



Shape optimization and sensitivity of compliant beams for prescribed load-displacement response

Giuseppe Radaelli^{1,2} and Just L. Herder¹

¹Dept. Precision and Microsystems Engineering, Delft University of Technology,
2628 CD Delft, the Netherlands

²Laevo BV, 2628 CA Delft, the Netherlands

Correspondence to: Giuseppe Radaelli (g.radaelli@tudelft.nl)

Received: 15 July 2016 – Revised: 17 October 2016 – Accepted: 24 October 2016 – Published: 17 November 2016

Abstract. This paper presents the shape optimization of a compliant beam for prescribed load-displacements response. The analysis of the design is based on the isogeometric analysis framework for an enhanced fidelity between designed and analysed shape. The sensitivities used for an improved optimization procedure are derived analytically, including terms due to the use of nonlinear state equations and nonlinear boundary constraint equations. A design example is illustrated where a beam shape is found that statically balances a pendulum over a range of 180° with good balancing quality. The analytical sensitivities are verified by comparison with finite difference sensitivities.

1 Introduction

Mechanism synthesis can be thought of as finding a mechanism with a certain force transmission, a certain motion transmission, or both. For conventional mechanisms the analysis and synthesis of motion and forces can be performed separately. This does not hold in general for compliant mechanisms. In compliant mechanisms, which move due to deformation of slender segments (Howell, 2001), every motion is associated to a restoring force. If a certain force and motion combination is desired at a part of the mechanism which is input and output at the same time, such mechanism is sometimes called a spring (Vehar-Jutte, 2008). Not limited to the conventional coil springs, where a motion over a straight path produces a linear force characteristic, a general spring mechanism can potentially exhibit infinite types of nonlinear load-displacement responses when moved along a general trajectory, which may be non-straight. A load-displacement response, in this context, is defined as the force/moment exerted by the spring given a series of applied boundary displacements/rotations.

Applications of non-linear springs can be found in many design disciplines including prosthetics, assistive devices, MEMS and user products. Often nonlinear springs are applied as balancing mechanisms where either an external load,

e.g. a weight, or an intrinsic stiffness, e.g. in a compliant mechanism, is counteracted by such nonlinear spring (Powell and Frecker, 2005; Chen and Zhang, 2011; Hoetmer et al., 2010). Types of non-linearity that are typically interesting are constant force mechanism, bi-stable or multi-stable mechanisms and negative stiffness mechanisms (Pucheta and Cardona, 2010; Oh, 2008; Tolou, 2012).

A way of obtaining nonlinear spring behaviour is by optimization of the shape of a chosen topology of elements such as rods, beams, shells etc. Other means are to manipulate the topology of a system (Sigmund, 1997; Du and Chen, 2008) or the material properties.

In tailoring the load-response of structures and, more in general, when dealing with large deflections the non-linearity of the equilibrium equations makes optimization more challenging. The optimization procedure, which is an iterative scheme, includes at every step the solution of a nonlinear set of equations, on their turn also iterative. Clear disadvantages are the complexity of the procedure and increased computation time, but also the smoothness of the optimization function space is often compromised due to e.g. singularities and bifurcations in the solution.

Eriksson (2014) proposes a method where tracing the non-linear equilibrium at every optimisation iteration is not needed. This is done by augmenting the system of equilib-

rium equations such that the unknowns include the displacements and design variables, including a load parameter. By keeping the load parameter constant a sequence of responses to that loading can be obtained for different designs, without computing the whole nonlinear equilibrium every time.

Similarly, in the concept of simultaneous analysis and design (SAND) (Haftka, 1985) the analysis unknowns, e.g. the displacements, and the design variables are all treated equivalently as optimization variables. Since equilibrium is not required at every iteration step, also tracing the nonlinear equilibrium path is not needed every time. At the end of the procedure equilibrium is hopefully satisfied, and the design optimized (Ringertz, 1989).

Also in the compliant mechanisms community several approaches have been proposed to deal with geometrical nonlinearities, often present in compliant mechanisms (Bruns and Tortorelli, 2001; Joo et al., 2001; Du and Chen, 2008).

In the majority of the cases the goal is to optimize the design for the situation where the full load is applied. If, however, the nonlinear load-response itself is what must be optimized, the problem gets more involved. Examples of shape and/or topology optimization where the whole load-response or part of it is optimized can be found in Saxena and Ananthasuresh (1999), Pedersen et al. (2005), Rai et al. (2006), Jutte and Kota (2010) and Leishman and Colton (2011).

In this work we focus on shape optimization and, as such, a simple given topology is assumed. Topology and material optimization are not considered in this work. In the context of shape optimization of structures there is an increasing interest in the isogeometric analysis (IGA) paradigm (Cottrell et al., 2009). This can be considered as an alternative to finite element analysis (FEA) with some peculiar additional advantages. There is an enhanced fidelity between designed shape and analysed shape. This comes from the use of B-splines as basis functions for the computer aided geometric design (CAGD) as well as for the structural analysis. This preserves the original shapes and guarantees a high level of continuity between elements (Hughes et al., 2005).

Having the same geometrical formulation at the basis of design and analysis also gives the advantage that there is no need to spend much effort in meshing, which is done repeatedly in an optimization procedure. Instead there are some well-performing and efficient refinement algorithms that are applied in order to work with a finer discretisation of the shape for the analysis.

A third advantage is that sensitivity properties are derivable in an analytical fashion, which supposedly can improve the efficiency of an optimization procedure (Cho and Ha, 2009; Nagy et al., 2010). Also, the derivable shape sensitivities are more accurate (Koo et al., 2013), leading to preciser results.

While the work done on isogeometric shape optimization is widespread (Nagy, 2011; Wall et al., 2008), specific attention to nonlinear settings is scarce (Kiendl, 2011; Koo et al., 2013). Moreover, while typical problems where the stiffness,

the weight, the volume or stresses are optimized have been analysed extensively (Nagy et al., 2010; Hsu, 1994; Ding, 1986), there is no work known by the authors where the load-displacement of a non-linear spring is optimized within the IGA framework.

The rotation-less character of the degrees of freedom in the used isogeometric formulation gives a complication with respect to the application of rotation constraints on the beam. These constraints, imposed here by nonlinear equations on the control points by Lagrange multipliers, have a relevant impact on the derivation of the sensitivity. Together with the nonlinearity of the equilibrium equations and the unusual type of objective function, this leads to a few non-trivial problems to be dealt with in this paper.

In previous work (Radaelli and Herder, 2014) the authors have applied isogeometric shape optimization to obtain a flexible beam with a rotational load-displacement response that matches a sine. This moment-angle characteristic can be used to balance a pendulum which has a similar but opposite moment-angle characteristic.

The current work is dedicated to the derivation of the sensitivities needed for the shape optimization of a flexible beam with prescribed load-displacement. This procedure is demonstrated on the same case study of the balanced pendulum. This case study is a comprehensible but not trivial case: it requires the stiffness to go from positive, through zero, to negative. The contribution of this paper is to enhance the procedure by adding the sensitivity analysis. Special attention is paid to the formulation of an objective function for general load-displacement tracing cases and to the application of general boundary conditions as nonlinear constraint equations. Putting together these pieces in one work is a contribution that has not been found in literature, but is believed to be helpful for designers of nonlinear springs.

The rest of the paper is structured as follows: After a brief introduction to the IGA framework Sect. 2 is dedicated to the derivation of the objective function and the terms needed for the sensitivity analysis. Section 3 shows a comparison of two optimization runs on a given example problem with various optimization algorithms, with and without use of gradient information. Section 4 shows the result of a validation by comparing the analytical sensitivities with the numerically approximated sensitivities.

2 Method

The present section starts with the problem description of a compliant mechanism with tailored load-displacement response formulated as the minimization of the difference between a desired and an obtained energy-path. Following, a brief explanation is given of the basic concepts of isogeometric analysis (IGA), needed for this paper. There are many literature sources about this method. The reader is referred to Cottrell et al. (2009) for more information. In the remain-

der all components needed for the evaluation of the sensitivity of the objective function are treated.

2.1 Problem description and objective formulation

In the given context, a typical problem consists of obtaining an elastic system where a given force is provided along a given trajectory. Herein the terms force and trajectory can be used interchangeably with moments and rotations. Examples include constant force mechanisms (Lan et al., 2010), bi-stable and multi-stable mechanisms where the load-displacement response has multiple intersections with the zero-load line (Pucheta and Cardona, 2010; Oh, 2008), and static balancing (Chen and Zhang, 2011) where the resulting load-displacement response neutralizes as much as possible an existing load-displacement response of the system to be balanced. In these examples obtaining a certain nonlinear load-displacement response is the goal of the design process.

Provided that the considered forces are conservative, the problem can be reformulated as obtaining a given potential energy along a given trajectory. The use of energy with respect to forces often proves to be convenient due to its scalar nature. The design challenge, here reformulated as a shape optimization problem, consists thus in finding a certain elastic system which, for a series of prescribed boundary displacements and rotations, is compliant with a given potential energy-path.

A simple two-dimensional elastic beam is treated of which the shape is to be determined. Note that the method can in principle be extended to multiple beams and/or applied to other type of elements like shells or solids. In the example treated in Sect. 3 the beam is connected to the base hinge of a pendulum and the shape of the beam is optimized such that the moment characteristic balances the pendulum, see Fig. 4.

In the current framework the given energy-path is described at a discrete number of steps m resulting in a vector of potential energy values, see Fig. 1,

$$\hat{U}(\delta) \approx \hat{U} = \begin{bmatrix} \hat{U}_1(\delta_1) & \hat{U}_2(\delta_2) & \dots & \hat{U}_m(\delta_m) \end{bmatrix} \quad (1)$$

where the hat ($\hat{\cdot}$) symbol refers to the target energy while the actual obtained energy-path is denoted by

$$U(\delta) \approx U = \begin{bmatrix} U_1(\delta_1) & U_2(\delta_2) & \dots & U_m(\delta_m) \end{bmatrix}. \quad (2)$$

In the previous equations the calligraphic \mathcal{U} denotes the potential energy of the system at a single configuration. δ is the imposed displacement value corresponding to the point of the trajectory.

In the case of load-displacement tailoring it is often useful to examine the shape of the energy-path without considering its amplitude. In a bending dominated problem the amplitude is scaled by sizing and material parameters through the Young's-modulus and cross-sectional properties, in the assumption that these parameters are constant over the length

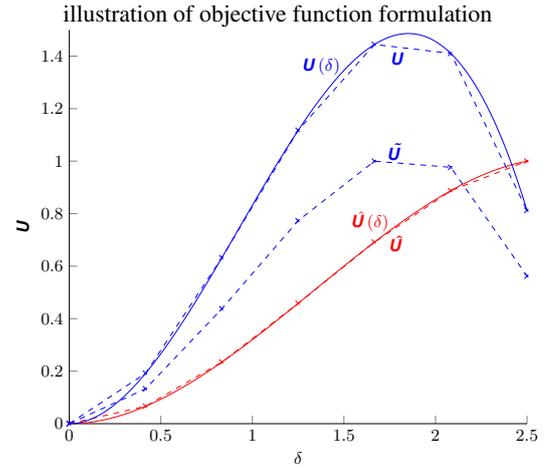


Figure 1. Illustration of energy-paths for objective function. $\hat{U}(\delta) \approx \hat{U}$ is the target behaviour, $U(\delta) \approx U$ is the actual obtained behaviour and \tilde{U} is the actual obtained behaviour normalized for an amplitude-independent comparison.

of the beam and that the Euler–Bernoulli assumptions hold (length \gg thickness). Therefore it is useful to do the optimization on the shape of the energy-path, and once the shape of the target energy-path is achieved, one can scale the amplitude by tuning the sizing and material parameters. In particular, the width of the beam can easily be adjusted to match the desired amplitude.

For this reason the target energy-path \hat{U} is required to be bounded between $[0, 1]$. The normalized obtained energy-path \tilde{U} is defined element-wise by

$$\tilde{U}_k = \frac{U_k - U_{\min}}{U_{\max} - U_{\min}} \quad (3)$$

where U_{\min} and U_{\max} are the minimum and maximum elements of the vector U and the index $k = 1 \dots m$ represents the k th entry of the discrete energy vectors.

The proposed objective function is stated as

$$f_0 = \frac{(\tilde{U} - \hat{U})(\tilde{U} - \hat{U})^T}{\hat{U}\hat{U}^T} \quad (4)$$

which can be interpreted as the normalized sum of squared residuals. As a consequence of the discretisation of the energy-path, the reader should be aware of the fact that the energy difference is minimized only at those discrete points. Fluctuations between the points can theoretically not be excluded. Increasing the resolution of the discretisation helps preventing this.

The given formulation of the objective function is convenient for the sensitivity analysis. Its derivative with respect to the design vector, containing the control point positions

$\mathbf{x} = [P_{1_x} \ P_{1_y} \ \dots \ P_{n_x} \ P_{n_y}]^T$, can be derived as

$$\frac{df_0}{d\mathbf{x}} = 2 \frac{(\tilde{\mathbf{U}} - \hat{\mathbf{U}}) d\tilde{\mathbf{U}}}{\hat{\mathbf{U}} \hat{\mathbf{U}}^T d\mathbf{x}} \quad (5)$$

where the derivative with respect to the normalized energy-path is

$$\frac{d\tilde{\mathbf{U}}}{d\mathbf{x}} = \frac{(U_{\max} - U_{\min}) \left(\frac{dU}{d\mathbf{x}} \Big|_{U=U_{\max}} - \frac{dU}{d\mathbf{x}} \Big|_{U=U_{\min}} \right)}{(U_{\max} - U_{\min})^2} - \frac{\left(\frac{dU}{d\mathbf{x}} \Big|_{U=U_{\max}} - \frac{dU}{d\mathbf{x}} \Big|_{U=U_{\min}} \right) (U - U_{\min})}{(U_{\max} - U_{\min})^2} \quad (6)$$

where $\frac{dU}{d\mathbf{x}} \Big|_{U=U_{\min}}$ and $\frac{dU}{d\mathbf{x}} \Big|_{U=U_{\max}}$ are the derivatives $\frac{dU}{d\mathbf{x}}$ evaluated only for the minimum and maximum entries of U . In the first term this vector, which would be one-dimensional, is replicated and tiled in order to match the dimensions of the vector $\frac{dU}{d\mathbf{x}}$, from which it is subtracted. Also the U_{\min} in the numerator of the second term is subtracted from all elements of the vector U .

Since the potential energy \mathcal{U} at every configuration is a function of both design variables \mathbf{x} and the state variables (displacements) $\mathbf{u}(\mathbf{x})$,

$$\mathcal{U}(\mathbf{x}, \mathbf{u}(\mathbf{x})) \quad (7)$$

the total derivative of the potential energy is given by

$$\frac{d\mathcal{U}(\mathbf{x}, \mathbf{u})}{d\mathbf{x}} = \frac{\partial \mathcal{U}(\mathbf{x}, \tilde{\mathbf{u}})}{\partial \mathbf{x}} + \frac{\partial \mathcal{U}(\tilde{\mathbf{x}}, \mathbf{u})}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\mathbf{x}}. \quad (8)$$

The partial derivative of the energy function with respect to the displacement vector \mathbf{u} is by definition equal to the internal force vector \mathbf{F}_i , which will be used next, but of which the derivation is omitted for the sake of conciseness. Substitution gives

$$\frac{d\mathcal{U}(\mathbf{x}, \mathbf{u})}{d\mathbf{x}} = \frac{\partial \mathcal{U}(\mathbf{x}, \tilde{\mathbf{u}})}{\partial \mathbf{x}} + \mathbf{F}_i \frac{d\mathbf{u}}{d\mathbf{x}}. \quad (9)$$

The partial derivative with respect to the design vector \mathbf{x} is given in explicit form in Sect. 2.3, while the total derivatives of the displacement vector \mathbf{u} with respect to the design vector \mathbf{x} is elaborated in Sect. 2.4.

Note that Eq. (9) must be evaluated at every converged load step solution in order to feed $\frac{d\mathcal{U}}{d\mathbf{x}}$ as columns of the matrix $\frac{dU}{d\mathbf{x}}$ in Eq. (6).

2.2 IGA introduction

Isogeometric analysis is a framework with growing popularity for a number of reasons that particularly hold for shape optimization. First, the fidelity between analysed shape and designed shape. There is no approximation involved in the

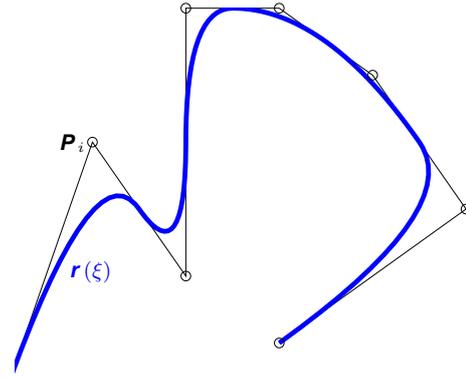


Figure 2. B-spline curve and control polygon.

discretisation of the geometry as with meshing. Instead, a refinement of the parametric description of the geometry is performed, which increases the amount of parameters without altering the geometry itself.

Secondly, and related to the first argument, not needing a conversion step between the geometric description of the design to the analysis gives speed advantages, which are of major impact in optimization where this conversion happens many times.

Third, the availability of analytic derivative information can enhance an optimization procedure significantly.

Since a lot of literature can be found about IGA, just the needed formulas are given to understand the notation. For the notation Nagy (2011) is followed. A B-spline, see Fig. 2, is defined as

$$\mathbf{r}(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{P}_i \quad (10)$$

where $\mathbf{P}_i \in \mathbb{R}^d$ is a control point, n is the number of control points and where the p th degree basis functions are constructed recursively starting with piecewise constants

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise} \end{cases}$$

and for $p = 1, 2, 3, \dots$, they are defined by

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \quad (11)$$

In the present work a geometrically nonlinear Euler–Bernoulli beam is used. The potential energy of the beam can be written as

$$\mathcal{U} = \frac{1}{2} \int (EA\epsilon^2 + EI\rho^2) dS \quad (12)$$

where the integral is evaluated numerically following a Gaussian quadrature rule using

$$\mathcal{U} \approx \mathcal{U} = \frac{1}{2} \sum_{j=1}^{n+p} \sum_{k=1}^{n_{\text{int}}} (EA\epsilon_k^2 + EI\rho_k^2) J_k \bar{J}_j \bar{w}_k \quad (13)$$

where E , A and I represent the Young's modulus, the cross-sectional area and the second moment of inertia, respectively, and are assumed constant over the length of the beam. Moreover J is the reference rod Jacobian, \bar{J} and \bar{w} are the Jacobian and the weight associated with the numerical integration, $n + p$ is the number of knot spans and n_{int} is the number of integration points per knot span.

The relative strain measures, i.e. the membrane strain ϵ and the bending strain ρ are given as defined in Simo (1985)

$$\epsilon = \frac{1}{2} \left(\left(\frac{ds}{dS} \right)^2 - 1 \right) \tag{14}$$

$$\rho = (\kappa - K) \frac{ds}{dS}. \tag{15}$$

The differential arch length and curvature of the line of centroids of the beam are denoted by the kinematic variables dS , ds , K and κ , where the capital symbols refer to the reference state $\mathbf{R}(S)$ and the minuscule symbols refer to the current state $\mathbf{r}(S)$. The definitions are

$$ds(\xi) = \left| \mathbf{r}^{(1)} \right| d\xi \tag{16}$$

$$\kappa(\xi) = \frac{[\mathbf{r}^{(1)} \times \mathbf{r}^{(2)}]_3}{|\mathbf{r}^{(1)}|^3} \tag{17}$$

where the subscript 3 refers to the third component of the vector in brackets and the superscript in parenthesis denote successive differentiation w.r.t. ξ , the variable of parametrization. The definition of dS and K is similar to Eqs. (16) and (17), only replacing the curve of the current configuration $\mathbf{r}(S)$ by the one of the reference configuration $\mathbf{R}(S)$.

It is customary to define the geometry of a system by a relatively small set of control points which are used as optimization variables ($\mathbf{X} = \mathbf{P}^{\text{design}}$). A refinement step, preserving the exact shape of the curve itself, yields a larger amount of control points ($\mathbf{x} = \mathbf{P}^{\text{analysis}}$) that are used for the numerical analysis.

The discrete structural arrays, i.e. tangent stiffness matrix \mathbf{K}_t and the internal force vector \mathbf{F}_i , are derived from the energy functional of Eq. (12). The displacement of the control points of the analysis (\mathbf{x}) are the unknowns of the equilibrium equations indicated by the vector of state variables \mathbf{u} .

2.3 Energy derivative

Often in structural optimization the stored elastic energy, assumed equivalent to the structural compliance, is simply evaluated as $c = \frac{1}{2} \mathbf{F}_e^T \mathbf{u}$. While this is correct for a linear analysis, namely it represents the area under the force-displacement curve, this is not valid in general for nonlinear cases. Therefore we must go back to the definition of the strain energy given in Eq. (12) and its numerical approximation given in Eq. (13). The partial derivative of Eq. (13) with

respect to the vector \mathbf{x} , where \mathbf{u} is kept constant, is given by

$$\frac{\partial \mathcal{U}}{\partial \mathbf{x}} = \frac{1}{2} \sum_{j=1}^{n+p} \sum_{k=1}^{n_{\text{int}}} \left(\left(EA \epsilon_k^2 + EI \rho_k^2 \right) \frac{\partial J_k}{\partial \mathbf{x}} + \left(EA \frac{\partial (\epsilon_k^2)}{\partial \mathbf{x}} + EI \frac{\partial (\rho_k^2)}{\partial \mathbf{x}} \right) J_k \right) \bar{J}_j \bar{w}_k. \tag{18}$$

The partial derivatives of the squared strain measures read

$$\frac{\partial (\epsilon^2)}{\partial \mathbf{x}} = 2\epsilon \frac{\partial \epsilon}{\partial \mathbf{x}} = 2\epsilon \left[\frac{ds}{dS} \frac{\left(\frac{\partial ds}{\partial \mathbf{x}} dS - \frac{\partial dS}{\partial \mathbf{x}} ds \right)}{dS^2} \right] \tag{19}$$

and

$$\frac{\partial (\rho^2)}{\partial \mathbf{x}} = 2\rho \frac{\partial \rho}{\partial \mathbf{x}} = 2\rho \left[\left(\frac{\partial \kappa}{\partial \mathbf{x}} - \frac{\partial K}{\partial \mathbf{x}} \right) \frac{ds}{dS} + (\kappa - K) \frac{\frac{\partial ds}{\partial \mathbf{x}}}{\frac{\partial dS}{\partial \mathbf{x}}} \right] \tag{20}$$

where

$$\frac{\partial ds}{\partial \mathbf{x}} = \frac{\mathbf{r}^{(1)} R_{i,p}^{(1)}}{|\mathbf{r}^{(1)}|} d\xi \tag{21}$$

and

$$\frac{\partial \kappa}{\partial \mathbf{x}} = \frac{\frac{\partial}{\partial \mathbf{x}} ([\mathbf{r}^{(1)} \times \mathbf{r}^{(2)}]_3) |\mathbf{r}^{(1)}|^3 - [\mathbf{r}^{(1)} \times \mathbf{r}^{(2)}]_3 \frac{\partial}{\partial \mathbf{x}} (|\mathbf{r}^{(1)}|^3)}{|\mathbf{r}^{(1)}|^6}. \tag{22}$$

Similar equations hold for $\frac{\partial dS}{\partial \mathbf{x}}$ and $\frac{\partial K}{\partial \mathbf{x}}$, where again the reference curve is used instead of the current curve. Furthermore

$$\frac{\partial}{\partial \mathbf{x}} \left([\mathbf{r}^{(1)} \times \mathbf{r}^{(2)}]_3 \right) = R_{i,p}^{(1)} (r_2^{(2)} - r_1^{(2)}) + (r_1^{(1)} - r_2^{(1)}) R_{i,p}^{(2)}. \tag{23}$$

The derivatives of the curves $\mathbf{r}^{(1)}$, $\mathbf{r}^{(2)}$ and their algorithmic implementations are readily available from e.g. Piegl and Tiller (1997).

2.4 State sensitivity

The more tedious parts of the derivation are not the partial derivatives but the total derivative of the state vector. Usually there is not an explicit relation between \mathbf{x} and \mathbf{u} , and thus the derivative cannot be found analytically. There are two common methods to compute them numerically. One is the direct method and the other one is the adjoint method. The direct method is used because of its simpler implementation and derivation. There is however no objection in using the adjoint method instead.

There are two aspects in the described situation that make the implementation not trivial. These two aspects have not been found combined in literature. The first aspect is that the set of equilibrium equations is nonlinear and thus requires an iterative, newton-like, procedure to solve it. The second

aspect is the application of nonlinear constraint equations as Lagrange multipliers, which creates an augmented system of equations. Both aspects together lead to the following derivations. The formulation of the constraint equations in the present work will be illustrated in Sect. 2.5.

The equilibrium conditions to be solved can be formulated in terms of the tangent stiffness matrix \mathbf{K}_t , the internal and external force vectors \mathbf{F}_i and \mathbf{F}_e

$$\mathbf{K}_t(\mathbf{x}, \mathbf{u})\Delta\mathbf{u} = \mathbf{F}_e(\mathbf{x}, \mathbf{u}) - \mathbf{F}_i(\mathbf{x}, \mathbf{u}) \quad (24)$$

which is solved for an increment of the displacement vector $\Delta\mathbf{u}$. While this is a good practice for the solution of the system itself, for the current derivation of the sensitivities it is more convenient to use the alternative formulation in terms of the secant stiffness matrix, where the internal force vector drops out

$$\mathbf{K}_s(\mathbf{x}, \mathbf{u})\mathbf{u} - \mathbf{F}_e(\mathbf{x}, \mathbf{u}) = \mathbf{0}. \quad (25)$$

Contrary to the former formulation, here the total displacement vector \mathbf{u} is used. This is convenient because the constraint equations will be formulated in terms of the total displacements as well. The general set of constraint equations are noted as

$$\mathcal{F}_2 = \mathbf{A}(\mathbf{x}, \mathbf{u})\mathbf{u} - \mathbf{b}(\mathbf{x}, \mathbf{u}) = \mathbf{0} \quad (26)$$

and adding the constraints as Lagrange multiplier terms to the system in Eq. (25), see Felippa (2004) for a concise explanation, gives

$$\mathcal{F}_1 = \mathbf{K}_s(\mathbf{x}, \mathbf{u})\mathbf{u} + \mathbf{A}(\mathbf{x}, \mathbf{u})^T\boldsymbol{\lambda} - \mathbf{F}_e(\mathbf{x}, \mathbf{u}) = \mathbf{0}. \quad (27)$$

The set of equations in \mathcal{F}_1 and \mathcal{F}_2 collected into the so called augmented form

$$\begin{bmatrix} \mathbf{K}_s(\mathbf{x}, \mathbf{u}) & \mathbf{A}(\mathbf{x}, \mathbf{u})^T \\ \mathbf{A}(\mathbf{x}, \mathbf{u}) & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_e(\mathbf{x}, \mathbf{u}) \\ \mathbf{b}(\mathbf{x}, \mathbf{u}) \end{pmatrix} \quad (28)$$

is normally solved simultaneously for both \mathbf{u} and the Lagrange multipliers $\boldsymbol{\lambda}$ which, pre-multiplied by the \mathbf{A} matrix, can be interpreted as the forces needed to impose the conditions.

Taking the total derivative of both vector equations \mathcal{F}_1 and \mathcal{F}_2 , applying the product rule and the chain rule and collecting $\frac{d\mathbf{u}}{dx}$ and $\frac{d\boldsymbol{\lambda}}{dx}$ gives

$$\begin{aligned} \frac{d\mathcal{F}_1}{dx} &= \left(\mathbf{K}_s(\mathbf{x}, \mathbf{u}) + \frac{\partial[\mathbf{K}_s(\mathbf{x}, \mathbf{u})\tilde{\mathbf{u}}]}{\mathbf{u}} - \frac{\partial\mathbf{F}_e(\mathbf{x}, \mathbf{u})}{\partial\mathbf{u}} \right) \frac{d\mathbf{u}}{dx} \\ &+ \mathbf{A}(\mathbf{x}, \mathbf{u})^T \frac{d\boldsymbol{\lambda}}{dx} + \frac{[d\mathbf{K}_s(\mathbf{x}, \tilde{\mathbf{u}})\tilde{\mathbf{u}}]}{dx} \\ &+ \frac{[d\mathbf{A}(\mathbf{x}, \mathbf{u})^T\tilde{\boldsymbol{\lambda}}]}{dx} - \frac{\partial\mathbf{F}_e(\mathbf{x}, \tilde{\mathbf{u}})}{\partial\mathbf{x}} = \mathbf{0} \end{aligned} \quad (29)$$

and

$$\frac{d\mathcal{F}_2}{dx} = \mathbf{A}(\mathbf{x}, \mathbf{u}) \frac{d\mathbf{u}}{dx} + \frac{d[\mathbf{A}(\mathbf{x}, \mathbf{u})\tilde{\mathbf{u}}]}{dx} - \frac{d\mathbf{b}(\mathbf{x}, \mathbf{u})}{dx} = \mathbf{0}. \quad (30)$$

Here and in the following the tilde ($\tilde{\quad}$) means that the variable is held constant during differentiation. Notice that for the terms containing $\mathbf{A}(\mathbf{x}, \mathbf{u})$ and $\mathbf{b}(\mathbf{x}, \mathbf{u})$ in the present work the total derivative is directly available as will be shown in Sect. 2.5, and thus the chain rule is not being applied to those terms. The following relations between secant and tangent stiffness matrices and between the secant matrix and the internal force vector are given in Ryu et al. (1985)

$$\mathbf{K}_t(\mathbf{x}, \mathbf{u}) = \mathbf{K}_s(\mathbf{x}, \mathbf{u}) + \frac{\partial[\mathbf{K}_s(\mathbf{x}, \mathbf{u})\tilde{\mathbf{u}}]}{\partial\mathbf{u}} \quad (31)$$

$$\frac{\partial[\mathbf{K}_s(\mathbf{x}, \tilde{\mathbf{u}})\tilde{\mathbf{u}}]}{\partial\mathbf{x}} = \frac{\partial\mathbf{F}_i(\mathbf{x}, \tilde{\mathbf{u}})}{\partial\mathbf{x}}. \quad (32)$$

Substituting Eqs. (31) and (32) into Eqs. (29) and (30) and rearranging them into matrix form gives

$$\begin{bmatrix} \mathbf{K}_t - \frac{\partial\mathbf{F}_e(\mathbf{x}, \mathbf{u})}{\partial\mathbf{u}}(\mathbf{x}, \mathbf{u}) & \mathbf{A}(\mathbf{x}, \mathbf{u})^T \\ \mathbf{A}(\mathbf{x}, \mathbf{u}) & \mathbf{0} \end{bmatrix} \begin{pmatrix} \frac{d\mathbf{u}}{dx} \\ \frac{d\boldsymbol{\lambda}}{dx} \end{pmatrix} = \begin{pmatrix} \frac{\partial\mathbf{F}_e(\mathbf{x}, \tilde{\mathbf{u}})}{\partial\mathbf{x}} - \frac{\partial\mathbf{F}_i(\mathbf{x}, \tilde{\mathbf{u}})}{\partial\mathbf{x}} - \frac{d[\mathbf{A}(\mathbf{x}, \mathbf{u})^T\tilde{\boldsymbol{\lambda}}]}{dx} \\ \frac{d\mathbf{b}(\mathbf{x}, \mathbf{u})}{dx} - \frac{d[\mathbf{A}(\mathbf{x}, \mathbf{u})\tilde{\mathbf{u}}]}{dx} \end{pmatrix} \quad (33)$$

which can be solved for $\frac{d\mathbf{u}}{dx}$. Consider that in the case that the external forces are not depending on the displacements, e.g. no follower forces or pressure, the coefficient matrix on the left hand side is equal to the coefficient matrix used for the analysis steps, and is therefore already available in inverted form. This can save considerable computation time. The used constraint equations and its derivative terms are elaborated in Sect. 2.5.

2.5 Boundary constraint equations and derivatives

2.5.1 Constraint equations

In the current setting the system is loaded by applying displacements typically at the endpoints of the beam. Thereby the forces appear as reaction forces of the applied constraints instead of external forces. Once the displacement is applied, equilibrium can be found and the energy is derived. Displacements in this case can also mean rotations. In a typical example one would clamp one end of the beam, i.e. both translations and rotations zero, and apply a given motion on the other end, e.g. travelling along a curved line, or applying a rotation at a fixed point.

There is a complication that arises from the use of the isogeometric analysis method. From the rotation-less character of the control points, it follows that rotations cannot be directly applied. Instead, as described earlier in Radaelli and Herder (2014), a set of nonlinear constraints is applied that dictates the position of the second control point with respect to the first one. It is a given notion that the line connecting the first and the second control point is tangent to the beginning of the curve and, similarly, the line connecting the

second-last to the last control point is tangent to the end of the curve. In the following only the beginning of the curve is considered, omitting the end of the curve. All the equations are similarly derivable replacing the subscripts 1 and 2 by $n - 1$ and n .

In general there are two types of constraints that are applied to the beam. The first is a linear set $\mathbf{A}_1 \mathbf{u} = \mathbf{b}_1$ prescribing the displacement on the endpoints as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \end{bmatrix} \begin{pmatrix} u_{1x} \\ u_{1y} \\ \vdots \end{pmatrix} = \begin{pmatrix} b_{1x} \\ b_{1y} \end{pmatrix} \quad (34)$$

where b_{1x} and b_{1y} are the applied x and y displacements. There are cases where the applied displacement is made dependent on the design vector \mathbf{x} . For instance in the design example given in Sect. 3, where the endpoint of the beam is pre-stressed by positioning the endpoint at the origin of the coordinate system. Here $b_{nx} = -P_{nx}$ and $b_{ny} = -P_{ny}$, and thus $\mathbf{b}_1 = \mathbf{b}_1(\mathbf{x})$.

The second type of constraints concerns the rotations and is somewhat more involved. The inclination h of the tangent line at the beginning of the curve is defined as

$$h(\mathbf{x}, \mathbf{u}) = \tan(\theta_0(\mathbf{x}) + \Delta\theta(\mathbf{x}, \mathbf{u})) \quad (35)$$

where $\Delta\theta$ is the difference of the current angle θ_k from the an initial angle θ_0 . The initial angle depends on the design vector \mathbf{x} and the current angle θ_k on both the design vector \mathbf{x} and the displacements vector \mathbf{u} . Therefore the inclination h is nonlinear in both \mathbf{x} and \mathbf{u} . The inclination h must equal the inclination of the line crossing the first two control points

$$h = \frac{(P_{2y} + u_{2y}) - (P_{1y} + u_{1y})}{(P_{2x} + u_{2x}) - (P_{1x} + u_{1x})}. \quad (36)$$

Rewriting and separating the displacements terms gives the nonlinear set of constraints $\mathbf{A}_{nl}(\mathbf{x}, \mathbf{u}) \mathbf{u} = \mathbf{b}_{nl}(\mathbf{x}, \mathbf{u})$ as

$$\begin{bmatrix} -h(\mathbf{x}, \mathbf{u}) & 1 & h(\mathbf{x}, \mathbf{u}) & -1 & \dots \end{bmatrix} \begin{pmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ \vdots \end{pmatrix} = \begin{bmatrix} h(\mathbf{x}, \mathbf{u})P_{1x} - P_{1y} - h(\mathbf{x}, \mathbf{u})P_{2x} + P_{2y} \end{bmatrix}. \quad (37)$$

The matrices \mathbf{A}_1 and \mathbf{A}_{nl} and the vectors \mathbf{b}_1 and \mathbf{b}_{nl} can be simply concatenated vertically to form a set of equations $\mathbf{A}\mathbf{u} = \mathbf{b}$ to be added to the system as Lagrange constraints, as described in Sect. 2.4.

2.5.2 Derivatives of constraint equations

The derivatives of the constraint equations, needed in Eq. (33), will be derived next.

For the linear part of the constraints only the case where \mathbf{b}_1 is a function of \mathbf{x} , i.e. $\mathbf{b}_1 = \mathbf{b}_1(\mathbf{x})$, is mentioned. This is the case when e.g. a pre-stress proportional to a design parameter is applied to the system. Typically, if \mathbf{b}_1 is linear in \mathbf{x} , the total derivative $\frac{d\mathbf{b}_1(\mathbf{x})}{d\mathbf{x}}$ is a vector of constants.

The derivations for the nonlinear part of the constraints requires more attention. The following holds

$$\frac{d\mathbf{b}_{nl}(\mathbf{x}, \mathbf{u})}{d\mathbf{x}} = \left(\frac{dh}{dx} (P_{1x} - P_{2x}) + h \left(\frac{dP_{1x}}{dx} - \frac{dP_{2x}}{dx} \right) - \frac{dP_{1y}}{dx} + \frac{dP_{2y}}{dx} \right), \quad (38)$$

$$\frac{d[\mathbf{A}_{nl}(\mathbf{x}, \mathbf{u})\tilde{\mathbf{u}}]}{d\mathbf{x}} = \left(\frac{dh}{dx} (u_{2x} - u_{1x}) \right) \quad (39)$$

and

$$\frac{d[\mathbf{A}_{nl}(\mathbf{x}, \mathbf{u})^T \tilde{\boldsymbol{\lambda}}]}{d\mathbf{x}} = \begin{pmatrix} -\frac{dh}{dx} \lambda \\ \vdots \\ \frac{dh}{dx} \lambda \\ \vdots \end{pmatrix}. \quad (40)$$

In these equations the derivative of h is calculated as

$$\begin{aligned} \frac{dh}{dx} &= \frac{d}{dx} (\tan(\theta_0 + \Delta\theta)) = \frac{d}{dx} \left(\frac{h_0 + T_{\Delta\theta}}{1 - h_0 T_{\Delta\theta}} \right) \\ &= \frac{\left(\frac{dh_0}{dx} + \frac{dT_{\Delta\theta}}{dx} \right) (1 - h_0 T_{\Delta\theta}) + (h_0 + T_{\Delta\theta}) \left(\frac{dh_0}{dx} T_{\Delta\theta} + h_0 \frac{dT_{\Delta\theta}}{dx} \right)}{(1 - h_0 T_{\Delta\theta})^2} \end{aligned} \quad (41)$$

where $T_{\Delta\theta}$ is a shorthand notation for $\tan(\Delta\theta)$, h_0 is the initial inclination and their derivatives are given respectively by

$$\frac{dT_{\Delta\theta}}{dx} = \sec(\Delta\theta) \frac{d\Delta\theta}{dx} \quad (42)$$

and

$$\frac{dh_0}{dx} = \frac{d}{dx} \left(\frac{P_{2y} - P_{1y}}{P_{2x} - P_{1x}} \right). \quad (43)$$

Now $\Delta\theta$, which is the difference between the current angle θ_k and the initial angle θ_0 , is split up in the angle of the converged solution of the last iterative step θ_{k-1} , the angle that is imposed in the current iteration θ_{step} , which is known and fixed, minus the reference angle θ_0 . This is a precaution measure. In fact, in the case that a rotation is applied from rest, $\Delta\theta$ is known and fixed. But in the case that the loading history is not fully known, e.g. a pre-stress is applied, than $\Delta\theta$ could contain a certain unknown rotation induced by a previous step. In order to avoid this type of error we define

$$\frac{d\Delta\theta}{dx} = \frac{d}{dx} (\theta_{k-1} + \theta_{\text{step}} - \theta_0) = \frac{d\theta_{k-1}}{dx} - \frac{d\theta_0}{dx} \quad (44)$$

where

$$\frac{d\theta_0}{dx} = \frac{d}{dx} \left(\tan^{-1}(h_0) \right) = \frac{1}{1 + (h_0)^2} \frac{dh_0}{dx} \quad (45)$$

is fairly simple to find, and where the following term is used which has been calculated at the end of the previous converged solution step, where the derivative of the state vector was already solved:

$$\frac{d\theta_{k-1}}{dx} = \frac{\partial\theta_{k-1}}{\partial x} + \frac{\partial\theta_{k-1}}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dx}. \quad (46)$$

The partial derivatives that are needed are

$$\frac{\partial\theta_{k-1}}{\partial x} = \frac{\partial}{\partial x} \left(\tan^{-1}(h_{k-1}) \right) = \frac{1}{1+(h_{k-1})^2} \frac{\partial h_{k-1}}{\partial x} \quad (47)$$

and

$$\frac{\partial\theta_{k-1}}{\partial \mathbf{u}} = \frac{\partial}{\partial \mathbf{u}} \left(\tan^{-1}(h_{k-1}) \right) = \frac{1}{1+(h_{k-1})^2} \frac{\partial h_{k-1}}{\partial \mathbf{u}} \quad (48)$$

which turn out to be the same since

$$\frac{\partial h_{k-1}}{\partial x} = \frac{\partial h_{k-1}}{\partial \mathbf{u}} = \begin{bmatrix} \frac{(P_{2y} + u_{2y}) - (P_{1y} + u_{1y})}{((P_{2x} + u_{2x}) - (P_{1x} + u_{1x}))^2} \\ \frac{(P_{2x} + u_{2x}) - (P_{1x} + u_{1x})}{((P_{2y} + u_{2y}) - (P_{1y} + u_{1y}))^2} \\ \frac{(P_{2x} + u_{2x}) - (P_{1x} + u_{1x})}{((P_{2x} + u_{2x}) - (P_{1x} + u_{1x}))^2} \\ \vdots \end{bmatrix}^T. \quad (49)$$

At this point Eqs. (38)–(40) are fully defined and can be used to find $\frac{d\mathbf{u}}{dx}$ in Eq. (33).

2.6 Refinement term

In geometric design optimization it is a common use to define a geometry at a level with relatively few parameters which is refined to a more dense level at which the analysis is performed. Commonly this is done by meshing, while in isogeometric analysis there are so called refinement techniques, where the same spline curve is refined to a spline with more control points and/or higher order basis functions, but maintaining the exact original shape. It is not worth going much into detail here, given the amount and quality of literature on this topic, e.g. Piegl and Tiller (1997); Cottrell et al. (2009).

The Jacobian of \mathbf{x} , the refined design vector, with respect to \mathbf{X} , the global design vector, is needed for the sensitivity of the objective with respect to the global design vector.

$$\frac{df_0}{d\mathbf{X}} = \frac{df_0}{d\mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{X}} \quad (50)$$

The derivation of term $\frac{d\mathbf{x}}{d\mathbf{X}}$ is not within the scope of this paper, but can be found e.g. in Qian (2010).

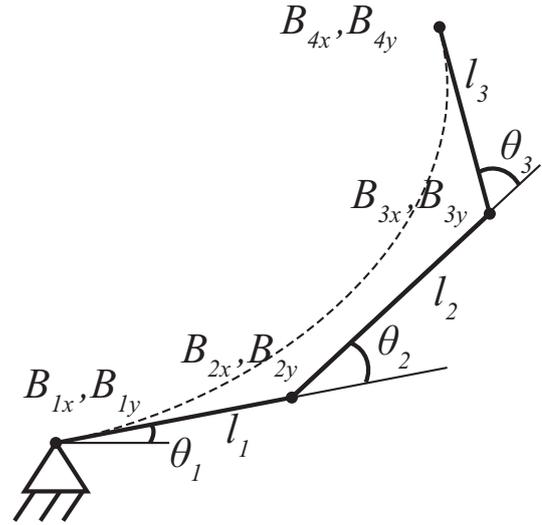


Figure 3. Transformation of coordinates of the control points in a set of generalized coordinates, described by the lengths and angles of the links of a linkage chain representing the control polygon.

2.7 Variable transform

As a last (optional) step a transformation of variables on the global design vector \mathbf{X} to a generalized design vector \mathbf{q} has been adopted. The latter is a vector containing lengths l and relative angles θ of the lines connecting the control points B of the global design as if it were a linkage chain, see Fig. 3. This way it becomes easy, by imposing boundaries on the search space of the optimization, to avoid loops in the curve and avoid consecutive control points lying too close to each other. The first is done by bounding the angles avoiding too sharp corners in the control polygon, and the second is realised by limiting the minimum lengths of the links. In general this avoids awkward shapes that are undesired.

The transformation is defined as

$$\mathbf{X} = \begin{bmatrix} B_{1x} \\ B_{1y} \\ B_{2x} \\ B_{2y} \\ B_{3x} \\ B_{3y} \\ B_{4x} \\ B_{4y} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_1 + q_3c(q_4) \\ q_2 + q_3s(q_4) \\ q_1 + q_3c(q_4) + q_5c(q_4 + q_6) \\ q_2 + q_3s(q_4) + q_5s(q_4 + q_6) \\ q_1 + \dots + q_7c(q_4 + q_6 + q_8) \\ q_2 + \dots + q_7s(q_4 + q_6 + q_8) \end{bmatrix} \quad (51)$$

where c and s are the shorthand notations for cos and sin, and \mathbf{q} defined as

$$\mathbf{q} = [B_{1x} \ B_{1y} \ l_1 \ \theta_1 \ l_2 \ \theta_2 \ l_3 \ \theta_3]. \quad (52)$$

Note that in a design with more control points \mathbf{q} can be longer than shown, in that case the expression would expand in a similar fashion. For the sensitivity of the objective with re-

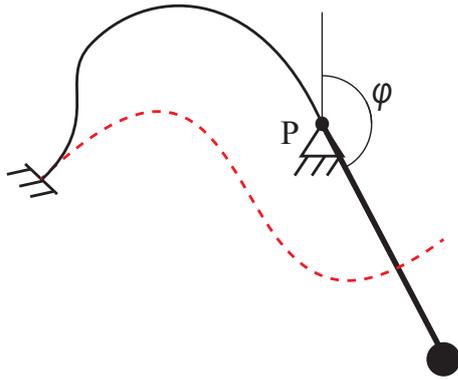


Figure 4. Topology of the system: A leaf spring (red dashed) is pre-stressed (black) by connecting one endpoint with a rigid pendulum, while the other end of the spring is clamped.

spect to the generalized variables $\frac{dX}{dq}$ is needed because

$$\frac{df_0}{dq} = \frac{df_0}{dX} \frac{dX}{dq}. \tag{53}$$

The full expression of $\frac{dX}{dq}$ is omitted for the sake of conciseness. Its derivation can however be considered straightforward. At this point all expressions for the sensitivity analysis are derived.

3 Design example

To assess the usefulness of the gradients in the given design problem several optimization runs are compared where one gradient-free algorithm and four gradient-based algorithms are used. The first type only needs Eq. (4) as an input, while the last four also use Eq. (53). The algorithms implemented in the *optimization toolbox* in Matlab® are used: Nelder–Mead Simplex (NMS), Trust-Region-Reflective (TRR), Interior-Point (IP), Active-Set (AS) and Sequential Quadratic Programming (SQP). See the Matlab® (2014) documentation for details on the algorithms.

Because of the high dependency with the starting point of the optimization, the runs are all performed starting at two different initial points, chosen such that the resulting behaviour would be clearly distinct.

The case study is similar to the one found in Radaelli and Herder (2014). The goal is to design a leaf spring able to statically balance a pendulum in the range from 0 to 180°, see Fig. 4. The objective is to make the reaction moment on the end of the beam follow a sinus-shaped characteristic with respect to the rotation of that point. Equivalently, knowing that the derivative of the energy with respect to the rotation equals the moment, the target energy-path $\hat{U}(\delta)$ is defined as a negative cosine function bounded in amplitude between 0 and 1, as prescribed for Eq. (3). Thus

$$\hat{U}(\varphi) = \frac{(1 - \cos(\varphi))}{2} \tag{54}$$

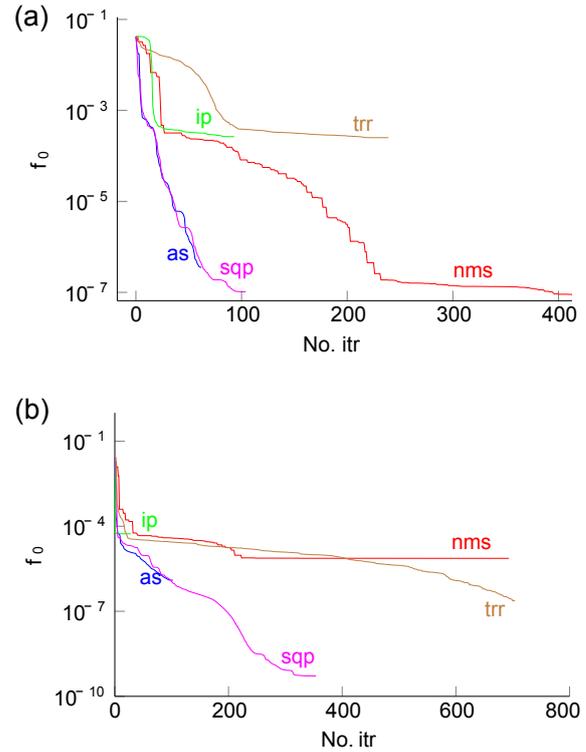


Figure 5. Objective function value vs iteration number for the algorithms Nelder–Mead Simplex (NMS), Trust-Region-Reflective (TRR), Interior-Point (IP), Active-Set (AS) and Sequential Quadratic Programming (SQP). (a) Example 1 and (b) Example 2.

where φ is the rotation of the endpoint P . Before this actual working range a pre-stress step is applied where one endpoint stays clamped, and the other end is brought to the origin of the coordinate system leaving the rotation free. This guarantees that the moment before the first actual step is zero, corresponding to the needed moment when the pendulum is upright.

Some details on relevant design choices are: cross-sectional width and height are 0.01 and 0.002 m, Young’s modulus is 135 GPa. The design curve is a second order B-spline with four control points and uniform knot vector. The curve is refined for analysis with 20 additional knot evenly spread over the knot vector. The used bounds on the variables of optimizations: $q_1, q_2 : [-0.3, 0.3]$, $q_3, q_5, q_7 : [1, 0.4]$ and $q_4, q_6, q_8 : [-2, 2]$.

Figure 5a and b show the progress of the objective function plotted against the iteration steps for the two initial points

$$q_0 = [-0.2 \quad -0.2 \quad 0.2 \quad 1.5 \quad 0.2 \quad -1.5 \quad 0.2 \quad -1.5] \tag{55}$$

and

$$q_0 = [-0.2 \quad -0.2 \quad 0.2 \quad 1.5 \quad 0.2 \quad -1.5 \quad 0.2 \quad 1.5]. \tag{56}$$

The corresponding initial shapes and the energy trajectories are shown in Figs. 6a–9a. The shapes and energy-paths after

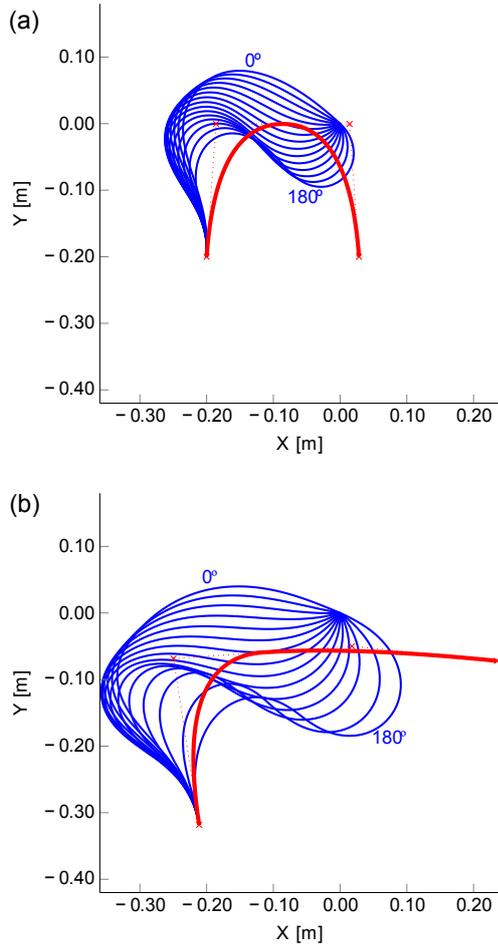


Figure 6. Example 1: shape in neutral position (red) and at the 15 increments of applied rotation of the endpoint about the origin (blue). (a) Initial shape and (b) optimum shape.

optimization (the best result obtained) are shown in Figs. 6b–9b. The thick red lines in the shape plots is the unloaded shape of the beam, while the thin blue lines represent the deformed states at the 15 load-steps. The red line in each energy plot is the target energy-path \hat{U} , while the blue crosses represent the actual normalized energy-path \tilde{U} . The optimized results have been evaluated at smaller increments of the displacement to verify sufficient smoothness between the original increment points. The energies at the smaller increments are shown as grey dots in Fig. 7a and b.

4 Comparison with finite differences

This section is dedicated to a comparison of the result obtained in Sect. 2 with a numerical approximation of the sensitivity. The goal is twofold. One is to verify the validity of the derived equations and the other is to underline one advantage of the analytical sensitivity. Namely the independence between sensitivity and perturbation size.

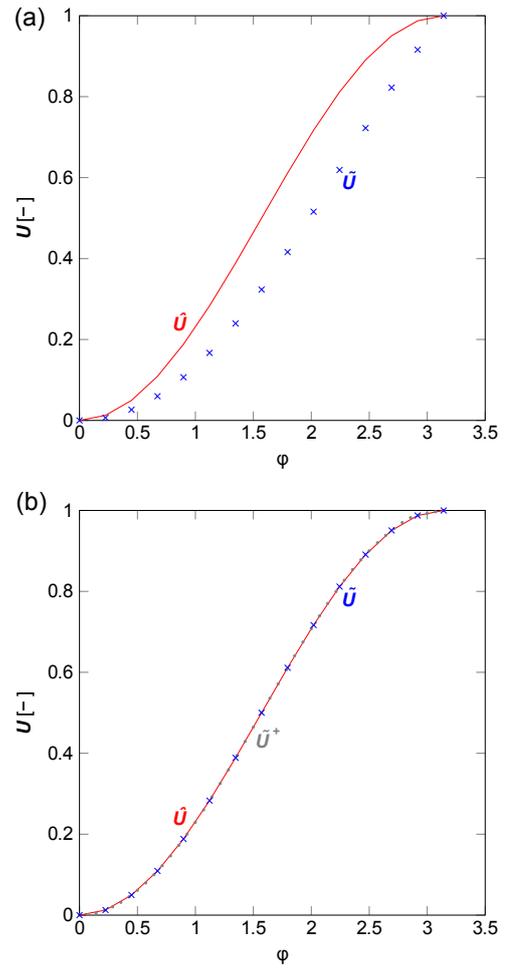


Figure 7. Example 1: target energy-path \hat{U} (red line), normalized actual energy-path \tilde{U} (blue crosses) and normalized actual energy path at smaller increments \tilde{U}^+ (grey dots). (a) Initial shape and (b) optimum shape.

In this analysis the gradient of the objective function with respect to the generalized coordinated $\frac{df_0}{dq}$ is approximated by finite differences, using different perturbation sizes p ranging from 10^{-18} to 10^0 . The first is near or smaller than machine precision, while the latter is in this case obviously an overly large perturbation with respect to the physical dimensions of the system.

An error norm is defined to compare the analytical sensitivities with the finite difference sensitivities. The used norm is the normalized mean error e between the values of the analytical $\frac{df_0}{dq}$, Eq. (55), and its numerical approximation $(\frac{df_0}{dq})_{FD}$.

$$e = \frac{\text{mean} \left(\left| \left(\frac{df_0}{dq} \right) - \left(\frac{df_0}{dq} \right)_{FD} \right| \right)}{\text{mean} \left(\left| \left(\frac{df_0}{dq} \right) \right| \right)}. \tag{57}$$

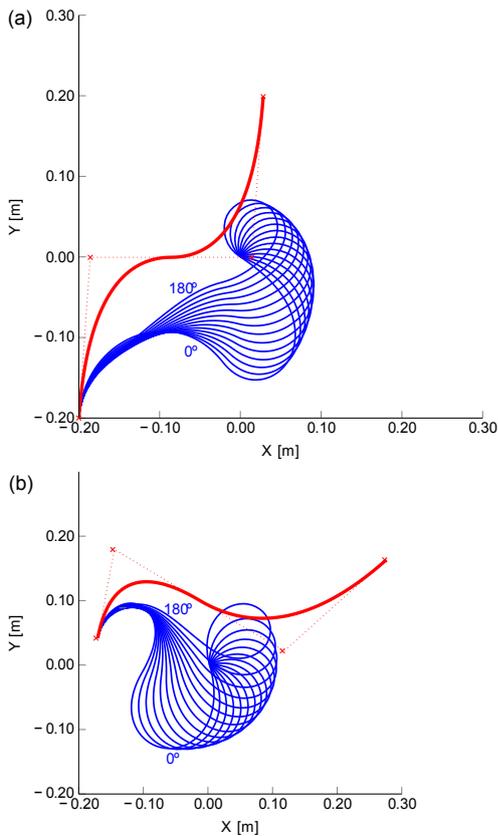


Figure 8. Example 2: shape in neutral position (red) and at the 15 increments of applied rotation of the endpoint about the origin (blue). **(a)** Initial shape and **(b)** optimum shape.

The analysis is performed at two different points. One is the first starting point of the optimization shown in Sect. 3. Another is near one of the found minima. The sensitivities at an optimum point are zero. The error e includes a division by zero in that case. Therefore this error is evaluated at a point near an optimum, and not at an optimum.

Figure 10 shows the error e for a range of perturbation sizes. The blue line (crosses) represents the error at the start point \mathbf{q}_0 and the red line (circles) represents the error at the point near the optimum.

5 Discussion

The optimization runs on the given example are considered successful in the sense that for both starting points, depending on the algorithm, a shape could be found that matches the given energy-path closely. Closely means that the significance of the remaining error is expected to be far beneath other types of errors expected in a physical realization of the concept. The optimized shapes have been analysed again with smaller increments of the rotation. Minimal differences

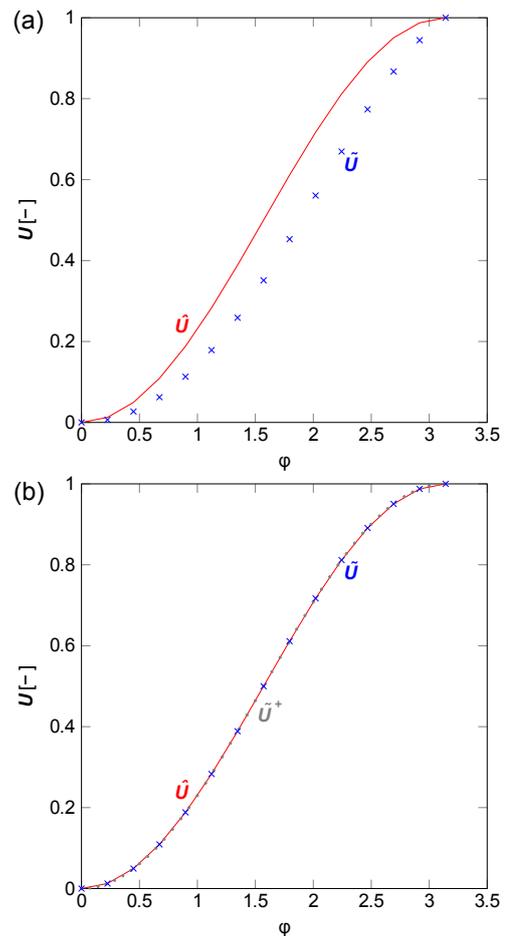


Figure 9. Example 2: target energy-path \hat{U} (red line), normalized actual energy-path \tilde{U} (blue crosses) and normalized actual energy path at smaller increments \tilde{U}^+ (grey dots). **(a)** Initial shape and **(b)** optimum shape.

can be observed, meaning that the chosen amount of load steps was suitable for this example.

The objective function is non-convex, meaning that it cannot be guaranteed that the global minima are found in the examples. However, in the current scope it is not of practical relevance, as long as a “good-enough” minimum is obtained. In fact the nature of the objective function, which cannot become negative, tells that if the solution is close enough to zero within relevant significance the goal has been achieved.

The use of sensitivity information on the illustrated example has shown its utility in Fig. 5a and b. It can be seen that with respect to the Nelder–Mead Simplex algorithm, especially with the Active-Set and the Sequential Quadratic Programming algorithms, either a quicker (more efficient) or deeper (more effective) descent is realized, and sometimes both. It is, however, not guaranteed that this is always the case.

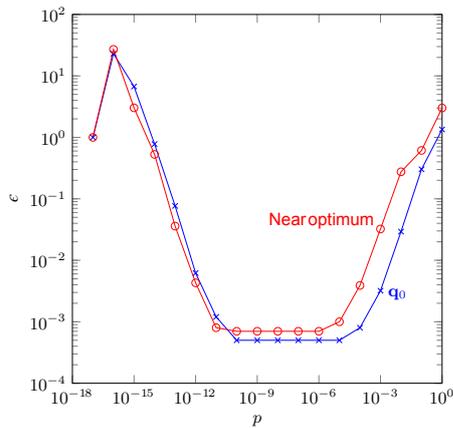


Figure 10. Error between finite-difference sensitivity and analytical sensitivities according to Eq. (57) at an initial configuration q_0 and near one of the optimum configurations, plotted on a range of perturbation sizes.

It is also noticed that the Interior-Point and the Trust-Region Reflective algorithms fail to obtain either a lower minimum or a quicker descent. This emphasizes the importance of the choice of the algorithm. Investigation on the optimal algorithm has not been the scope of this paper but is an important topic of further research. The optimal choice of the algorithm could be subject to factors like the number of design variables, boundary conditions and other constraints, and thus it may vary from case to case.

The comparison plots in Fig. 5a and b show the number of iterations on the horizontal axis. It must be noted however that for the Nelder–Mead Simplex the computation time of every iteration is shorter because it does not need to compute the sensitivity terms. The time saved at every iteration step highly depends on the performance of the code and the used hardware. To give an indication: Based on a Matlab code running on a Intel® Core™ i7 processor the additional computation cost for the sensitivity terms is about 60%. Nevertheless, the advantages of using sensitivity seem to hold.

The finite difference check is quite satisfying. It shows that there is a large range of perturbation sizes, from $p = 10^{-6}$ to $p = 10^{-11}$ where the difference between analytical and finite-difference sensitivities is small. On the one hand this tells that the calculations of the analytical sensitivities are correct. On the other hand it says that within this range the numerical sensitivities would give similar results, only less efficiently. However it still remains a risk that this range may vary from case to case, rising the chance to obtain imprecise or even useless numerical sensitivities.

The points where the comparison is performed are at one of the starting points of the optimization, and in the neighbourhood of the end of the optimization run. As discussed above, it would make little sense to compare the sensitivities at the end of an optimization run. Assuming this is an opti-

num, the sensitivities would be zero. At an optimum Eq. (57) would divide by zero giving an invalid error norm. It is therefore not surprising that the error norm near the optimum solution is higher than at a point far from an optimum. Similar tendencies as in Fig. 10 have been observed starting at various initial points.

In Sect. 2.5.2 a measure is taken to include the possibility to apply pre-stress on the system, thus inducing the system into a configuration where the angles of the endpoints are not known a priori. Pre-stress is crucial in certain applications, especially those involving static balancing. However, if this is not required, then the procedure described in Sect. 2.5.2 will simplify significantly. Namely θ_{k-1} becomes fixed and thus its derivative zero. This reduces Eq. (44) to

$$\frac{d\Delta\theta}{dx} = -\frac{d\theta_0}{dx} \quad (58)$$

and the part after that, Eqs. (45)–(49), can be skipped.

In the definition of an amplitude invariant objective function the assumption was made that the sizing parameters would not affect the results. The Euler–Bernoulli conditions must be met in order for the numerical model to be valid in the first place. This means that shear terms are neglected and this is true for relatively long and slender beams. In the second place, also the stretch terms must be neglected in order for the thickness not to influence the energy-path. For a beam with the same shape but different thickness, namely, the stress condition is influenced by the stretch terms, and thus it affects the deformation and stored energy. It is verified that in the given example, multiplying the thickness by a factor of two yields a maximum amplitude deviation of 2.5%, and multiplying by a factor of ten yields a deviation of 4%. Care is thus needed in the choice of a suitable thickness when designing such system. The width of the spring seems in practice the most appropriate parameter to adjust after optimization without affecting the result.

6 Conclusions

The paper presents the shape sensitivity analysis for the shape optimization of beams with prescribed load-displacement response. The work is based on an isogeometric framework. It has been shown that it is possible to derive the sensitivity parameters for this type of problem analytically.

A novel and general objective function is formulated for problems with prescribed load-displacement response. The objective function is based on the potential energy of the beam determined at discrete steps of the applied displacement path.

The travelled path is imposed by application of constraints on the endpoints of the beam, involving displacements but also rotations, which is more complex due to the rotation-less character of the degrees of freedom in isogeometric analysis.

The sensitivity of the state variables is determined through the direct method, with special attention to the complications brought by nonlinear equilibrium equations and the nonlinear constraints equations, applied as Lagrange multipliers.

The effect of the load history on the sensitivity analysis, e.g. by application of pre-stress in a previous load-step, is neutralized by making smart use of the sensitivities previously calculated for the previous load steps.

The optimization is performed on an example where the optimal shape of a beam is sought that is optimal for the static balancing of a pendulum. The influence of the use of sensitivity information is shown by comparison on different optimization algorithms, with and without the use of gradients. The optimization results in very satisfying balancing springs. The use of gradients is positively influencing the efficiency of the optimization, although determining the algorithm that performs best is still an open question.

The correctness of the derived sensitivity equations is verified by comparison with finite-difference gradients. It is shown that the numerical and the analytical gradients have a good match within a certain range of perturbation sizes.

7 Data availability

The matlab figure files that are used to generate the figures of this paper are attached as Supplement.

The Supplement related to this article is available online at doi:10.5194/ms-7-219-2016-supplement.

Acknowledgements. The authors would like to acknowledge STW (HTSM-2012 12814: ShellMech) for the financial support of this project.

Edited by: A. Eriksson

Reviewed by: A. Hatamizadeh and one anonymous referee

References

- Bruns, T. E. and Tortorelli, D. A.: Topology optimization of nonlinear elastic structures and compliant mechanisms, *Comput. Meth. Appl. Mech. Eng.*, 190, 3443–3459, doi:10.1016/S0045-7825(00)00278-4, 2001.
- Chen, G. and Zhang, S.: Fully-compliant statically-balanced mechanisms without prestressing assembly: concepts and case studies, *Mech. Sci.*, 2, 169–174, doi:10.5194/ms-2-169-2011, 2011.
- Cho, S. and Ha, S.-H.: Isogeometric shape design optimization: exact geometry and enhanced sensitivity, *Struct. Multidiscip. Optimiz.*, 38, 53–70, doi:10.1007/s00158-008-0266-z, 2009.
- Cottrell, J. A., Hughes, T. J. R., and Bazilevs, Y.: *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons, Chichester, England, 2009.
- Ding, Y.: Shape optimization of structures a literal survey, *Comput. Struct.*, 24, 985–1004, 1986.
- Du, Y. and Chen, L.: Topology optimization for large-displacement compliant mechanisms using element free galerkin method, *Int. J. CAD/CAM*, 8, 1–10, 2008.
- Eriksson, A.: Constraint paths in non-linear structural optimization, *Comput. Struct.*, 140, 39–47, doi:10.1016/j.compstruc.2014.05.003, 2014.
- Felippa, C. A.: Introduction to finite element methods (ASEN 5007), University Lecture Notes, University of Colorado, <http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/> (last access: 1 November 2016), 2004.
- Haftka, R. T.: Simultaneous analysis and design, *AIAA J.*, 23, 1099–1103, doi:10.2514/3.9043, 1985.
- Hoetmer, K., Woo, G., Kim, C., and Herder, J.: Negative Stiffness Building Blocks for Statically Balanced Compliant Mechanisms: Design and Testing, *J. Mech. Robot.*, 2, 041007-1-7, doi:10.1115/1.4002247, 2010.
- Howell, L. L.: *Compliant Mechanisms*, John Wiley and Sons Inc., New York, USA, 2001.
- Hsu, Y.-L.: A review of structural shape optimization, *Comput. Indust.*, 26, 3–3, 1994.
- Hughes, T., Cottrell, J., and Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput. Meth. Appl. Mech. Eng.*, 194, 4135–4195, doi:10.1016/j.cma.2004.10.008, 2005.
- Joo, J., Kota, S., and Kikuchi, N.: Large deformation behavior of compliant mechanisms, *Proceedings of DETC'01 ASME 2001 Design Engineering Technical Conference and Computers and Information in Engineering Conference*, 9–12 September 2001, Pittsburgh, PA, 2001.
- Jutte, C. V. and Kota, S.: Design of Single, Multiple, and Scaled Nonlinear Springs for Prescribed Nonlinear Responses, *J. Mech. Design*, 132, 011003, doi:10.1115/1.4000595, 2010.
- Kiendl, J. M.: *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, PhD Thesis, Munich University of Technology, Munich, Germany, 2011.
- Koo, B., Ha, S.-H., Kim, H.-S., and Cho, S.: Isogeometric Shape Design Optimization of Geometrically Nonlinear Structures, *Mech. Bas. Design Struct. Mach.*, 41, 337–358, doi:10.1080/15397734.2012.750226, 2013.
- Lan, C.-C., Wang, J.-H., and Chen, Y.-H.: A compliant constant-force mechanism for adaptive robot end-effector operations, in: *2010 IEEE International Conference on Robotics and Automation*, Anchorage Convention District, 3–8 May 2010, Anchorage, Alaska, USA, 2010.
- Leishman, L. C. and Colton, M. B.: A Pseudo-Rigid-Body Model Approach for the Design of Compliant Mechanism Springs for Prescribed Force – Deflections, *ASME*, 93–102, doi:10.1115/DETC2011-47590, 2011.
- Matlab®: *MATLAB and Optimization Toolbox Release R2014a*, The MathWorks, Inc., Natick, Massachusetts, USA, http://www.mathworks.nl/help/releases/R2014a/pdf_doc/optm/optm_tb.pdf (last access: 1 November 2016), 2014.
- Nagy, A. P.: *Isogeometric Design Optimization*, PhD Thesis, Delft University of Technology, Delft, the Netherlands, 2011.
- Nagy, A. P., Abdalla, M. M., and Gürdal, Z.: Isogeometric sizing and shape optimisation of beam structures, *Comput. Meth. Appl.*

- Mech. Eng., 199, 1216–1230, doi:10.1016/j.cma.2009.12.010, 2010.
- Oh, Y.: Synthesis of Multistable Equilibrium Compliant Mechanisms, PhD Thesis, The University of Michigan, Michigan, USA, 2008.
- Pedersen, C. B., Fleck, N. A., and Ananthasuresh, G. K.: Design of a Compliant Mechanism to Modify an Actuator Characteristic to Deliver a Constant Output Force, *J. Mech. Design*, 128, 1101–1112, doi:10.1115/1.2218883, 2005.
- Piegl, L. and Tiller, W.: *The NURBS Book*, 2nd Edn., Springer-Verlag, New York, NY, USA, 1997.
- Powell, K. M. and Frecker, M. I.: Method for Optimization of a Nonlinear Static Balance Mechanism With Application to Ophthalmic Surgical Forceps, in: *Proceedings ASME IDETC/CIE 2005*, 24–28 September 2005, Long Beach, California, USA, doi:10.1115/DETC2005-84759, 2005.
- Pucheta, M. A. and Cardona, A.: Design of bistable compliant mechanisms using precision – position and rigid-body replacement methods, *Mech. Mach. Theory*, 45, 304–326, doi:10.1016/j.mechmachtheory.2009.09.009, 2010.
- Qian, X.: Full analytical sensitivities in NURBS based isogeometric shape optimization, in: *Computer Methods in Applied Mechanics and Engineering*, 2010.
- Radaelli, G. and Herder, J. L.: Isogeometric Shape Optimization for Compliant Mechanisms with Prescribed Load Paths, in: *Proceedings ASME IDETC/CIE 2014*, 17–20 August 2014, Buffalo, New York, USA, doi:10.1115/DETC2014-35373, 2014.
- Rai, A., Saxena, A., Mankame, N. D., and Upadhyay, C. S.: On Optimal Design of Compliant Mechanisms for Specified Nonlinear Path Using Curved Frame Elements and Genetic Algorithm, *ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 2, 30th Annual Mechanisms and Robotics Conference, Parts A and B, 10–13 September 2006, Philadelphia, Pennsylvania, USA, 91–100, doi:10.1115/DETC2006-99298, 2006.
- Ringertz, U. T.: Optimization of Structures with Nonlinear Response, *Eng. Optimiz.*, 14, 179–188, doi:10.1080/03052158908941210, 1989.
- Ryu, Y., Haririan, M., Wu, C., and Arora, J.: Structural design sensitivity analysis of nonlinear response, *Comput. Struct.*, 21, 245–255, 1985.
- Saxena, A. and Ananthasuresh, G. K.: Topology Synthesis of Compliant Mechanisms for Nonlinear Force – Deflection and Curved Path Specifications, *J. Mech. Design*, 123, 33–42, doi:10.1115/1.1333096, 1999.
- Sigmund, O.: On the Design of Compliant Mechanisms Using Topology Optimization, *Mech. Struct. Mach.*, 25, 493–524, doi:10.1080/08905459708945415, 1997.
- Simo, J.: A finite strain beam formulation. The three-dimensional dynamic problem. Part I, *Comput. Meth. Appl. Mech. Eng.*, 49, 55–70, doi:10.1016/0045-7825(85)90050-7, 1985.
- Tolou, N.: *Statically Balanced Compliant Mechanisms for MEMS and Precision Engineering*, PhD Thesis, Delft University of Technology, Delft, the Netherlands, 2012.
- Vehar-Jutte, C.: *Generalized Synthesis Methodology of Nonlinear Springs for Predescribed Load-Displacement Functions*, PhD Thesis, The University of Michigan, Michigan, USA, 2008.
- Wall, W. A., Frenzel, M. A., and Cyron, C.: Isogeometric structural shape optimization, *Comput. Meth. Appl. Mech. Eng.*, 197, 2976–2988, doi:10.1016/j.cma.2008.01.025, 2008.