{J}(\mathbf{q})$  is the robot stiffness matrix which relates the nodal displacements  $\mathbf{u}$  to external  
 786 forces  $\mathbf{f}$  exerted on the nodes by the relation [68]:

$$787 \quad \mathbf{f} = \frac{\partial U_e}{\partial \mathbf{u}} = \mathbf{K} \mathbf{u} \quad (\text{C.5})$$

#### 788 Appendix D. Usual computation of the robot natural frequencies

789 Considering the robot made of  $n$  bodies, and by using the expression of the body kinetic energy (31) the  
 790 full kinetic elastic of the system is given by:

$$791 \quad T_e = \sum_{i=1}^n T_{e_i} = \frac{1}{2} \sum_{i=1}^n \dot{\mathbf{u}}_i^T \mathbf{M}_i \dot{\mathbf{u}}_i = \frac{1}{2} \dot{\mathbf{u}}_{tot}^T \mathbf{M}_{tot} \dot{\mathbf{u}}_{tot} \quad (\text{D.1})$$

792 where  $\mathbf{M}_{tot}$  is a block-diagonal matrix stacking on its diagonal all bodies mass matrices as follows:

$$793 \quad \mathbf{M}_{tot} = \begin{bmatrix} \mathbf{M}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_n \end{bmatrix} \quad (\text{D.2})$$

794 Differentiating (C.3) with respect to time, and keeping in mind that for the robot frequency analysis, the  
 795 configuration  $\mathbf{q}$  is nominal is considered as constant:

$$796 \quad \dot{\mathbf{u}}_{tot} = \mathbf{J}(\mathbf{q})\dot{\mathbf{u}} \quad (\text{D.3})$$

797 Introducing (D.3) into (D.1), we get

$$798 \quad T_e = \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{M} \dot{\mathbf{u}} \quad (\text{D.4})$$

799 where  $\mathbf{M} = \mathbf{J}^T(\mathbf{q})\mathbf{M}_{tot}\mathbf{J}(\mathbf{q})$  is the robot mass matrix.

800 Using the Lagrange equations, which state that, in absence of external efforts, we have

$$801 \quad \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{u}}} \right) - \frac{\partial L}{\partial \mathbf{u}} = \mathbf{0} \quad (\text{D.5})$$

802 where  $L = T_e - U_e$  in which the expression of  $U_e$  comes from (C.1), we get the dynamic equation characterizing  
 803 the robot free oscillations

$$804 \quad \mathbf{M} \ddot{\mathbf{u}} + \mathbf{K} \mathbf{u} = \mathbf{0} \quad (\text{D.6})$$

805 A solution  $\mathbf{u}_{tk}$  of this equation satisfies:

$$806 \quad (\omega_k^2 \mathbf{M} - \mathbf{K}) \mathbf{u}_{tk} = \mathbf{0} \quad (\text{D.7})$$

807 where  $\omega_k = 2\pi f_k$ ,  $f_k$  is the natural frequency associated with the  $k$ th natural mode of vibrations and  $\mathbf{u}_{tk}$  is its  
 808 associated eigenvector.

809 **Appendix E. Derivative of the elastostatic performance index with respect to the decision variables**

810 The derivative of the criterion (28) with respect to the decision variable  $\rho_{ij}$  is equal to:

$$811 \quad \frac{\partial c_1}{\partial \rho_{ij}} = 2\mathbf{u}_c^T \mathbf{e} \frac{\partial \mathbf{u}_r}{\partial \rho_{ij}} \quad (\text{E.1})$$

812 The term  $\frac{\partial \mathbf{u}_r}{\partial \rho_{ij}}$  can be obtained by the differentiation of the equation (27) with respect to the decision  
813 variable  $\rho_{ij}$ :

$$814 \quad \mathbf{0} = \frac{\partial}{\partial \rho_{ij}} (\mathbf{K}_r \mathbf{u}_r) = \frac{\partial \mathbf{K}_r}{\partial \rho_{ij}} \mathbf{u}_r + \mathbf{K}_r \frac{\partial \mathbf{u}_r}{\partial \rho_{ij}} \quad (\text{E.2})$$

815 leading thus to

$$816 \quad \frac{\partial \mathbf{u}_r}{\partial \rho_{ij}} = -\mathbf{K}_r^{-1} \frac{\partial \mathbf{K}_r}{\partial \rho_{ij}} \mathbf{u}_r \quad (\text{E.3})$$

817 The computation of the derivative of the matrix  $\mathbf{K}_r$  with respect to the decision variable is not computa-  
818 tionally efficient (problem of data storage due to the large number of decision variables), but, by analyzing the  
819 expression (E.3), we see that it is not necessary to compute  $\frac{\partial \mathbf{K}_r}{\partial \rho_{ij}}$  but the product  $\frac{\partial \mathbf{K}_r}{\partial \rho_{ij}} \mathbf{u}_r$  which can be obtained  
820 in a more efficient way. Indeed, by using (25), this product becomes equal to

$$821 \quad \frac{\partial \mathbf{K}_r}{\partial \rho_{ij}} \mathbf{u}_r = \mathbf{J}_r^T(\mathbf{q}) \begin{bmatrix} \mathbf{0} \\ \vdots \\ \frac{\partial \mathbf{K}_i^{red}}{\partial \rho_{ij}} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad \mathbf{J}_r(\mathbf{q}) \mathbf{u}_r = \mathbf{J}_r^T(\mathbf{q}) \begin{bmatrix} \mathbf{0} \\ \vdots \\ \frac{\partial \mathbf{K}_i^{red}}{\partial \rho_{ij}} \\ \vdots \\ \mathbf{0} \end{bmatrix} \mathbf{u}_{tot}^{red} \quad (\text{E.4})$$

822 or also

$$823 \quad \frac{\partial \mathbf{K}_r}{\partial \rho_{ij}} \mathbf{u}_r = \mathbf{J}_r^T(\mathbf{q}) \begin{bmatrix} \mathbf{0} \\ \vdots \\ \frac{\partial \mathbf{K}_i^{red}}{\partial \rho_{ij}} \mathbf{u}_{il} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (\text{E.5})$$

824 where  $\frac{\partial \mathbf{K}_i^{red}}{\partial \rho_{ij}}$  is the  $i$ th block term on the diagonal, all other terms being null.

825 Then, by differentiating the equation (21), it is possible to compute the term  $\frac{\partial \mathbf{K}_i^{red}}{\partial \rho_{ij}} \mathbf{u}_{il}$  in (E.5) as follows:

$$826 \quad \frac{\partial \mathbf{K}_i^{red}}{\partial \rho_{ij}} \mathbf{u}_{il} = \frac{\partial}{\partial \rho_{ij}} (\mathbf{K}_i^{(21)} \Phi_{si} + \mathbf{K}_i^{(22)}) \mathbf{u}_{il} = \frac{\partial \mathbf{K}_i^{(21)}}{\partial \rho_{ij}} \Phi_{si} \mathbf{u}_{il} + \mathbf{K}_i^{(21)} \frac{\partial \Phi_{si}}{\partial \rho_{ij}} \mathbf{u}_{il} + \frac{\partial \mathbf{K}_i^{(22)}}{\partial \rho_{ij}} \mathbf{u}_{il} \quad (\text{E.6})$$

827 The derivatives of  $\mathbf{K}_i^{(21)}$  and  $\mathbf{K}_i^{(22)}$ , which are nothing else than the derivatives of the terms of the matrix  $\mathbf{K}_i$   
828 given in (17), can be found from the differentiation of the matrix  $\mathbf{K}_i$  given by:

$$829 \quad \frac{\partial \mathbf{K}_i}{\partial \rho_{ij}} = p \rho_{ij}^{p-1} E_0 \mathbf{J}_i^T \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{K}_{ij}^{(0)} \\ \vdots \\ \mathbf{0} \end{bmatrix} \mathbf{J}_i \quad (\text{E.7})$$

830 where  $\mathbf{K}_{ij}^{(0)}$  is the  $j$ th block term on the diagonal, all other terms being null.

831 The other terms  $\mathbf{K}_i^{(21)} \frac{\partial \Phi_{si}}{\partial \rho_{ij}}$  in (E.6) can be found from the differentiation of the expression  $\mathbf{K}_i^{(11)} \Phi_{si} =$   
832  $-\mathbf{K}_i^{(12)}$  deduced from (20)

$$833 \quad \frac{\partial \mathbf{K}_i^{(11)}}{\partial \rho_{ij}} \Phi_{si} + \mathbf{K}_i^{(11)} \frac{\partial \Phi_{si}}{\partial \rho_{ij}} = -\frac{\partial \mathbf{K}_i^{(12)}}{\partial \rho_{ij}} \Rightarrow \frac{\partial \Phi_{si}}{\partial \rho_{ij}} = -(\mathbf{K}_i^{(11)})^{-1} \left( \frac{\partial \mathbf{K}_i^{(11)}}{\partial \rho_{ij}} \Phi_{si} + \frac{\partial \mathbf{K}_i^{(12)}}{\partial \rho_{ij}} \right) \quad (\text{E.8})$$

834 in which the derivatives of  $\mathbf{K}_i^{(11)}$  and  $\mathbf{K}_i^{(12)}$  can be found from the partitioning of the matrix  $\partial \mathbf{K}_i / \partial \rho_{ij}$  in (E.7),  
835 using the same partitioning as for the matrix  $\mathbf{K}_i$  of (17).

836 Finally, by multiplying this equation by  $\mathbf{K}_i^{(21)}$  (which is equal to  $\mathbf{K}_i^{(12)T}$ ) and by using the expression of  
837  $\Phi_{si}$  given in (20), we found

$$838 \quad \mathbf{K}_i^{(21)} \frac{\partial \Phi_{si}}{\partial \rho_{ij}} = \Phi_{si}^T \left( \frac{\partial \mathbf{K}_i^{(11)}}{\partial \rho_{ij}} \Phi_{si} + \frac{\partial \mathbf{K}_i^{(12)}}{\partial \rho_{ij}} \right) \quad (\text{E.9})$$

## 839 Appendix F. Derivative of the elastodynamic performance index with respect to the decision variables

840 From [84], we know that derivative of the criterion (52) with respect to the decision variable  $\rho_{ij}$  can be  
841 obtained as follows. First, let us consider the Rayleigh quotient which is given by:

$$842 \quad \omega_k^2 = \frac{\mathbf{u}_{dk}^T \mathbf{K} \mathbf{u}_{dk}}{\mathbf{u}_{dk}^T \mathbf{M} \mathbf{u}_{dk}} \quad (\text{F.1})$$

843 Expanding (F.1) for the first mode of vibration, we have

$$844 \quad \mathbf{u}_{d1}^T \mathbf{K}_d \mathbf{u}_{d1} - \omega_1^2 \mathbf{u}_{d1}^T \mathbf{M}_d \mathbf{u}_{d1} = 0 \quad (\text{F.2})$$

845 By introducing (12), (36) and (46) into (F.2), we know that

$$846 \quad \mathbf{u}_{all} = \mathbf{J}_d \mathbf{B} \mathbf{J}_{tot} \mathbf{u}_{d1} = \mathbf{J}_{all} \mathbf{u}_{d1}, \text{ where } \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{B}_n \end{bmatrix} \text{ and } \mathbf{J}_{tot} = \begin{bmatrix} \mathbf{J}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{J}_n \end{bmatrix} \quad (\text{F.3})$$

847 and that

$$848 \quad \mathbf{K}_d = \mathbf{J}_{all}^T \mathbf{K}_{all} \mathbf{J}_{all}, \quad \mathbf{M}_d = \mathbf{J}_{all}^T \mathbf{M}_{all} \mathbf{J}_{all} \quad (\text{F.4})$$

849 in which  $\mathbf{J}_{all} = \mathbf{J}_d \mathbf{B} \mathbf{J}_{tot}$  and  $\mathbf{u}_{all} = [\mathbf{u}_{1tot}^T \dots \mathbf{u}_{ntot}^T]^T$  groups all nodes variables ( $\mathbf{u}_{itot}$  being defined in (10)), and  
850 where

$$851 \quad \mathbf{K}_{all} = \begin{bmatrix} \mathbf{K}_{1tot} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_{ntot} \end{bmatrix} \text{ and } \mathbf{M}_{all} = \begin{bmatrix} \mathbf{M}_{1tot} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{ntot} \end{bmatrix} \quad (\text{F.5})$$

852 Thus, introducing all these expressions in (F.2), we get

$$853 \quad \mathbf{u}_{all}^T \mathbf{K}_{all} \mathbf{u}_{all} - \omega_1^2 \mathbf{u}_{all}^T \mathbf{M}_{all} \mathbf{u}_{all} = 0 \quad (\text{F.6})$$

854 Then, differentiating (F.6) with respect to  $\rho_{ij}$ , we get the derivative of  $\omega_1$  with respect to  $\rho_{ij}$  as:

$$855 \quad \frac{\partial \omega_1}{\partial \rho_{ij}} = \frac{1}{2\omega_1 \mathbf{u}_{all}^T \mathbf{M}_{all} \mathbf{u}_{all}} \left( \mathbf{u}_{all}^T \left( \frac{\partial \mathbf{K}_{all}}{\partial \rho_{ij}} \right) \mathbf{u}_{all} - \omega_1^2 \mathbf{u}_{all}^T \left( \frac{\partial \mathbf{M}_{all}}{\partial \rho_{ij}} \right) \mathbf{u}_{all} + 2 \frac{\partial \mathbf{u}_{all}^T}{\partial \rho_{ij}} (\mathbf{K}_{all} - \omega_1^2 \mathbf{M}_{all}) \mathbf{u}_{all} \right) \quad (\text{F.7})$$

856 where

$$857 \quad \frac{\partial \mathbf{K}_{all}}{\partial \rho_{ij}} = \begin{bmatrix} \mathbf{0} & & & & \\ & \ddots & & & \\ & & \frac{\partial \mathbf{K}_{itot}}{\partial \rho_{ij}} & & \\ & & & \ddots & \\ & & & & \mathbf{0} \end{bmatrix} \text{ and } \frac{\partial \mathbf{M}_{all}}{\partial \rho_{ij}} = \begin{bmatrix} \mathbf{0} & & & & \\ & \ddots & & & \\ & & \frac{\partial \mathbf{M}_{itot}}{\partial \rho_{ij}} & & \\ & & & \ddots & \\ & & & & \mathbf{0} \end{bmatrix} \quad (\text{F.8})$$

858 in which the expressions of  $\frac{\partial \mathbf{K}_{i\text{tot}}}{\partial \rho_{ij}}$  and  $\frac{\partial \mathbf{M}_{i\text{tot}}}{\partial \rho_{ij}}$  are given by

$$859 \quad \frac{\partial \mathbf{K}_{i\text{tot}}}{\partial \rho_{ij}} = p \rho_{ij}^{p-1} E_0 \begin{bmatrix} \mathbf{0} & & & \\ & \ddots & & \\ & & \mathbf{K}_{ij}^{(0)} & \\ & & & \ddots \\ & & & & \mathbf{0} \end{bmatrix} \quad (\text{F.9})$$

860 and

$$861 \quad \frac{\partial \mathbf{M}_{i\text{tot}}}{\partial \rho_{ij}} = \begin{bmatrix} \mathbf{0} & & & \\ & \ddots & & \\ & & \mathbf{M}_{ij}^{(0)} & \\ & & & \ddots \\ & & & & \mathbf{0} \end{bmatrix} \quad (\text{F.10})$$

862 where  $\mathbf{K}_{ij}^{(0)}$  and  $\mathbf{M}_{ij}^{(0)}$  are defined at (8) and (29), respectively, and they are the  $j$ th block terms on the diagonal,  
863 all other terms being null.

864 As, at the natural frequency  $f_1$ ,  $(\mathbf{K}_{all} - \omega_1^2 \mathbf{M}_{all}) \mathbf{u}_{all} = \mathbf{0}$ , and taking into account that  $f_1 = 2\pi\omega_1$ , we finally  
865 have:

$$866 \quad \frac{\partial f_1}{\partial \rho_{ij}} = \frac{1}{4\pi\omega_1 \mathbf{u}_{all}^T \mathbf{M}_d \mathbf{u}_{all}} \left( \mathbf{u}_{all}^T \left( \frac{\partial \mathbf{K}_{all}}{\partial \rho_{ij}} \right) \mathbf{u}_{all} - \omega_1^2 \mathbf{u}_{all}^T \left( \frac{\partial \mathbf{K}_{all}}{\partial \rho_{ij}} \right) \mathbf{u}_{all} \right) \quad (\text{F.11})$$

## 867 Acknowledgements

868 This work was supported by the French Région Pays de la Loire in the frame of the project RobEcolo  
869 (Convention No. 2015-10773).

870 The assistance on the MMA algorithm from Professor K. Svanberg is gratefully acknowledged.

## 871 References

- 872 [1] J. Merlet, Optimal design of robots, in: Robotics: Science and Systems (RSS 2005), Cambridge, Mas-  
873 sachusetts, 2005.
- 874 [2] M. French, Conceptual Design for Engineers, Springer, 1999.
- 875 [3] G. Gogu, Structural Synthesis of Parallel Robots, Springer, 2008.
- 876 [4] X. Kong, C. Gosselin, Type Synthesis of Parallel Mechanisms, Springer, 2007.
- 877 [5] S. Caro, W. A. Khan, D. Pasini, J. Angeles, The rule-based conceptual design of the architecture of serial  
878 Schönflies-motion generators, Mechanism and Machine Theory 45 (2) (2010) 251–260.
- 879 [6] P. Ben-Horin, M. Shoham, Singularity analysis of a class of parallel robots based on grassmanncayley  
880 algebra, Mechanism and Machine Theory 41 (8) (2006) 958–970.
- 881 [7] J. Merlet, Designing a parallel manipulator for a specific workspace, The International Journal of  
882 Robotics Research 16 (4) (1997) 545–556.
- 883 [8] J. Merlet, C. G. G. Mouly, Workspaces of planar parallel manipulators, Mechanism and Machine Theory  
884 33 (1-2) (1998) 7–20.
- 885 [9] S. Briot, I. Bonev, Accuracy analysis of 3-dof planar parallel robots, Mechanism and Machine Theory  
886 43 (4) (2008) 445–458.



- 887 [10] N. Binaud, P. Cardou, S. Caro, P. Wenger, The kinematic sensitivity of robotic manipulators to joint clear-  
888 ances, in: Proceedings of ASME Design Engineering Technical Conferences, Montreal, QC, Canada,  
889 2010.
- 890 [11] S. Venanzi, V. Parenti-Castelli, A new technique for clearance influence analysis in spatial mechanisms,  
891 Journal of Mechanical Design 127 (2005) 446–455.
- 892 [12] S. Caro, P. Wenger, F. Bennis, D. Chablat, Sensitivity analysis of the orthoglide: A three-dof translational  
893 parallel kinematic machine, ASME Journal of mechanical design 128 (2) (2006) 392–402.
- 894 [13] X.-J. Liu, J. Wang, G. Pritschow, Kinematics, singularity and workspace of planar 5R symmetrical  
895 parallel mechanisms, Mechanism and Machine Theory 44 (2) (2006) 145169.
- 896 [14] J. Merlet, Jacobian, manipulability, condition number, and accuracy of parallel robots, ASME Transac-  
897 tions Journal of Mechanical Design 128 (1) (2006) 199–206.
- 898 [15] C. Gosselin, J. Angeles, A global performance index for the kinematic optimization of robotic manipu-  
899 lators, ASME Journal of Mechanical Design 113 (3) (1991) 220–226.
- 900 [16] V. Arakelian, S. Briot, V. Glazunov, Increase of singularity-free zones in the workspace of parallel ma-  
901 nipulators using mechanisms of variable structure, Mechanism and Machine Theory 43 (9) (2008) 1129–  
902 1140.
- 903 [17] S. Balli, S. Chand, Transmission angle in mechanisms, Mechanism and Machine Theory 37 (2) (2002)  
904 175–195.
- 905 [18] C. Chen, J. Angeles, Generalized transmission index and transmission quality for spatial linkages, Mech-  
906 anism and Machine Theory 42 (9) (2007) 1225–1237.
- 907 [19] Y. Takeda, H. Funabachi, Motion transmissibility of in-parallel actuated manipulators, Transactions  
908 JSME series C 8 (4) (1995) 749–755.
- 909 [20] J. Hubert, J. Merlet, Static of parallel manipulators and closeness to singularity, Journal of Mechanisms  
910 and Robotics 1 (1).
- 911 [21] X. Chen, C. Chen, X.-J. Liu, Evaluation of force/torque transmission quality for parallel manipulators,  
912 ASME Journal of Mechanisms and Robotics 7 (4).
- 913 [22] C. Germain, S. Caro, S. Briot, P. Wenger, Optimal design of the IRSBot-2 based on an optimized test  
914 trajectory, in: Proceedings of the ASME 2013 International Design Engineering Technical Conferences  
915 & Computers and Information in Engineering Conference (IDETC/CIE 2013), Portland, Oregon, USA,  
916 2013.
- 917 [23] S. Briot, A. Pashkevich, D. Chablat, Technology-oriented optimization of the secondary design param-  
918 eters of robots for high-speed machining applications, in: Proceedings of the ASME 2010 International  
919 Design Engineering Technical Conferences and Computers and Information in Engineering Conference  
920 (IDETC/CIE 2010), Montreal, Quebec, Canada, 2010.
- 921 [24] R. Barents, M. Schenk, W. van Dorsser, B. Wisse, J. Herder, Spring-to-spring balancing as energy-free  
922 adjustment method in gravity equilibrators, ASME Journal of Mechanical Design 133 (6).
- 923 [25] H. Chaudhary, S. Saha, Dynamics and balancing of multibody systems, Springer, 2009.
- 924 [26] V. Arakelian, M. Smith, Shaking force and shaking moment balancing of mechanisms: a historical  
925 review with new examples, ASME Journal of Mechanical Design 127 (2005) 334–339.
- 926 [27] H. Asada, K. Youcef-Toumi, Decoupling of manipulator inertia tensor by mass redistribution, in: Pro-  
927 ceedings of the Design Engineering Technical Conference (DETC), Cambridge, 1984.

- 928 [28] T. Hess-Coelho, L. Yong, V. Alves, Decoupling of dynamic equations by means of adaptive balancing  
929 of 2-dof open-loop mechanisms, *Mechanism and Machine Theory* 39 (8) (2004) 871–881.
- 930 [29] S. Briot, A. Pashkevich, D. Chablat, On the optimal design of parallel robots taking into account their  
931 deformations and natural frequencies, in: *Proceedings of the ASME 2009 International Design Engi-  
932 neering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE,  
933 San Diego, California, 2009.*
- 934 [30] B. Bouzgarrou, J. Fauroux, G. Gogu, Y. Heerah, Rigidity analysis of t3r1 parallel robot uncoupled  
935 kinematics, in: *Proceedings of the 35th International Symposium on Robotics, Paris, France, 2004.*
- 936 [31] A. Cammarata, R. Sinatra, Elastodynamic optimization of a 3T1R parallel manipulator, *Mechanism and  
937 Machine Theory* 73 (2014) 184–196.
- 938 [32] S. Caro, F. Bennis, P. Wenger, Tolerance synthesis of mechanisms: a robust design approach, in: *Pro-  
939 ceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers  
940 and Information in Engineering Conference (IDETC 2003), Chicago, IL, 2003.*
- 941 [33] S. Briot, A. Pashkevich, D. Chablat, Optimal technology-oriented design of parallel robots for high-  
942 speed machining applications, in: *Proceedings of the 2010 IEEE International Conference on Robotics  
943 and Automation (ICRA 2010), Anchorage, Alaska, USA, 2010.*
- 944 [34] A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, 2005.
- 945 [35] O. Sigmund, K. Maute, Topology optimization approaches: A comparative review, *Structural and Mul-  
946 tidisciplinary Optimization* 48 (6) (2013) 1031–1055.
- 947 [36] Q. Chen, X. Zhang, The local optimum in topology optimization of compliant mechanisms, in: X. Zhang,  
948 N. Wang, Y. Huang (Eds.), *Lecture Notes in Electrical Engineering*, vol 408, Vol. 408, Springer, 2017.
- 949 [37] V. Venkiteswaran, O. Turkkan, H. Su, Compliant mechanism design through topology optimization us-  
950 ing pseudo-rigid-body models, in: *Proceedings of the ASME 2016 International Design Engineering  
951 Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2016),  
952 2016.*
- 953 [38] A. Saxena, G. Ananthasuresh, Topology optimization of compliant mechanisms with strength consider-  
954 ations 29 (2001) 199–221.
- 955 [39] O. Sigmund, A 99 line topology optimization code written in MATLAB, *Structural and Multidisciplinary  
956 Optimization* 21 (2) (2001) 120–127.
- 957 [40] M. Rossow, J. Taylor, A finite element method for the optimal design of variable thickness sheets, *AIAA  
958 Journal* 11 (11) (1973) 1566–1569.
- 959 [41] A. Albers, S. Brudniok, J. Ottndad, C. Sauter, K. Sedchaicham, Upper body of a new humanoid robot -  
960 the design of ARMAR III, in: *Proceedings of the 6th IEEE-RAS international conference on humanoid  
961 robots, 2006.*
- 962 [42] S. Lohmeier, T. Buschmann, M. Schwienbacher, H. Ulbrich, F. Pfeiffer, Leg design for a humanoid  
963 walking robot, in: *Proceedings of the 6th IEEE-RAS international conference on humanoid robots,  
964 2006, pp. 536–541.*
- 965 [43] S. Lohmeier, T. Buschmann, H. Ulbrich, Humanoid robot LOLA, in: *Proceedings of the 2009 IEEE  
966 International Conference on Robotics and Automation (ICRA 2009), 2009.*
- 967 [44] W. Kwon, H. Kim, J. Park, C. Roh, J. Lee, J. Park, W. Kim, K. Roh, Biped humanoid robot Mahru III,  
968 in: *Proceedings of the 7th IEEE-RAS international conference on humanoid robots, 2007, pp. 583–588.*

- 969 [45] B. Yunfei, C. Ming, L. Yongyao, Structural Topology Optimization for a Robot Upper Arm Based on  
970 SIMP Method, Springer International Publishing, Cham, 2016, pp. 725–733.
- 971 [46] H. Huang, G. Zhang, The topology optimization for l-shape arm of Motorman-HP20 robot, Applied  
972 Mechanics and Materials 201-202 (2012) 871–874.
- 973 [47] S. Oliveri, M. Cali, G. Sequenzia, G. Fatuzzo, Structural and topological optimization in robot design, in:  
974 Proceedings of the IMProVe 2011 International conference on Innovative Methods in Product Design,  
975 Venice, Italy, 2011.
- 976 [48] A. Albers, J. Ottnad, W. Häussler, Methods for lightweight design of mechanical components in hu-  
977 manoid robots, in: Proceedings of the 7th IEEE-RAS international conference on humanoid robots,  
978 2007, pp. 609–615.
- 979 [49] A. Albers, J. Ottnad, System based topology optimization as development tools for lightweight compo-  
980 nents in humanoid robots, in: Proceedings of the 8th IEEE-RAS international conference on humanoid  
981 robots, 2008, pp. 674–680.
- 982 [50] B. Kim, D. Yun, S. Lee, G. Jang, Topology optimization of industrial robots for system-level stiffness  
983 maximization by using part-level metamodels, Structural and Multidisciplinary Optimization (2016) 1–  
984 11.
- 985 [51] E. Hong, B. You, C. Kim, G. Park, Optimization of flexible components of multibody systems via  
986 equivalent static loads, Structural and Multidisciplinary Optimization 40 (1-6) (2010) 549–562.
- 987 [52] B. Pshenichnyj, The Linearization Method for Constrained Optimization, Computational Mathematics,  
988 Springer, 1994.
- 989 [53] K. Svanberg, The method of moving asymptotes – a new method for structural optimization, Numerical  
990 Methods in Engineering 24 (2) (1987) 359–373.
- 991 [54] L. Yin, W. Yang, Optimality criteria method for topology optimization under multiple constraints, Com-  
992 puters and Structures 79 (20-21) (2001) 1839–1850.
- 993 [55] C. Chang, A. Borgart, A. Chen, M. Hendriks, Direct gradient projection method with transformation of  
994 variables technique for structural topology optimization, Structural and Multidisciplinary Optimization  
995 49 (1) (2014) 107–119.
- 996 [56] C. Fleury, Conlin: an efficient dual optimizer based on convex approximation concepts, Structural Opti-  
997 mization 1 (2) (1989) 81–89.
- 998 [57] B. N. Pshenichnyi, Algorithms for general mathematical programming problems, Cybernetics 6 (5)  
999 (1970) 677–684.
- 1000 [58] B. N. Pshenichnyj, The Linearization Method for Constrained Optimization, Springer Series in Compu-  
1001 tational Mathematics, Springer, 1994.
- 1002 [59] B. T. Poljak, Introduction to optimization, Optimization Software, 1987.
- 1003 [60] P. L. Combettes, H. J. Trussell, Method of successive projections for finding a common point of sets in  
1004 metric spaces, Journal of Optimization Theory and Applications 67 (3) (1990) 487–507.
- 1005 [61] D. Bertsekas, Nonlinear Programming, Second Edition, Athena Scientific, 1999.
- 1006 [62] G. Pagis, N. Bouton, S. Briot, P. Martinet, Enlarging parallel robot workspace through type-2 singularity  
1007 crossing, Control Engineering Practice 39 (2015) 1–11.
- 1008 [63] M. Gautier, Dynamic identification of robots with power model, in: Proceedings IEEE ICRA, Albu-  
1009 querre, USA, 1997, pp. 1922–1927.

- 1010 [64] W. Khalil, E. Dombre, Modeling, Identification and Control of Robots., Hermes Penton London, 2002.
- 1011 [65] W. E. Singhose, N. C. Singer, W. P. Seering, Design and implementation of time-optimal negative input  
1012 shapers, International Mechanical Engineering Congress and Exposition.
- 1013 [66] L. Douat, I. Queinnec, G. Garcia, M. Michelin, F. Pierrot, Infinitely control applied to the vibration min-  
1014 imization of the parallel robot Par2, in: IEEE Multiconference on Systems and Control (MSC 2011),  
1015 2011.
- 1016 [67] M. Bendsoe, O. Sigmund, Material interpolation schemes in topology optimization, Archive of Applied  
1017 Mechanics 69.
- 1018 [68] J. Imbert, Analyse des structures par éléments finis, Editions Cépaduès, 1984.
- 1019 [69] R. R. Craig, M. C. C. Bampton, Coupling of substructures for dynamic analysis, AIAA Journal 6 (7).
- 1020 [70] A. Torii, A. Novotny, R. dos Santos, Robust compliance topology optimization based on the topological  
1021 derivative concept 106 (2016) 889–903.
- 1022 [71] J. W. Blankenship, J. E. Falk, Infinitely constrained optimization problems, Journal of Optimization  
1023 Theory and Applications 19 (2) (1976) 261–281.
- 1024 [72] J. Hollerbach, W. Khalil, M. Gautier, Handbook of Robotics - Model Identification, Springer, 2008.
- 1025 [73] F. Bourbonais, P. Bigras, I. Bonev, Minimum-time trajectory planning and control of a reconfigurable  
1026 pick-and-place parallel robot, IEEE/ASME Transactions on Mechatronics.
- 1027 [74] C. Gosselin, J. Angeles, Singularity analysis of closed-loop kinematic chains, IEEE Transactions on  
1028 Robotics and Automation 6 (3) (1990) 281–290.
- 1029 [75] I. Bonev, Geometric analysis of parallel mechanisms, Ph.D. thesis, Université Laval, QC, Canada (nov  
1030 2002).
- 1031 [76] C. Germain, S. Caro, S. Briot, P. Wenger, Optimal Design of the IRSBot-2 Based on an Optimized Test  
1032 Trajectory, in: Proc. of ASME Design Engineering Technical Conferences, Portland, OR, USA, 2013.
- 1033 [77] J. Merlet, Parallel Robots, 2nd Edition, Springer, 2006.
- 1034 [78] G. Hao, H. Li, S. Kemalcan, G. Chen, J. Yu, Understanding coupled factors that affect the modelling  
1035 accuracy of typical planar compliant mechanisms 11 (2016) 129–134.
- 1036 [79] B. Bourdin, Filters in topology optimization, International Journal for Numerical Methods in Engineer-  
1037 ing 50.
- 1038 [80] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine,  
1039 H. Van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods,  
1040 SIAM, 1994.
- 1041 [81] V. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned con-  
1042 jugate gradient method, SIAM Journal on Scientific Computing 23 (2).
- 1043 [82] M. Gautier, W. Khalil, Direct calculation of minimum set of inertial parameters of serial robots, Robotics  
1044 and Automation, IEEE Transactions on 6 (3) (1990) 368–373.
- 1045 [83] M. Gautier, Numerical calculation of the base inertial parameters, Journal of Robotics Systems 8 (4)  
1046 (1991) 485–506.
- 1047 [84] X. Huang, Z. Zuo, Y. Xie, Evolutionary topological optimization of vibrating continuum structures for  
1048 natural frequencies, Computers and Structures 88.