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

Sébastien Briot, Alexandre Goldsztejn

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

Sébastien Briot[a,*], Alexandre Goldsztejn[a]

[a]*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 leaded 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@irccyn.ec-nantes.fr (Sébastien Briot), Alexandre.Goldsztejn@irccyn.ec-nantes.fr (Alexandre Goldsztejn)

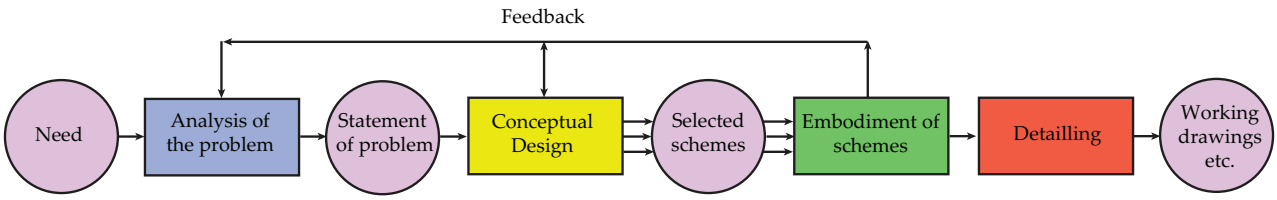


Figure 1: French design process [2]

- 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:

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

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 $\rho_1^2 + 1000\rho_2^2 \leq 1$[1]. 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 $\boldsymbol{\rho}_{k+1} = \boldsymbol{\rho}_k + \alpha_k \boldsymbol{d}_k$, where α_k is a step size and $\boldsymbol{d}_k$ 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 $\boldsymbol{u}_k = \boldsymbol{\rho}_k - s_k \nabla f(\boldsymbol{\rho}_k)$ is projected on the linearization of the constraint to obtain $\boldsymbol{v}_k$. Then, $\boldsymbol{d}_k$ is the vector $\boldsymbol{v}_k - \boldsymbol{\rho}_k$, 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 $\boldsymbol{d}_k$ 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:

$$\begin{aligned} \boldsymbol{v}_k &= \mathrm{proj}_{L_k}(\boldsymbol{\rho}_k - s_k \nabla f(\boldsymbol{\rho}_k)) && (2) \\ \boldsymbol{\rho}_{k+1} &= \boldsymbol{\rho}_k + \alpha_k\,(\boldsymbol{v}_k - \boldsymbol{\rho}_k). && (3) \end{aligned}$$

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 $\mathrm{proj}_{L_k}(\boldsymbol{u})$ projects the vector $\boldsymbol{u}$ 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:

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

[1] 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.

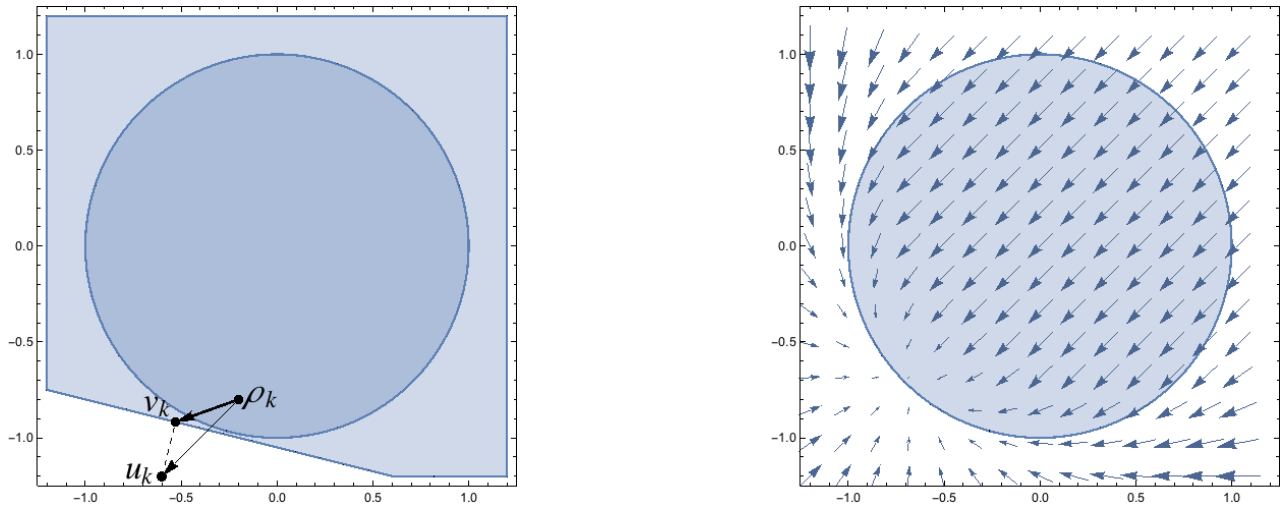


Figure 2: Left: Projected gradient. Right: Linearization method (vectors are scaled for the sake of lisibility).

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

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

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

3. Performance indices

3.1. Selection of the performance indices

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

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

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

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

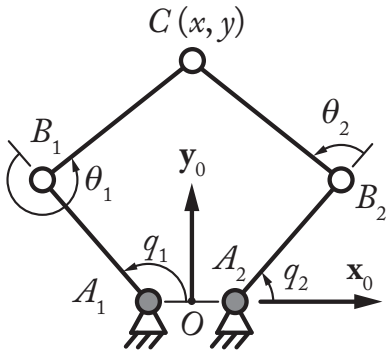


Figure 3: A five-bar mechanism (the gray pairs denote the actuated joints).

robot design). However, the robot mass is not necessarily the only inertial parameter to be taken into account. For instance, as shown in [62], the most influential term in the dynamic model of the five-bar 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 of inertia around z_0 of the link between the points A_i and B_i computed at A_i, $\ell_{A_iB_i}$ is the distance between 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 is more likely to minimize the robot input torques than minimizing the robot mass.

- *robot input efforts or energy consumption:* the cost of an actuator is related to its power or to its continuous or peak torques. We know that, for a usual robot, more than 90 % of the dynamic effects come from the "rigid behavior" of the mechanical architecture (remaining 10 % being due to elasticity, impacts due to clearance in joints, and unmodelled friction effects) [63]. Therefore, criteria based on the input efforts, energy or power consumption can be computed with a very good accuracy neglecting the elastic behavior of the links.

- *robot static deformations:* static deformations under external (potentially variable) loading are classical performance indices when designing a mechanical system.

- *robot natural frequencies:* the first natural frequencies are associated with the highest level of energy due to vibrations, and the first modes represent the displacements with the highest amplitude. Moreover, the first natural frequencies are also used in order to set the bandwidth of controllers: it is usually said that the cutoff frequency for a robot controller must be set around the half of the first natural frequency of the mechanical architecture [64]. If not, the controller can become unstable due to unmodeled dynamics. We believe that natural frequencies are more important indices than the amplitude of the robot dynamic deformations used [48, 49, 50, 51]: The lack of accuracy due to oscillation can be compensated thanks to advanced controller (e.g. [65, 66]) while low robot natural frequencies cannot be increased through control.

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

Below, we prefer to stress the issue of the calculation of the robot deformations and natural frequencies, whose cost of computation is huge, and which must be computed for numerous robot configurations. As mentioned in introduction, in order to considerably decrease the computation cost of these performance indices, we exploit the structure of the elastic models (which have the highest computational costs) in order to reduce their computational complexity. Indeed, we show below 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.

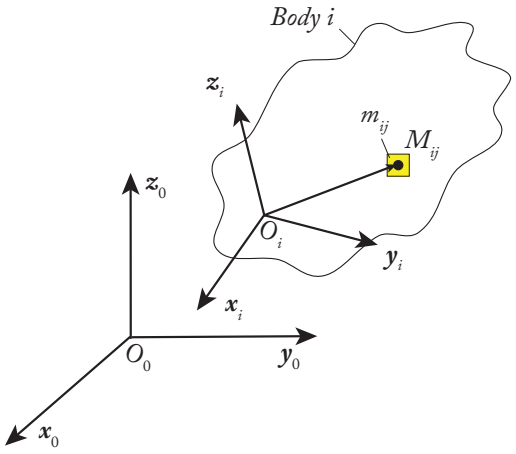


Figure 4: A body in space meshed with finite elements.

3.2. Static deformations

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

3.2.1. Material interpolation scheme and stiffness matrix of a link

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

The SIMP scheme is defined as follows:

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

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

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

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

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

$$U_{e_{ij}} = \frac{1}{2}\mathbf{u}_{ij}^T \mathbf{K}_{ij}\mathbf{u}_{ij} \tag{9}$$

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

$$U_{e_i} = \sum_{j=1}^{m} U_{e_{ij}} = \frac{1}{2}\sum_{j=1}^{m} \mathbf{u}_{ij}^T \mathbf{K}_{ij}\mathbf{u}_{ij} = \frac{1}{2}\mathbf{u}_{i\,tot}^T \mathbf{K}_{i\,tot}\mathbf{u}_{i\,tot} \tag{10}$$

where

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

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

$$\mathbf{K}_{i\,tot} = \begin{bmatrix} \mathbf{K}_{i1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_{im} \end{bmatrix} \tag{11}$$

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

$$\mathbf{u}_{i\,tot} = \mathbf{J}_i \mathbf{u}_i \tag{12}$$

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

$$U_{e_i} = \frac{1}{2} \mathbf{u}_i^T \mathbf{K}_i \mathbf{u}_i \tag{13}$$

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

$$\mathbf{f}_i = \frac{\partial U_{e_i}}{\partial \mathbf{u}_i} = \mathbf{K}_i \mathbf{u}_i \tag{14}$$

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

$$\mathbf{u}_i = \begin{bmatrix} \mathbf{u}_{i\,f} \\ \mathbf{u}_{i\,l} \end{bmatrix} \tag{15}$$

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

3.2.2. Efficient computation of the elastostatic model

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

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

In order to perform the reduction of the model size, we use the partitioning of the vector $\mathbf{u}_i$ shown in (15). 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 $\mathbf{u}_{i\,l}$, we also partition the vector $\mathbf{f}_i$ in (14) such that

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

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

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

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.

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

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

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

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:

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

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

Introducing (20) into (19), we get

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

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

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

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

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

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

where

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

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

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

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

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 perturbations).

Introducing (25) into (23), we get

$$U_e = \frac{1}{2}\mathbf{u}_r^T\mathbf{K}_r\mathbf{u}_r \tag{26}$$

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 forces $\mathbf{f}_r$ exerted on the considered nodes by the relation:

$$\mathbf{f}_r = \frac{\partial U_e}{\partial \mathbf{u}_r} = \mathbf{K}_r \mathbf{u}_r \tag{27}$$

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 ($(12n \times 12n)$ for 3D problems). Therefore, even if the problems (C.5) and (27) will give exactly the same results of computation, the resolution of the equation (27) is much more efficient than the resolution of the equation (C.5) due to the considerably reduced size of the problem.

3.2.3. Stiffness performance index

Usually, it is not necessary to constrain all deformations $\mathbf{u}_r$, but a set of them. Let us define a row vector $\mathbf{e}$ 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 this vector $\mathbf{u}_c$ or its square (in order to avoid the use of square roots in the definition of the performance index) given by:

$$c_1 = \mathbf{u}_c^T \mathbf{u}_c = \mathbf{u}_r^T \mathbf{e}^T \mathbf{e}\, \mathbf{u}_r \tag{28}$$

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

3.3. Natural frequencies

3.3.1. Mass matrix of a link

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

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

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

$$T_{e_{ij}} = \frac{1}{2}\dot{\mathbf{u}}_{ij}^T \mathbf{M}_{ij} \dot{\mathbf{u}}_{ij} \tag{30}$$

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 to

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

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

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

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

$$\dot{\mathbf{u}}_{i\,tot} = \mathbf{J}_i \dot{\mathbf{u}}_i \tag{33}$$

Introducing (33) into (31), we get

$$T_{e_i} = \frac{1}{2}\dot{\mathbf{u}}_i^T \mathbf{M}_i \dot{\mathbf{u}}_i \tag{34}$$

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:

$$\mathbf{u}_{if} = \mathbf{\Phi}_{si}\mathbf{u}_{il} + \mathbf{\Phi}_{di}\mathbf{q}_i \tag{35}$$

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

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

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

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

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 $\mathbf{\Phi}_{di}$, s vectors are chosen among the a eigenmodes associated with (37). Usually, $s << a$ (typically, in our examples, s is lower than 10).

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

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

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

$$\dot{\mathbf{u}}_i = \mathbf{B}_i\,\dot{\mathbf{u}}_{ir} \tag{39}$$

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

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

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

where $\mathbf{K}_{ir} = \mathbf{B}_i^T\mathbf{K}_{itot}\mathbf{B}_i$ and $\mathbf{M}_{ir} = \mathbf{B}_i^T\mathbf{M}_{itot}\mathbf{B}_i$ are the reduced body stiffness and mass matrices. These matrices are usually of small dimension (typically, in our 2D problems, of dimension (12×12), while in 3D it is of dimension (18×18)) 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 $\mathbf{\Phi}_{di}$ in (36). However, for two bodies i and k, the computation of the matrices $\mathbf{\Phi}_{di}$ and $\mathbf{\Phi}_{dk}$ is independent. Thus the computation can be made in parallel on multi-core computers in order to save computational time.

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

$$U_e = \sum_{i=1}^{n} U_{e_i} = \frac{1}{2} \sum_{i=1}^{n} \mathbf{u}_{i\,r}^T \mathbf{K}_i^{red} \mathbf{u}_{i\,r} = \frac{1}{2} \mathbf{u}_{tot}^{dyn\,T} \mathbf{K}_{tot}^{dyn} \mathbf{u}_{tot}^{dyn} \tag{42}$$

$$T_e = \sum_{i=1}^{n} T_{e_i} = \frac{1}{2} \sum_{i=1}^{n} \dot{\mathbf{u}}_{i\,r}^T \mathbf{M}_i^{red} \dot{\mathbf{u}}_{i\,r} = \frac{1}{2} \dot{\mathbf{u}}_{tot}^{dyn\,T} \mathbf{M}_{tot}^{dyn} \dot{\mathbf{u}}_{tot}^{dyn} \tag{43}$$

where

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

$$\mathbf{K}_{tot}^{dyn} = \begin{bmatrix} \mathbf{K}_{1\,r} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_{n\,r} \end{bmatrix}, \quad \mathbf{M}_{tot}^{dyn} = \begin{bmatrix} \mathbf{M}_{1\,r} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{n\,r} \end{bmatrix} \tag{44}$$

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

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

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 perturbations). Moreover, for the analysis of the oscillatory free behavior, the configuration $\mathbf{q}$ is considered constant, thus

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

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

$$U_e = \frac{1}{2} \mathbf{u}_d^T \mathbf{K}_d \mathbf{u}_d \tag{47}$$

$$T_e = \frac{1}{2} \dot{\mathbf{u}}_d^T \mathbf{M}_d \dot{\mathbf{u}}_d \tag{48}$$

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.

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

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

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

$$\mathbf{M}_d \ddot{\mathbf{u}}_d + \mathbf{K}_d \mathbf{u}_d = \mathbf{0} \tag{50}$$

A solution $\mathbf{u}_{d\,k}$ of this equation satisfies:

$$\left(\omega_k^2 \mathbf{M}_d - \mathbf{K}_d \right) \mathbf{u}_{d\,k} = \mathbf{0} \tag{51}$$

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

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 robot bodies ($(18n \times 18n)$ for 3D problems). Therefore, even if the problems (F.1) and (51) will give exactly the same results of computation for the first frequencies, the resolution of the equation (51) is much more efficient than the resolution of the equation (F.1) due to the considerably reduced size of the problem.

3.3.3. Vibration performance index

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

$$c_2 = f_1 \tag{52}$$

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

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

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

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

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

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

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

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

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 position, velocity and acceleration, respectively, and the matrix $\boldsymbol{\Gamma}$ is the Jacobian matrix of $\boldsymbol{\tau}$ with respect to $\hat{\boldsymbol{\rho}}$.

Let us now denote as $\mathcal{T}$ a trajectory described by N samples of time, leading to N samples of the joint 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 instant t_k. Considering all samples, we rewrite (53) as

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

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 performance (such as the maximal velocity and motion ranges), the "exciting" trajectory will be found as the solution of the following optimization problem:

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

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

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

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

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

- the initial and final robot configurations for each portion of the trajectory $\mathcal{T}$

- plus the duration of the motion profile corresponding to each portion of the trajectory $\mathcal{T}$.

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

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

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

$$\begin{array}{ll} \max\limits_{\mathbf{q}} & c_1(\hat{\boldsymbol{\rho}}, \mathbf{f}) \\ \text{under} & \mathbf{q} \in [\mathbf{q}] \text{ and } \mathbf{x} \in [\mathbf{x}] \end{array} \tag{56}$$

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

- in the case of the natural frequencies:

$$\begin{array}{ll} \min\limits_{\mathbf{q}} & c_2(\hat{\boldsymbol{\rho}}) \\ \text{under} & \mathbf{q} \in [\mathbf{q}] \text{ and } \mathbf{x} \in [\mathbf{x}] \end{array} \tag{57}$$

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

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

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

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

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

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

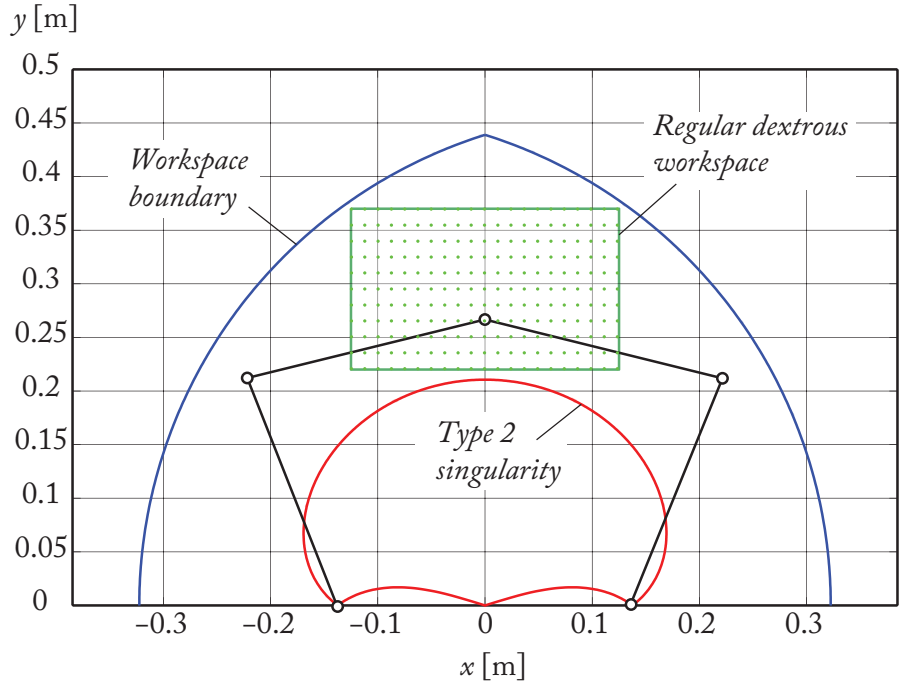


Figure 5: The five-bar mechanism under consideration and its workspace (to scale).

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

5.1. Definition of the optimal design problem in 2D

In this section, we propose as a case study the topology optimization of a five-bar mechanism (Fig. 3) which is a robot composed of five revolute joints, two of them being active (at points A_1 and A_2) while the other are passive. This mechanism with two dof is able to position in a plane its end-effector located at point C with coordinates $(x\ y)$. The bodies between points A_i and B_i are called the proximal links and the bodies 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 located at A_i, B_i and C.

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

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

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

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

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

Finally, we decide to solve the following optimization problem:

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

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

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

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

The design optimization problem can thus be formalized as

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

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

5.2. Results and discussion for the 2D model

5.2.1. Results

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

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

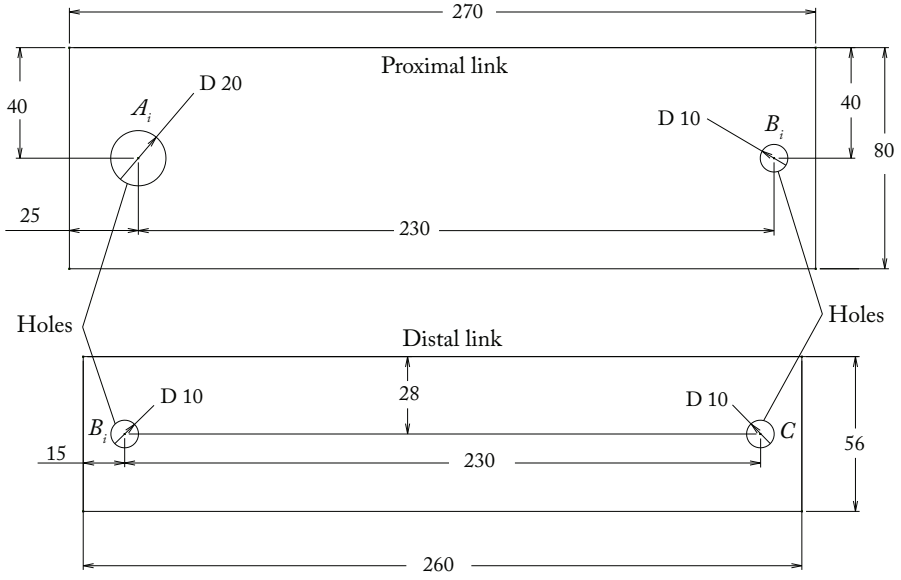


Figure 6: Initial design domain for the design of the robot link (to scale). All dimensions are in millimeters.

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

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

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

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

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

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

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

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

16 GB of RAM.

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

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

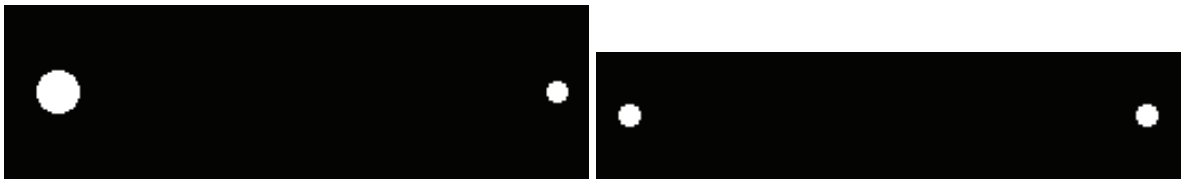
In order to show the importance of the update of the value of $\mathbf{q}_2^\theta$ between Round 1 and Round 2, we optimized the robot for constraints calculated for the configurations and trajectories selected at the initial round only (row corresponding to round 1 in Tabs. 1 and 2 for the configurations,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

5.2.2. Discussions

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

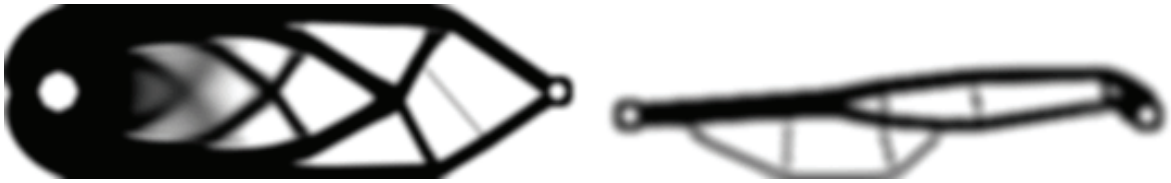
The evolution of the torque RMS as a function of the computational time in the last optimization round is shown in Fig. 13. It can be seen than both functions reach the same values of the objective in almost the same time, even if MMA goes a bit lower (but in a more longer time) than LM. Thus, LM is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

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



(a) Design of the proximal links: initial design domain

(b) Design of the distal links: initial design domain



(c) Design of the proximal links: end of round 1

(d) Design of the distal links: end of round 1

(e) Design of the proximal links: end of round 2

(f) Design of the distal links: end of round 2

(g) Design of the proximal links: end of the final round

(h) Design of the distal links: end of the final round

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 $0 < \rho_{ij} < 1$)

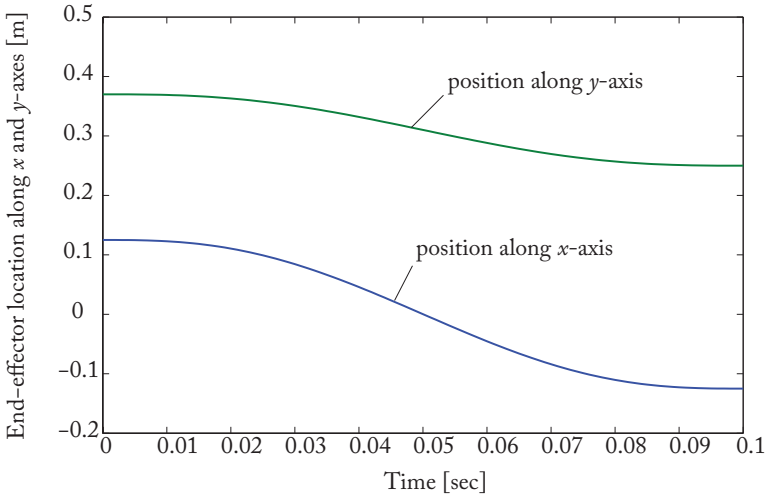


Figure 8: Trajectory used for the computation of the input torques.

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

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

5.3. Solving the problem in 3D

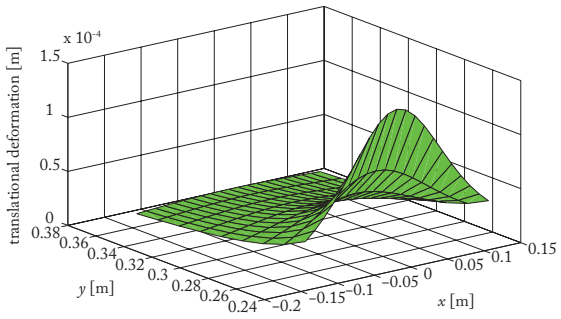
The optimization problem can also be solved in 3D, even if in such a case the computational time is increase. In Fig. 15, we show the proximal and distal links of a five-bar mechanism which have been obtained when solving the following optimization problem:

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

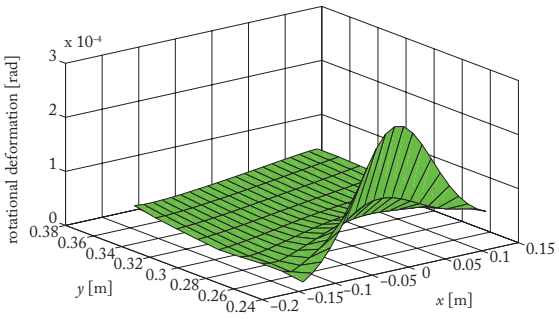
The design optimization problem can thus be formalized as

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

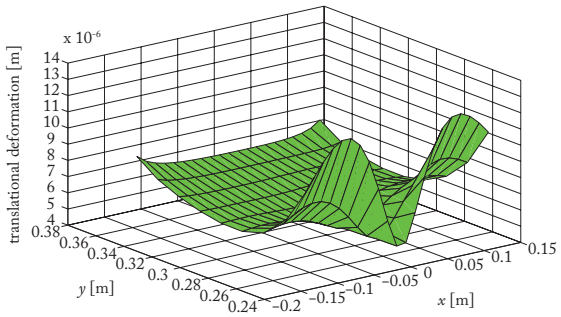
Here, we chose as link lengths the design parameters of the DexTAR robot [73] from the company Mecademic. The link lengths are thus: $\ell_{OA_1} = \ell_{OA_2} = 0.056$ m, $\ell_{A_1B_1} = \ell_{A_2B_2} = 0.09$ m and $\ell_{B_1C} = \ell_{B_2C} = 0.09$ m



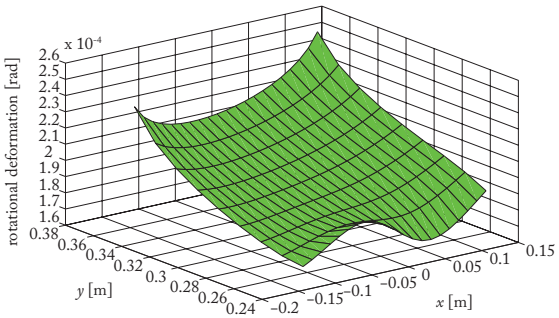
(a) Norm of the translational deformations under the application of the loading $\mathbf{f}_1$



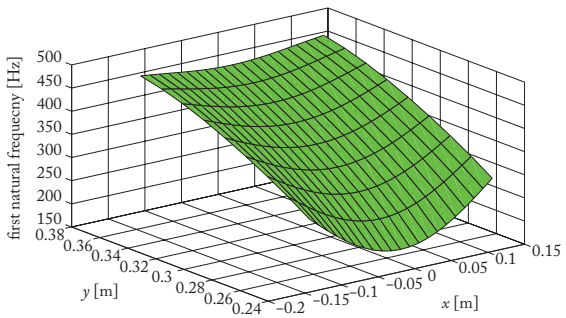
(b) Absolute value of the rotational deformations under the application of the loading $\mathbf{f}_1$



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

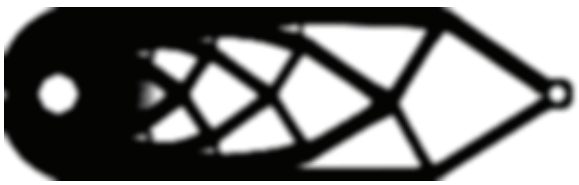


(d) Absolute value of the rotational deformations under the application of the loading $\mathbf{f}_2$



(e) First natural frequency

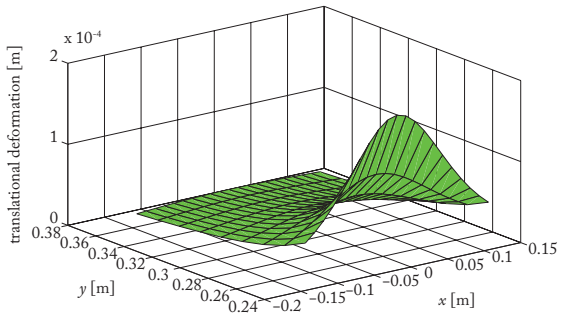
Figure 9: Evolution of the deformations and of the first natural frequency in the workspace for the optimized design



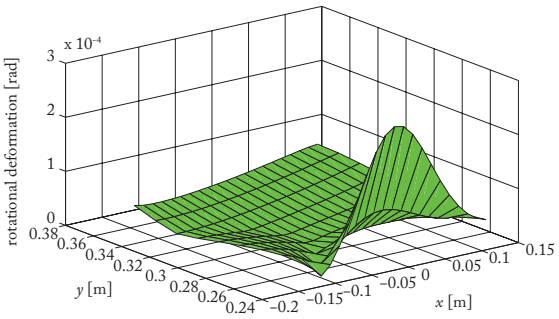
(a) Design of the proximal links

(b) Design of the distal links

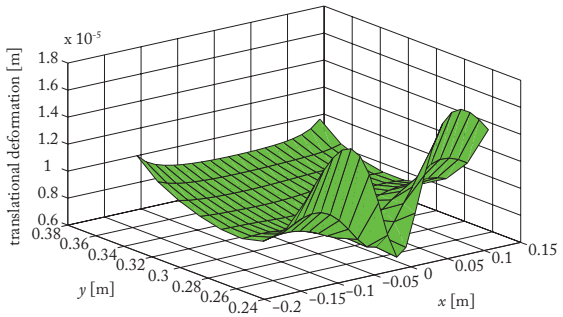
Figure 10: Optimal designs of the five-bar links using the configurations and trajectories selected at the initial round for the computation of the objective and constraints: the links are shown in gray-scale (black elements correspond to $\rho_{ij} = 1$, white elements to $\rho_{ij} = 0$, and gray elements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



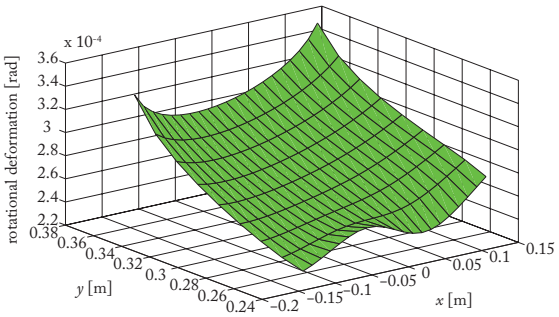
(a) Norm of the translational deformations under the application of the loading $\mathbf{f}_1$



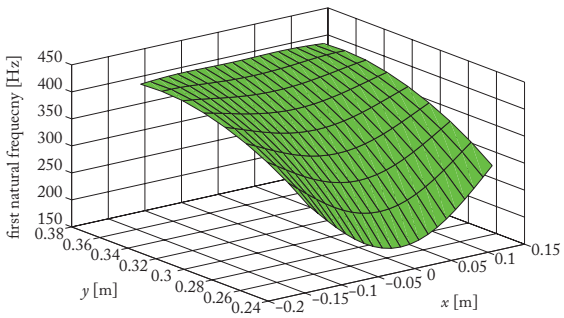
(b) Absolute value of the rotational deformations under the application of the loading $\mathbf{f}_1$



(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 $0 < \rho_{ij} < 1$)

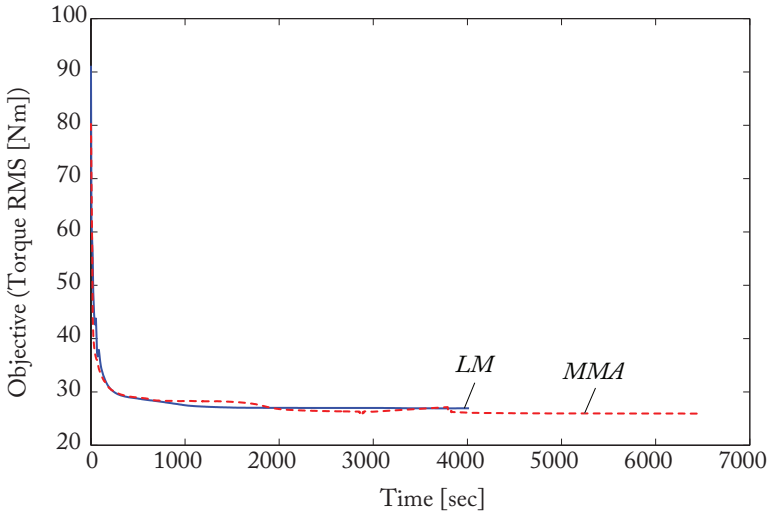


Figure 13: Torque RMS as a function of the computational time in the last optimization round for LM (full line) and MMA (dotted line) procedures.

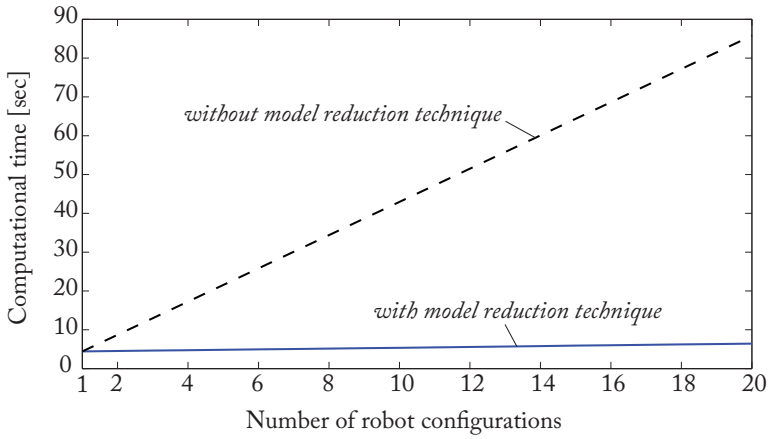


Figure 14: Evolution of computational cost for the computation of the first natural frequency as a function of the number of tested configurations.

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

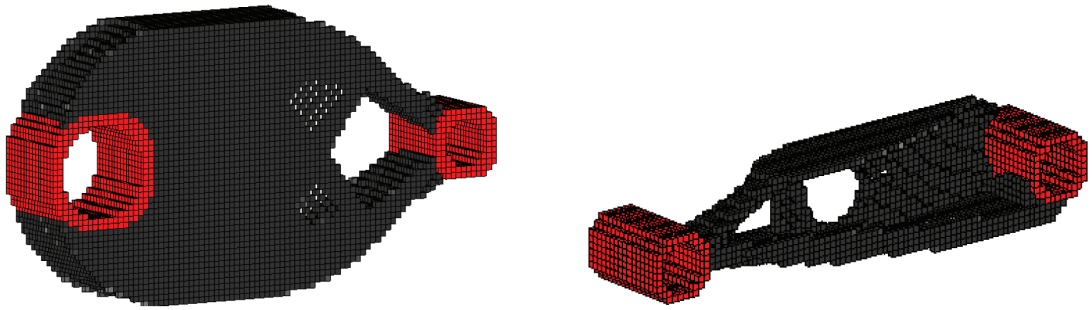
The initial design domain for the proximal and distal links is represented in Fig. 16 (planar view, the initial width of both links is of 20 mm). For the meshing of the links, HEXA8 finite elements (i.e. eight-nodes cubic planar elements) of size $1.3 \times 1.3 \times 1.3$ mm are used. Links are considered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

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

In order to accelerate the computation of the elastostatic properties by avoiding the inversion of the matrices $\mathbf{K}_i^{(11)}$ in (20) for both links, we computed the matrices $\boldsymbol{\Phi}_{si}$ using a "pcg" (Preconditioned Conjugate Gradient) algorithm [80]. Indeed, we computed each column of the matrix $\boldsymbol{\Phi}_{si}$ by solving the system of equations with the "pcg" algorithm:

$$\mathbf{K}_i^{(11)} \boldsymbol{\Phi}_{sij} = -\mathbf{K}_{ij}^{(12)} \tag{60}$$

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



(a) Design of the proximal links (b) Design of the distal links

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

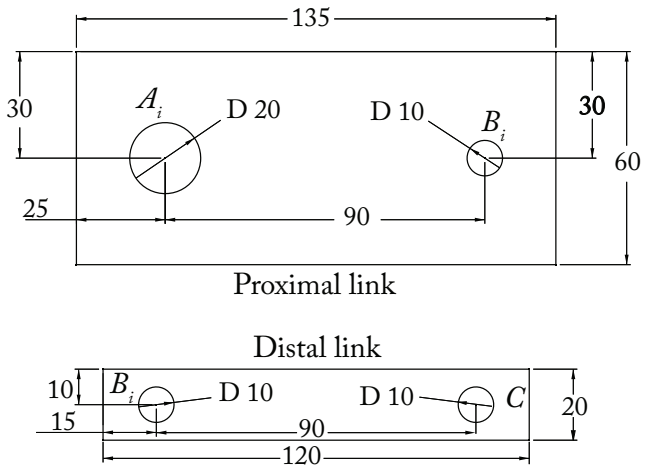


Figure 16: Initial design domain for the design of the robot link (planar view, to scale). The initial width of both links is of 20 mm. All dimensions are in millimeters.

doing a computation in parallel of each column of the matrix $\mathbf{\Phi}_{si}$ (indeed, we used 6 CPUs in parallel on our computer to do so).

We also introduced in the optimization problem some constraints on the limits of the first natural frequency. The results obtained were similar to the ones shown in Fig. 15, but with much longer computational time (around 6 days and 8 hours). This considerable increase of the computational cost was due to the calculation of the matrices $\mathbf{\Phi}_{di}$ in (36) for both links which is obtained by solving an eigenvalue problem of large dimension (matrices of size (256581×256581) for the proximal links). Furthermore, it should be mentioned that this eigenvalue problem cannot be solved in Matlab with the classical "eig" or "eigs" functions which requests too much memory allocation (our computer went out of memory), but with a "lobpcg" (Locally Optimal Block Preconditioned Conjugate Gradient) algorithm [81] which is known to be efficient for computing eigenvalues problems of very large dimensions.

6. Conclusion

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

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

was 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 exploited the structure of the elastic models in order to reduce their computational complexity. Indeed, we showed that it was 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.

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

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

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

Appendix A. Inertial parameters

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

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

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

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

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

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

finally have the following expression:

$$\boldsymbol{\chi}_i = \boldsymbol{\Psi}_i \boldsymbol{\rho}_i, \text{ where } \boldsymbol{\Psi}_i = m_{ij} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_{i1} & x_{i2} & \dots & x_{im} \\ y_{i1} & y_{i2} & \dots & y_{im} \\ z_{i1} & z_{i2} & \dots & z_{im} \\ y_{i1}^2 + z_{i1}^2 & y_{i2}^2 + z_{i2}^2 & \dots & y_{im}^2 + z_{im}^2 \\ x_{i1}^2 + z_{i1}^2 & x_{i2}^2 + z_{i2}^2 & \dots & x_{im}^2 + z_{im}^2 \\ x_{i1}^2 + y_{i1}^2 & x_{i2}^2 + y_{i2}^2 & \dots & x_{im}^2 + y_{im}^2 \\ x_{i1}y_{i1} & x_{i2}y_{i2} & \dots & x_{im}y_{im} \\ x_{i1}z_{i1} & x_{i2}z_{i2} & \dots & x_{im}z_{im} \\ y_{i1}z_{i1} & y_{i2}z_{i2} & \dots & y_{im}z_{im} \end{bmatrix} \text{ and } \boldsymbol{\rho}_i = \begin{bmatrix} \rho_{i1} \\ \rho_{i2} \\ \vdots \\ \rho_{im} \end{bmatrix} \tag{A.1}$$

in which m_{ij} is a constant mass as we consider here that the body is meshed with equal size elements. The matrix $\boldsymbol{\Psi}_i$ is the Jacobian of $\boldsymbol{\chi}_i$ with respect to $\boldsymbol{\rho}_i$.

Grouping all inertial terms for all bodies in the vector $\boldsymbol{\chi} = [\boldsymbol{\chi}_1^T \dots \boldsymbol{\chi}_n^T]^T$ (n being the number of robot bodies), we finally get the expression:

$$\boldsymbol{\chi} = \boldsymbol{\Psi} \boldsymbol{\rho}, \text{ where } \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \boldsymbol{\Psi}_n \end{bmatrix} \text{ and } \boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{\rho}_1 \\ \vdots \\ \boldsymbol{\rho}_n \end{bmatrix} \tag{A.2}$$

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

$$c_3 = \mathbf{a}\boldsymbol{\chi} = \mathbf{a}\boldsymbol{\Psi}\boldsymbol{\rho} \tag{A.3}$$

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

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

Appendix B. Input efforts and energy consumption

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

$$\boldsymbol{\tau} = \boldsymbol{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\chi} \tag{B.1}$$

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

$$H = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\chi} \tag{B.3}$$

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

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

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

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

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

- the infinite norm $\|\tau_i\|_\infty$ of the input effort for the motor i (τ_i being the ith component of the vector $\boldsymbol{\tau}$) along the trajectory $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ that should be lower than the actuator peak effort specification,

- the root-mean-square $\overline{\tau}_i$ of the input effort for the motor i along the trajectory $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ than should be lower than the actuator continuous effort specification.

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 N samples for $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. We denote as $\mathbf{y}_i$ the vector containing all values of τ_i computed along the sampled trajectory $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$: the kth component of $\mathbf{y}_i$ is equal to $\tau_i^{(k)} = \boldsymbol{\Gamma}_i(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k)\,\boldsymbol{\rho}$, where $(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k)$ is the value of $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ at the sample k and $\boldsymbol{\Gamma}_i$ is the ith row of $\boldsymbol{\Gamma}$. As a result, $\mathbf{y}_i$ takes the form:

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

By definition, we have

$$\|\tau_i\|_\infty = \max \sqrt{\mathbf{y}_i * \mathbf{y}_i} \tag{B.8}$$

in which the operator "$*$" represents the term-by-term product and

$$\overline{\tau}_i = \sqrt{\frac{\mathbf{y}_i^T \mathbf{y}_i}{N}} \tag{B.9}$$

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

$$c_4 = \|\tau_i\|_\infty^2 = \max(\mathbf{y}_i * \mathbf{y}_i) = \max\left(\mathbf{W}_i\,\boldsymbol{\rho} * \mathbf{W}_i\,\boldsymbol{\rho}\right) \tag{B.10}$$

$$c_5 = \overline{\tau}_i^2 = \frac{1}{N}\,\mathbf{y}_i^T \mathbf{y}_i = \frac{1}{N}\,\boldsymbol{\rho}^T \mathbf{W}_i^T\,\mathbf{W}_i\,\boldsymbol{\rho} \tag{B.11}$$

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

Appendix C. Usual computation of the robot elastostatic model

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

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

where

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

$$\mathbf{K}_{tot} = \begin{bmatrix} \mathbf{K}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_n \end{bmatrix} \tag{C.2}$$

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

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

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

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

$$U_e = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} \tag{C.4}$$

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 forces $\mathbf{f}$ exerted on the nodes by the relation [68]:

$$\mathbf{f} = \frac{\partial U_e}{\partial \mathbf{u}} = \mathbf{K}\mathbf{u} \tag{C.5}$$

Appendix D. Usual computation of the robot natural frequencies

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

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

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

$$\mathbf{M}_{tot} = \begin{bmatrix} \mathbf{M}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_n \end{bmatrix} \tag{D.2}$$

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

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

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

$$T_e = \frac{1}{2}\dot{\mathbf{u}}^T\mathbf{M}\dot{\mathbf{u}} \tag{D.4}$$

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

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

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

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

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{0} \tag{D.6}$$

A solution $\mathbf{u}_{t\,k}$ of this equation satisfies:

$$\left(\omega_k^2\mathbf{M} - \mathbf{K}\right)\mathbf{u}_{t\,k} = \mathbf{0} \tag{D.7}$$

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

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

The derivative of the criterion (28) with respect to the decision variable ρ_{ij} is equal to:

$$\frac{\partial c_1}{\partial \rho_{ij}} = 2\mathbf{u}_c^T \mathbf{e} \frac{\partial \mathbf{u}_r}{\partial \rho_{ij}} \tag{E.1}$$

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 variable ρ_{ij}:

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

leading thus to

$$\frac{\partial \mathbf{u}_r}{\partial \rho_{ij}} = -\mathbf{K}_r^{-1} \frac{\partial \mathbf{K}_r}{\partial \rho_{ij}} \mathbf{u}_r \tag{E.3}$$

The computation of the derivative of the matrix $\mathbf{K}_r$ with respect to the decision variable is not computationally efficient (problem of data storage due to the large number of decision variables), but, by analyzing the 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 in a more efficient way. Indeed, by using (25), this product becomes equal to

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

or also

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

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

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:

$$\frac{\partial \mathbf{K}_i^{red}}{\partial \rho_{ij}} \mathbf{u}_{il} = \frac{\partial}{\partial \rho_{ij}} \left(\mathbf{K}_i^{(21)} \mathbf{\Phi}_{si} + \mathbf{K}_i^{(22)} \right) \mathbf{u}_{il} = \frac{\partial \mathbf{K}_i^{(21)}}{\partial \rho_{ij}} \mathbf{\Phi}_{si} \mathbf{u}_{il} + \mathbf{K}_i^{(21)} \frac{\partial \mathbf{\Phi}_{si}}{\partial \rho_{ij}} \mathbf{u}_{il} + \frac{\partial \mathbf{K}_i^{(22)}}{\partial \rho_{ij}} \mathbf{u}_{il} \tag{E.6}$$

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 $\mathbf{K}_i$ given in (17), can be found from the differentiation of the matrix $\mathbf{K}_i$ given by:

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

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

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

$$\frac{\partial \mathbf{K}_i^{(11)}}{\partial \rho_{ij}}\mathbf{\Phi}_{s\,i} + \mathbf{K}_i^{(11)}\frac{\partial \mathbf{\Phi}_{s\,i}}{\partial \rho_{ij}} = -\frac{\partial \mathbf{K}_i^{(12)}}{\partial \rho_{ij}} \Rightarrow \frac{\partial \mathbf{\Phi}_{s\,i}}{\partial \rho_{ij}} = -\left(\mathbf{K}_i^{(11)}\right)^{-1}\left(\frac{\partial \mathbf{K}_i^{(11)}}{\partial \rho_{ij}}\mathbf{\Phi}_{s\,i} + \frac{\partial \mathbf{K}_i^{(12)}}{\partial \rho_{ij}}\right) \tag{E.8}$$

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), using the same partitioning as for the matrix $\mathbf{K}_i$ of (17).

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 $\mathbf{\Phi}_{s\,i}$ given in (20), we found

$$\mathbf{K}_i^{(21)}\frac{\partial \mathbf{\Phi}_{s\,i}}{\partial \rho_{ij}} = \mathbf{\Phi}_{s\,i}^T\left(\frac{\partial \mathbf{K}_i^{(11)}}{\partial \rho_{ij}}\mathbf{\Phi}_{s\,i} + \frac{\partial \mathbf{K}_i^{(12)}}{\partial \rho_{ij}}\right) \tag{E.9}$$

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

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

$$\omega_k^2 = \frac{\mathbf{u}_{d\,k}^T\mathbf{K}\mathbf{u}_{d\,k}}{\mathbf{u}_{d\,k}^T\mathbf{M}\mathbf{u}_{d\,k}} \tag{F.1}$$

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

$$\mathbf{u}_{d\,1}^T\mathbf{K}_d\mathbf{u}_{d\,1} - \omega_1^2\mathbf{u}_{d\,1}^T\mathbf{M}_d\mathbf{u}_{d\,1} = 0 \tag{F.2}$$

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

$$\mathbf{u}_{all} = \mathbf{J}_d\mathbf{B}\mathbf{J}_{tot}\mathbf{u}_{d\,1} = \mathbf{J}_{all}\mathbf{u}_{d\,1},\ \text{where}\ \mathbf{B} = \begin{bmatrix}\mathbf{B}_1 & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & \mathbf{B}_n\end{bmatrix}\ \text{and}\ \mathbf{J}_{tot} = \begin{bmatrix}\mathbf{J}_1 & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & \mathbf{J}_n\end{bmatrix} \tag{F.3}$$

and that

$$\mathbf{K}_d = \mathbf{J}_{all}^T\mathbf{K}_{all}\mathbf{J}_{all},\ \ \mathbf{M}_d = \mathbf{J}_{all}^T\mathbf{M}_{all}\mathbf{J}_{all} \tag{F.4}$$

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

$$\mathbf{K}_{all} = \begin{bmatrix}\mathbf{K}_{1\,tot} & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & \mathbf{K}_{n\,tot}\end{bmatrix}\ \text{and}\ \mathbf{M}_{all} = \begin{bmatrix}\mathbf{M}_{1\,tot} & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{n\,tot}\end{bmatrix} \tag{F.5}$$

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

$$\mathbf{u}_{all}^T\mathbf{K}_{all}\mathbf{u}_{all} - \omega_1^2\mathbf{u}_{all}^T\mathbf{M}_{all}\mathbf{u}_{all} = 0 \tag{F.6}$$

Then, differentiating (F.6) with respect to ρ_{ij}, we get the derivative of ω_1 with respect to ρ_{ij} as:

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

where

$$\frac{\partial \mathbf{K}_{all}}{\partial \rho_{ij}} = \begin{bmatrix}\mathbf{0} & & & \\ & \ddots & & \\ & & \frac{\partial \mathbf{K}_{i\,tot}}{\partial \rho_{ij}} & \\ & & & \ddots & \\ & & & & \mathbf{0}\end{bmatrix}\ \text{and}\ \frac{\partial \mathbf{M}_{all}}{\partial \rho_{ij}} = \begin{bmatrix}\mathbf{0} & & & \\ & \ddots & & \\ & & \frac{\partial \mathbf{M}_{i\,tot}}{\partial \rho_{ij}} & \\ & & & \ddots & \\ & & & & \mathbf{0}\end{bmatrix} \tag{F.8}$$

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

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

and

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

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

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

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

Acknowledgements

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

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

References

[1] J. Merlet, Optimal design of robots, in: Robotics: Science and Systems (RSS 2005), Cambridge, Massachusetts, 2005.

[2] M. French, Conceptual Design for Engineers, Springer, 1999.

[3] G. Gogu, Structural Synthesis of Parallel Robots, Springer, 2008.

[4] X. Kong, C. Gosselin, Type Synthesis of Parallel Mechanisms, Springer, 2007.

[5] S. Caro, W. A. Khan, D. Pasini, J. Angeles, The rule-based conceptual design of the architecture of serial Schönflies-motion generators, Mechanism and Machine Theory 45 (2) (2010) 251–260.

[6] P. Ben-Horin, M. Shoham, Singularity analysis of a class of parallel robots based on grassmanncayley algebra, Mechanism and Machine Theory 41 (8) (2006) 958–970.

[7] J. Merlet, Designing a parallel manipulator for a specific workspace, The International Journal of Robotics Research 16 (4) (1997) 545–556.

[8] J. Merlet, C. G. G. Mouly, Workspaces of planar parallel manipulators, Mechanism and Machine Theory 33 (1-2) (1998) 7–20.

[9] S. Briot, I. Bonev, Accuracy analysis of 3-dof planar parallel robots, Mechanism and Machine Theory 43 (4) (2008) 445–458.

[10] N. Binaud, P. Cardou, S. Caro, P. Wenger, The kinematic sensitivity of robotic manipulators to joint clearances, in: Proceedings of ASME Design Engineering Technical Conferences, Montreal, QC, Canada, 2010.

[11] S. Venanzi, V. Parenti-Castelli, A new technique for clearance influence analysis in spatial mechanisms, Journal of Mechanical Design 127 (2005) 446–455.

[12] S. Caro, P. Wenger, F. Bennis, D. Chablat, Sensitivity analysis of the orthoglide: A three-dof translational parallel kinematic machine, ASME Journal of mechanical design 128 (2) (2006) 392–402.

[13] X.-J. Liu, J. Wang, G. Pritschow, Kinematics, singularity and workspace of planar 5R symmetrical parallel mechanisms, Mechanism and Machine Theory 44 (2) (2006) 145169.

[14] J. Merlet, Jacobian, manipulability, condition number, and accuracy of parallel robots, ASME Transactions Journal of Mechanical Design 128 (1) (2006) 199–206.

[15] C. Gosselin, J. Angeles, A global performance index for the kinematic optimization of robotic manipulators, ASME Journal of Mechanical Design 113 (3) (1991) 220–226.

[16] V. Arakelian, S. Briot, V. Glazunov, Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure, Mechanism and Machine Theory 43 (9) (2008) 1129–1140.

[17] S. Balli, S. Chand, Transmission angle in mechanisms, Mechanism and Machine Theory 37 (2) (2002) 175–195.

[18] C. Chen, J. Angeles, Generalized transmission index and transmission quality for spatial linkages, Mechanism and Machine Theory 42 (9) (2007) 1225–1237.

[19] Y. Takeda, H. Funabachi, Motion transmissibility of in-parallel actuated manipulators, Transactions JSME series C 8 (4) (1995) 749–755.

[20] J. Hubert, J. Merlet, Static of parallel manipulators and closeness to singularity, Journal of Mechanisms and Robotics 1 (1).

[21] X. Chen, C. Chen, X.-J. Liu, Evaluation of force/torque transmission quality for parallel manipulators, ASME Journal of Mechanisms and Robotics 7 (4).

[22] C. Germain, S. Caro, S. Briot, P. Wenger, Optimal design of the IRSBot-2 based on an optimized test trajectory, in: Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2013), Portland, Oregon, USA, 2013.

[23] S. Briot, A. Pashkevich, D. Chablat, Technology-oriented optimization of the secondary design parameters of robots for high-speed machining applications, in: Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2010), Montreal, Quebec, Canada, 2010.

[24] R. Barents, M. Schenk, W. van Dorsser, B. Wisse, J. Herder, Spring-to-spring balancing as energy-free adjustment method in gravity equilibrators, ASME Journal of Mechanical Design 133 (6).

[25] H. Chaudhary, S. Saha, Dynamics and balancing of multibody systems, Springer, 2009.

[26] V. Arakelian, M. Smith, Shaking force and shaking moment balancing of mechanisms: a historical review with new examples, ASME Journal of Mechanical Design 127 (2005) 334–339.

[27] H. Asada, K. Youcef-Toumi, Decoupling of manipulator inertia tensor by mass redistribution, in: Proceedings of the Design Engineering Technical Conference (DETC), Cambridge, 1984.

[28] T. Hess-Coelho, L. Yong, V. Alves, Decoupling of dynamic equations by means of adaptive balancing of 2-dof open-loop mechanisms, Mechanism and Machine Theory 39 (8) (2004) 871–881.

[29] S. Briot, A. Pashkevich, D. Chablat, On the optimal design of parallel robots taking into account their deformations and natural frequencies, in: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE, San Diego, California, 2009.

[30] B. Bouzgarrou, J. Fauroux, G. Gogu, Y. Heerah, Rigidity analysis of t3r1 parallel robot uncoupled kinematics, in: Proceedings of the 35th International Symposium on Robotics, Paris, France, 2004.

[31] A. Cammarata, R. Sinatra, Elastodynamic optimization of a 3T1R parallel manipulator, Mechanism and Machine Theory 73 (2014) 184–196.

[32] S. Caro, F. Bennis, P. Wenger, Tolerance synthesis of mechanisms: a robust design approach, in: Proceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC 2003), Chicago, IL, 2003.

[33] S. Briot, A. Pashkevich, D. Chablat, Optimal technology-oriented design of parallel robots for high-speed machining applications, in: Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska, USA, 2010.

[34] A. Shabana, Dynamics of Multibody Systems, Cambridge University Press, 2005.

[35] O. Sigmund, K. Maute, Topology optimization approaches: A comparative review, Structural and Multidisciplinary Optimization 48 (6) (2013) 1031–1055.

[36] Q. Chen, X. Zhang, The local optimum in topology optimization of compliant mechanisms, in: X. Zhang, N. Wang, Y. Huang (Eds.), Lecture Notes in Electrical Engineering, vol 408, Vol. 408, Springer, 2017.

[37] V. Venkiteswaran, O. Turkkan, H. Su, Compliant mechanism design through topology optimization using pseudo-rigid-body models, in: Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2016), 2016.

[38] A. Saxena, G. Ananthasuresh, Topology optimization of compliant mechanisms with strength considerations 29 (2001) 199–221.

[39] O. Sigmund, A 99 line topology optimization code written in MATLAB, Structural and Multidisciplinary Optimization 21 (2) (2001) 120–127.

[40] M. Rossow, J. Taylor, A finite element method for the optimal design of variable thickness sheets, AIAA Journal 11 (11) (1973) 1566–1569.

[41] A. Albers, S. Brudniok, J. Ottnad, C. Sauter, K. Sedchaicham, Upper body of a new humanoid robot - the design of ARMAR III, in: Proceedings of the 6th IEEE-RAS international conference on humanoid robots, 2006.

[42] S. Lohmeier, T. Buschmann, M. Schwienbacher, H. Ulbrich, F. Pfeiffer, Leg design for a humanoid walking robot, in: Proceedings of the 6th IEEE-RAS international conference on humanoid robots, 2006, pp. 536–541.

[43] S. Lohmeier, T. Buschmann, H. Ulbrich, Humanoid robot LOLA, in: Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), 2009.

[44] W. Kwon, H. Kim, J. Park, C. Roh, J. Lee, J. Park, W. Kim, K. Roh, Biped humanoid robot Mahru III, in: Proceedings of the 7th IEEE-RAS international conference on humanoid robots, 2007, pp. 583–588.

[45] B. Yunfei, C. Ming, L. Yongyao, Structural Topology Optimization for a Robot Upper Arm Based on SIMP Method, Springer International Publishing, Cham, 2016, pp. 725–733.

[46] H. Huang, G. Zhang, The topology optimization for l-shape arm of Motorman-HP20 robot, Applied Mechanics and Materials 201-202 (2012) 871–874.

[47] S. Oliveri, M. Cali, G. Sequenzia, G. Fatuzzo, Structural and topological optimization in robot design, in: Proceedings of the IMProVe 2011 International conference on Innovative Methods in Product Design, Venice, Italy, 2011.

[48] A. Albers, J. Ottnad, W. Häussler, Methods for lightweight design of mechanical components in humanoid robots, in: Proceedings of the 7th IEEE-RAS international conference on humanoid robots, 2007, pp. 609–615.

[49] A. Albers, J. Ottnad, System based topology optimization as development tools for lightweight components in humanoid robots, in: Proceedings of the 8th IEEE-RAS international conference on humanoid robots, 2008, pp. 674–680.

[50] B. Kim, D. Yun, S. Lee, G. Jang, Topology optimization of industrial robots for system-level stiffness maximization by using part-level metamodels, Structural and Multidisciplinary Optimization (2016) 1–11.

[51] E. Hong, B. You, C. Kim, G. Park, Optimization of flexible components of multibody systems via equivalent static loads, Structural and Multidisciplinary Optimization 40 (1-6) (2010) 549–562.

[52] B. Pshenichnyj, The Linearization Method for Constrained Optimization, Computational Mathematics, Springer, 1994.

[53] K. Svanberg, The method of moving asymptotes – a new method for structural optimization, Numerical Methods in Engineering 24 (2) (1987) 359–373.

[54] L. Yin, W. Yang, Optimality criteria method for topology optimization under multiple constraints, Computers and Structures 79 (20-21) (2001) 1839–1850.

[55] C. Chang, A. Borgart, A. Chen, M. Hendriks, Direct gradient projection method with transformation of variables technique for structural topology optimization, Structural and Multidisciplinary Optimization 49 (1) (2014) 107–119.

[56] C. Fleury, Conlin: an efficient dual optimizer based on convex approximation concepts, Structural Optimization 1 (2) (1989) 81–89.

[57] B. N. Pshenichnyi, Algorithms for general mathematical programming problems, Cybernetics 6 (5) (1970) 677–684.

[58] B. N. Pshenichnyj, The Linearization Method for Constrained Optimization, Springer Series in Computational Mathematics, Springer, 1994.

[59] B. T. Poljak, Introduction to optimization, Optimization Software, 1987.

[60] P. L. Combettes, H. J. Trussell, Method of successive projections for finding a common point of sets in metric spaces, Journal of Optimization Theory and Applications 67 (3) (1990) 487–507.

[61] D. Bertsekas, Nonlinear Programming, Second Edition, Athena Scientific, 1999.

[62] G. Pagis, N. Bouton, S. Briot, P. Martinet, Enlarging parallel robot workspace through type-2 singularity crossing, Control Engineering Practice 39 (2015) 1–11.

[63] M. Gautier, Dynamic identification of robots with power model, in: Proceedings IEEE ICRA, Albuquerque, USA, 1997, pp. 1922–1927.

[64] W. Khalil, E. Dombre, Modeling, Identification and Control of Robots., Hermes Penton London, 2002.

[65] W. E. Singhose, N. C. Singer, W. P. Seering, Design and implementation of time-optimal negative input shapers, International Mechanical Engineering Congress and Exposition.

[66] L. Douat, I. Queinnec, G. Garcia, M. Michelin, F. Pierrot, Hinfiny control applied to the vibration minimization of the parallel robot Par2, in: IEEE Multiconference on Systems and Control (MSC 2011), 2011.

[67] M. Bendsoe, O. Sigmund, Material interpolation schemes in topology optimization, Archive of Applied Mechanics 69.

[68] J. Imbert, Analyse des structures par lments finis, Editions Cépaduès, 1984.

[69] R. R. Craig, M. C. C. Bampton, Coupling of substructures for dynamic analysis, AIAA Journal 6 (7).

[70] A. Torii, A. Novotny, R. dos Santos, Robust compliance topology optimization based on the topological derivative concept 106 (2016) 889–903.

[71] J. W. Blankenship, J. E. Falk, Infinitely constrained optimization problems, Journal of Optimization Theory and Applications 19 (2) (1976) 261–281.

[72] J. Hollerbach, W. Khalil, M. Gautier, Handbook of Robotics - Model Identification, Springer, 2008.

[73] F. Bourbonais, P. Bigras, I. Bonev, Minimum-time trajectory planning and control of a reconfigurable pick-and-place parallel robot, IEEE/ASME Transactions on Mechatronics.

[74] C. Gosselin, J. Angeles, Singularity analysis of closed-loop kinematic chains, IEEE Transactions on Robotics and Automation 6 (3) (1990) 281–290.

[75] I. Bonev, Geometric analysis of parallel mechanisms, Ph.D. thesis, Université Laval, QC, Canada (nov 2002).

[76] C. Germain, S. Caro, S. Briot, P. Wenger, Optimal Design of the IRSBot-2 Based on an Optimized Test Trajectory, in: Proc. of ASME Design Engineering Technical Conferences, Portland, OR, USA, 2013.

[77] J. Merlet, Parallel Robots, 2nd Edition, Springer, 2006.

[78] G. Hao, H. Li, S. Kemalcan, G. Chen, J. Yu, Understanding coupled factors that affect the modelling accuracy of typical planar compliant mechanisms 11 (2016) 129–134.

[79] B. Bourdin, Filters in topology optimization, International Journal for Numerical Methods in Engineering 50.

[80] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vors, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, 1994.

[81] V. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method, SIAM Journal on Scientific Computing 23 (2).

[82] M. Gautier, W. Khalil, Direct calculation of minimum set of inertial parameters of serial robots, Robotics and Automation, IEEE Transactions on 6 (3) (1990) 368–373.

[83] M. Gautier, Numerical calculation of the base inertial parameters, Journal of Robotics Systems 8 (4) (1991) 485–506.

[84] X. Huang, Z. Zuo, Y. Xie, Evolutionary topological optimization of vibrating continuum structures for natural frequencies, Computers and Structures 88.