Computer-Aided Design 79 (2016) 27–35

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Automatic routing of flexible 1D components with functional and manufacturing constraints

T. Hermansson[a,*], R. Bohlin[a], J.S. Carlson[a], R. Söderberg[b]

[a] Fraunhofer-Chalmers Centre, Chalmers Science Park, SE-412 88 Göteborg, Sweden

[b] Product and Production Development, Chalmers University of Technology, SE-412 96 Göteborg, Sweden

ARTICLE INFO

Article history:
Received 3 July 2015
Accepted 13 May 2016

Keywords:
Automatic routing
Design constraints
Flexible 1D components
Motion planning
Non-linear optimization

ABSTRACT

This article presents a novel and unifying method for routing of flexible one-dimensional components such as cables, hoses and pipes with geometric design constraints. A deterministic and resolution complete grid search is used to find a nominal configuration of the component that is collision-free and satisfies functional and manufacturing constraints. Local refinement is done in tandem with a computationally efficient and physically accurate simulation model based on Cosserat rod theory to ensure that the deformed configuration still satisfies functional constraints when influenced by gravity. Test results show that the method is able to solve industrial scenarios involving complex geometries and real constraints with different objectives in mere seconds.

1. Introduction

1.1. Background

In the early design phase of a product, the competition for the free design space is tough between different components. For example, the automotive industry of today is focusing on electrified and hybrid solutions, where often both conventional combustion engines and battery supplied electrical engines need to fit in an already densely packed vehicle. The placement of each component must be evaluated with respect to geometric interference with other disciplines and their components.

Specifically, flexible 1D components such as pipes, tubes, hoses and electric cables need to be routed as efficiently as possible, while still respecting functional and manufacturing constraints and requirements. Manufacturing constraints are typically imposed by process limitations and design limits, e.g. a pipe can only be pre-bent by a machine with a certain bending radius and a required straight clamping length in between bends. Functional constraints can be clearance to the surrounding obstacles, stress limits or design requirements for the component to function, e.g. the bending radius of a hose may impact the flow of the contained fluid. If there are several feasible design alternatives at hand, then one usually wants the one that is optimal with respect to material consumption (routing length), manufacturing cost (preformation) and gives preference to occupation of certain regions of space (placement).

It is of high industrial impact if a flexible 1D component can be routed by means of virtual methods already in early design phases (see Fig. 1).

1.2. Routing of flexible 1D components

A *flexible 1D component* is characterized as a slender object in $\mathbb{R}^3$ where one dimension (the length) is significantly larger than the other two (the cross section) and that exhibits an elastic behaviour. It could for instance be a rubber hose, an electric cable or a plastic or steel pipe.

Let q denote the *configuration* of a flexible 1D component and $\mathcal{S}(q) \subset \mathbb{R}^3$ its corresponding volumetric shape. Also, let $\mathcal{W} \subset \mathbb{R}^3$ denote the surrounding obstacles. A configuration is said to be collision-free if $\mathcal{S}(q) \cap \mathcal{W} = \emptyset$.

Routing of flexible 1D components amounts to finding a 'good' configuration of a component in static mechanical equilibrium that satisfies geometric design constraints inherited from both manufacturing limitations and functional requirements. The influence of gravity makes this a tremendously hard task. Therefore, in a segregated approach, the task can be separated into two procedures:

Nominal routing. Nominal routing is the task of finding a 'good' nominal configuration q_0 that (1) connects two connection points

[*] Corresponding author.
E-mail address: tomas.hermansson@fcc.chalmers.se (T. Hermansson).

http://dx.doi.org/10.1016/j.cad.2016.05.018
0010-4485/© 2016 Elsevier Ltd. All rights reserved.

Fig. 1. Assembly of a high-voltage power cable in a hybrid car. © 2014, Volvo Car Corporation.

Q_S and Q_G, (2) is collision-free and (3) is feasible with respect to a set of geometric functional and manufacturing constraints of the form $h_m(q_0) \geq 0$—or reporting that no such configuration exists. The nominal configuration q_0 represents the manufactured[1] component at rest. A 'good' configuration is usually characterized as the minimizer of a cost functional related to component length and placement.

Local refinement. When a component with a nominal configuration q_0 is held at Q_S and Q_G and subjected to gravity, it assumes a configuration q in static mechanical equilibrium. Local refinement is the task of ensuring that the deformed configuration q of the flexible component is still 'good', collision-free and feasible with respect to a set of geometric functional constraints of the form $h_f(q) \geq 0$.

1.3. Related work

Extensive research has been done in the field of routing of flexible 1D components; specialized engineering methods have been derived from motion planning techniques and tailored for different component types and objectives.

Specialized hose, cable and pipe routing methods can be categorized into cell-decomposition methods [1,2], graph-based network optimization techniques [3,4] and heuristic algorithms [5]. The aim is usually to generate schematic Manhattan-style layouts in, for instance, plants and ships using as few turns as possible or following geodesic lines on, for instance, aero-engine surfaces. Engineering methods for routing electrical cables have been developed for different scenarios [6]. Channel routing algorithms have been developed for automated design of circuit layouts [7]. In power cable routing, macro-scale methods are used to optimize routes with respect to terrain information [8]. Also, genetic algorithms have been applied in some extent to solve the problem [9].

From a motion planning perspective, nominal routing of a slender component with a constant cross section profile in $\mathbb{R}^3$ can be interpreted as the problem of finding a collision free motion for a slice of the component cross section subject to curvature constraints in the Special Euclidean Group SE(3).

Determining the existence of a collision-free trajectory has proven to be NP complete for the general motion planning problem [10]. Since complete algorithms are of little industrial relevance because they are too slow, different sampling based techniques trading completeness for speed and simplicity have gained much interest. Probabilistic complete methods such as the Probabilistic Roadmap Method (PRM, [11,12]) and Rapidly-Exploring Random Trees (RRT, [13]), are capable of solving problems with many degrees of freedom. Deterministic and resolution complete methods solve the problem in finite time with a sufficiently fine resolution [14,15]. The relationship between deterministic grid search and probabilistic sampling methods is described in [16]. Dynamic planning methods that incrementally improve upon suboptimal solutions given updated information about the problem or limited computation time have been developed [17]. In *non holonomic motion planning*, differential constraints usually reduce the controllable degrees of freedom to fewer than the state space dimension. Analytical methods have been derived for car-like vehicles in 2D in [18,19]. PRM and RRT based methods have been adapted to different under actuated robots in the presence of obstacles; [20–23] implemented deterministic space-filling trees suited for this type of motion planning. Specifically, methods for routing of steerable needles with curvature constraints have been presented in [24,25]. For good and comprehensive overviews on motion planning, the reader is encouraged to read [26–29].

Adding the requirement that the component must also be in static mechanical equilibrium makes the motion planning tremendously more difficult; Incorporating a simulation model in a routing algorithm is very expensive due to the problem's computational complexity. Also, a flexible component can have infinitely many configurations corresponding to the same boundary conditions. Motion planning algorithms for linear deformable objects, most notably in [30–33], have been developed with a focus on verifying the existence of an assembly operation rather than optimizing the design of the intended target configuration. However, no deterministic grid search has been coupled to a simulation model for routing with both manufacturing and functional constraints. In separating the problem into a nominal routing step and a local refinement step, routing of flexible objects can be achieved with both high resolution and physically accurate deformations.

1.4. Scope

This article presents a novel and unifying method for routing of flexible one-dimensional components with geometric design constraints. The proposed method combines a deterministic and resolution complete grid search with a local optimization algorithm coupled to a computationally efficient and physically accurate simulation model.

An overview of the method framework is given in Section 2. In Sections 3 and 4 we formally state and present solution methods for the nominal routing and local refinement problems respectively. Test results are presented in Section 5 and our findings are concluded in Section 6.

2. Method overview

Our proposed method solves the nominal routing problem and the local refinement problem in a closed loop framework as illustrated in Fig. 2.

The `NominalRouter` algorithm seeks a nominal configuration q_0 between Q_S and Q_G that is collision-free, feasible with respect to geometric functional and manufacturing constraints $h_m \geq 0$ and favourable with respect to a given cost functional. A set of routing parameters dictate the quality and the properties of the configuration.

If a solution is found, it may serve as the initial and (optionally) as the nominal configuration in a simulation model of the component. The `LocalRefiner` algorithm attempts to ensure that the deformed configuration q is still collision-free, feasible with respect to the geometric functional constraints $h_f \geq 0$ and

[1] Nominal configurations are typically realized through piecewise bending of pipes, moulding of hoses and clip attachment and/or taping of electric cables respectively.

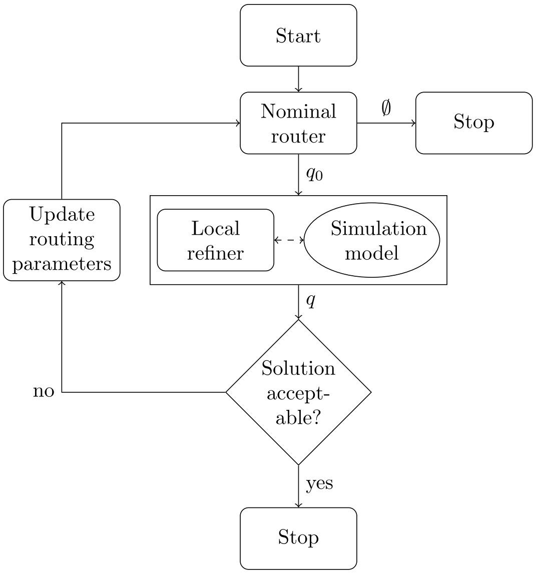


Fig. 2. Method overview.

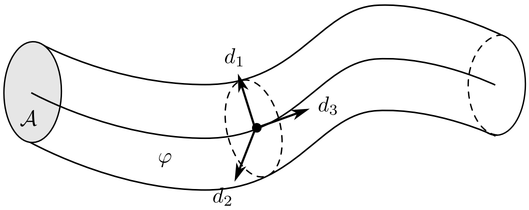


Fig. 3. A configuration of a flexible 1D component.

(locally) optimal under influence of gravity by tuning a set of design variables.

If `LocalRefiner` does not find a feasible configuration, `NominalRouter` is restarted with a revised set of routing parameters.

3. Nominal routing

3.1. Kinematics

A configuration q of a flexible 1D component of length L can – under the assumption that its cross section is planar and rigid – be described by an arc length parametrized curve in $\mathrm{SE}(3) = \mathbb{R}^3 \times \mathrm{SO}(3)$;

$$q : [0, L] \ni s \mapsto (\varphi(s), R(s)) \in \mathrm{SE}(3). \tag{1}$$

Here, $R = (d_1, d_2, d_3) \in \mathrm{SO}(3)$ describes the evolution of the cross section orientation along the centre curve φ (see Fig. 3).

The volumetric shape $\mathcal{S}(q)$ of the component can be traced via the deformation mapping $(s, \xi_1, \xi_2) \mapsto \varphi(s) + \xi_1 d_1(s) + \xi_2 d_2(s)$, for material coordinates $(s, \xi_1, \xi_2) \in [0, L] \times \mathcal{A}$, where $\mathcal{A} \in \mathbb{R}^2$ is the cross sectional domain.

The curvature k of φ is written

$$k(s) = \|\partial_s^2 \varphi(s)\| \tag{2}$$

and the bending radius (or radius of curvature) r is defined as

$$r(s) = \frac{1}{k(s)}. \tag{3}$$

3.2. Problem description

The nominal routing problem for flexible 1D components of the form (Eq. (1)) is to find a nominal configuration q_0 that is collision-free,

$$\mathcal{S}(q_0) \cap \mathcal{W} = \emptyset, \tag{4}$$

and connects the two connection points $Q_S = (p_S, R_S)$ and $Q_G = (p_G, R_G) \in \mathrm{SE}(3)$,

$$q_0(0) = Q_S \tag{5}$$

$$q_0(L) = Q_G, \tag{6}$$

and is feasible with respect to the geometric functional and manufacturing constraints,

$$h_m(q_0) := \begin{pmatrix} r - r_{\min} \\ l - l_{\min} \end{pmatrix} \geq 0. \tag{7}$$

Here, the constraints are manifested as a minimum allowed bending radius $r_{\min}$ and a minimum allowed straight length $l_{\min}$. Furthermore, r is the smallest measured bending radius $r(s)$ along $[0, L]$ and l is the length of the smallest connected interval on $[0, L]$ where $k(s) \equiv 0$ (i.e. the smallest straight segment). Hence, locally the trajectory of a feasible configuration always lies inside the reachable volume as illustrated in Fig. 4(a).

A configuration q_0 is said to be optimal if it minimizes the cost functional π defined as a weighted sum of different cost terms penalizing component length and placement,

$$\pi(q_0; w_L, w_\phi) = w_L \int_{s=0}^{L} ds + w_\phi \int_S \phi dV. \tag{8}$$

Here, w_L and w_ϕ are non-negative weights for the different cost terms. Also, $\phi : \mathbb{R}^3 \mapsto \mathbb{R}^+$ is a general cost field giving preference to certain areas in space. In this article, ϕ is a decreasing function over the distance field with respect to $\mathcal{W}$,

$$\phi(\cdot) := \frac{1}{1 + \mathrm{dist}(\cdot, \mathcal{W})^2}. \tag{9}$$

3.3. Restrictions

The component is assumed to have a circular cross sectional profile and a configuration curve where the tangent is always normal to the cross section, $\partial_s \varphi \equiv d_3$. For components of this type, the torsional twist is redundant for the shape of the component.[2] Hence, the configuration space we operate in can be reduced from $\mathrm{SE}(3)$ to $\mathbb{R}^3 \times S^2$.

Moreover, we assume that the nominal configuration curve has a piecewise constant bending radius. Thus, all feasible configuration curves can be represented by a continuously differentiable sequence of construction segments of the form

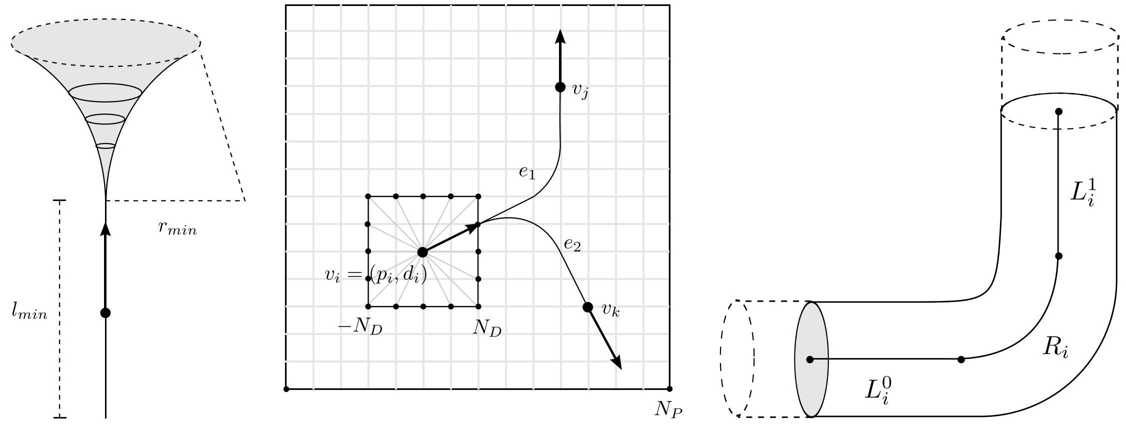
$$C = \{C_0, C_1, C_2, \ldots, C_{N-1}, C_N\}. \tag{10}$$

A construction segment C_i is either a straight line segment L_i or a concatenation of two straight line segments L_i^0 and L_i^1 of length $\geq l_{\min}/2$ with a circular arc R_i of bending radius $\geq r_{\min}$ in between;

$$C_i := \begin{cases} L_i & \text{if straight,} \\ (L_i^0, R_i, L_i^1) & \text{otherwise.} \end{cases} \tag{11}$$

A connected sequence of construction segments on this form is guaranteed to satisfy the manufacturing constraints (Eq. (7)) provided that the sequence starts and ends with straight segments L_0 and L_N of length $\geq l_{\min}/2$.

[2] The torsional twist needed to satisfy the boundary conditions (Eqs. (5) and (6)) can be distributed as a post-processing step.



(a) The reachable volume. (b) The uniform grid. (c) A construction segment.

Fig. 4. Illustrations of the structure of the routing roadmap.

The cost functional π can now be expressed as a discrete sum of cost terms evaluated for the respective construction segments;

$$\pi(C; w_L, w_\phi, w_N) = \sum_{i=0}^{N} \left(\pi(C_i, w_L, w_\phi) + w_N \right). \qquad (12)$$

Here, w_N is an added weight that also assigns a penalty to the discrete number of construction segments N.

3.4. The routing roadmap

Let $G = (V, E)$ denote a routing roadmap over $\mathbb{R}^3 \times S^2$ consisting of a set of nodes V and a set of connecting edges E. The relatively low dimension of the configuration space allows for distribution of V on a uniform grid; $V := V_P \times V_D$. The *spatial grid* $V_P \subset \mathbb{R}^3$ is a uniform rectangular grid with a spatial resolution N_P that encloses the work space region. The *directional grid* $V_D \subset S^2$ consists of the unit directions that connect points in V_P with a certain hit resolution N_D.

The roadmap edge set E is generated by connecting nodes in V with feasible and collision-free construction segments of the form (Eq. (11)). Hence, every edge $e_i \in E$ is associated with a feasible construction segment $e_i \leftrightarrow C_i$ and with edge cost $\pi(C_i, w_L, w_\phi) + w_N$. The bending radius of the construction segments is chosen from a discrete set of allowed bending radii, providing a natural coupling between the functional and manufacturing constraints h_m and the local connectivity of the roadmap. In fact, a connectivity stencil E_0 can be generated once and super positioned to help generate the set of all feasible edges E_i from a given node $v_i \in V$.

3.5. Roadmap search strategies

Different strategies can be employed when searching a roadmap for a path of lowest cost between a start node v_S and a goal node v_G. In this article we will evaluate the following basic strategies.[3]

Dijkstra's algorithm. A candidate set of nodes is maintained and expanded by extracting the candidate node with the least accumulated cost so far and then pushing its neighbouring nodes to the candidate set. The search starts by pushing v_S to the candidate set and finishes as soon as v_G is extracted. If the search is continued until the candidate set is empty, the optimal paths to all reachable nodes are generated.

A^* search. The extraction candidate is chosen using a heuristic function that guides the search by estimating the cost from the node to v_G. If the heuristic function is admissible (i.e. never overestimates) and monotonic (i.e. satisfies the triangle inequality on V), then it is guaranteed that the optimal solution is found when v_G is extracted. The heuristic can sometimes be improved by introducing and relating to a set of *landmark nodes*, for which the heuristic function gives a better estimate. For example, the landmark nodes could correspond to nodes in a coarser roadmap where the optimal solution to the goal is known.

Bi-directed search. Two candidate sets are initiated with v_S and v_G respectively. The sets are then alternatively expanded until they meet in a node that has been extracted from both sets.

The sets of encountered candidates for each strategy are illustrated in Fig. 5.

3.6. Algorithm description

The main idea behind the `NominalRouter` algorithm is to first do an exhaustive search on a roadmap over V_P in order to

- exclude regions of non-interest and
- generate a heuristic cost field

as a pre-processing step to allow for an efficient exploration of the full routing roadmap G. This pre-processing step will from now on be referred to as the $\mathbb{R}^3$ sweep.

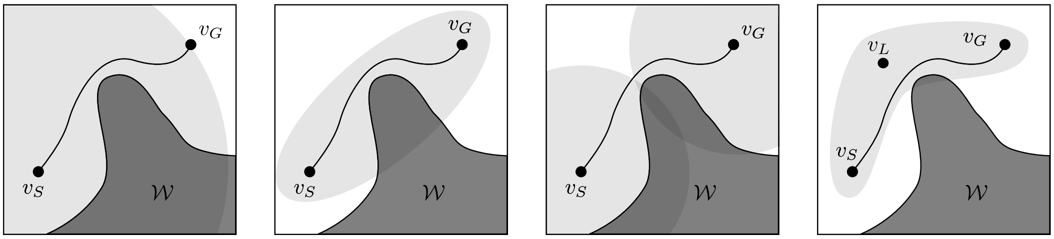
3.6.1. $\mathbb{R}^3$ sweep

Let $G_P = (V_P, E_P)$ be a roadmap over the spatial grid V_P where the edge set E_P connects each node in V_P to the closest nodes in the canonical and diagonal directions with a collision-free straight segment. A sweep of Dijkstra's algorithm on G_P from a start node generates a tree of optimal paths to all reachable nodes in V_P. These optimal paths in $\mathbb{R}^3$ do not consider the manufacturing constraints but do provide a hint of the location of collision-free passages. Let π_S and π_G be the cost fields corresponding to the optimal path trees from p_S and p_G respectively. Clearly, if a solution exists, the optimal path between p_S and p_G has the cost $\pi^* := \pi_S(p_G) = \pi_G(p_S)$. The cost field $\pi_{SG} : p \mapsto \pi_S(p) + \pi_G(p)$ then gives the cost of the optimal path between p_S and p_G that contains p (if one exists).

3.6.2. Routing roadmap search

Given the full routing roadmap $G = (V, E)$ and a cost field $\pi_{SG} : V_P \mapsto \mathbb{R}$ generated from an $\mathbb{R}^3$ sweep over V_P, we introduce a non-negative exclusion factor μ in order to discard roadmap regions

[3] See e.g. [29] for a more detailed description.



(a) Dijkstra.

(b) A^*.

(c) Bi-directed.

(d) Landmark-guided.

Fig. 5. Roadmap exploration by different search strategies.

that are unlikely to contain the optimal solution but at the same time allocate room for fulfilling the manufacturing constraints. The node set

$$V_\mu := \{(p, d) \in V : \pi_{SG}(p) \leq (1 + \mu)\pi^*\} \tag{13}$$

then contains all nodes in V for which the cost of the optimal path passing through is at most μ times larger than π^*.

Let E_μ be the edge subset of E that connects the nodes in V_μ. A roadmap search is carried out on $G_\mu := (V_\mu, E_\mu)$ based on the A^* strategy using π_G as the heuristic function over V_μ. If a solution cannot be found, then μ is gradually increased to extend the roadmap search region and the search is repeated until a solution is found or no exclusion occurs.

Algorithm 1 The `NominalRouter` algorithm

```
function NominalRouter($G, Q_S, Q_G$)
    $\pi_S \leftarrow$ search($G_P, p_S$)
    $\pi_G \leftarrow$ search($G_P, p_G$)
    for all $v \in V_P$ do
        if $\pi_S(v) \neq \infty$ and $\pi_G(v) \neq \infty$ then
            $\pi_{SG}(v) \leftarrow \pi_S(v) + \pi_G(v)$
        else $\pi_{SG}(v) \leftarrow \infty$
        end if
    end for

    $\mu \leftarrow 0.1$
    repeat
        $V_\mu := \emptyset$
        for all $v \in V$ do
            if $\pi_{SG}(v) < (1 + \mu)\pi_S(p_G)$ then
                $V_\mu \leftarrow v$
            end if
        end for
        $Q \leftarrow$ search($V_\mu, E(V_\mu), Q_S, Q_G$)
        if $Q \neq \emptyset$ then return $Q$
        end if
        $\mu \leftarrow \mu + 0.1$
    until $V_\mu = \{v \in V : \pi_{SG}(v) < \infty\}$
return $\emptyset$
end function
```

3.6.3. *Implementation notes*

Collision structures. A crucial bottleneck in the roadmap search is verifying whether or not an edge $e_i \in E$ is collision free (i.e. checking if $S(C_i) \cap \mathcal{W} = \emptyset$). Pre-computed collision structures can speed up this verification significantly. In our implementation, bounding volume hierarchies as described in [34] are used for checking collision and computing shortest distance between the construction segments in the connectivity stencil and tessellated representations of $\mathcal{W}$ with potentially millions of triangles.

Search tree maintenance. When maintaining the candidate set of nodes in the roadmap search, a heap is often used in order to quickly extract the best current candidate and update the cost estimate of its adjacent nodes. By storing and updating the node's position on the heap, adjacent nodes already placed on the heap can be looked up in $\mathcal{O}(1)$ time.

Parallel computing. The computations of cost fields π_S and π_G in the $\mathbb{R}^3$ sweep can be executed in parallel to increase computational performance.

3.6.4. *Discussion*

One should carefully note that in using the $\mathbb{R}^3$ sweep procedure there is trade-off between computational performance and optimality if the structure of the optimal solution set to G_P differs from that of G. If a solution is found on G_μ, then it is by necessity also feasible on G but not necessarily optimal on G due to the exclusion. Also, using π_G as the heuristic function in the A^* search strategy this can also produce suboptimal solutions, since it is not always admissible on V.

4. Local refinement

A configuration q_0 produced by the nominal routing algorithm provides

- an initial collision-free route of the component (placement), and
- a suggestion for the nominal design of the component (preformation).

However, due to gravity and internal stress, the flexible component will assume a deformed configuration q in mechanical equilibrium that might violate the functional constraints and/or no longer be collision-free. With a simulation model and refinement with respect to a set of design variables, a feasible and locally optimal configuration of the component in mechanical equilibrium can be recovered.

4.1. *Simulation model*

The representation of a flexible 1D component as a framed curve in Eq. (1) is commonly referred to in continuum mechanics as a *rod*. *Cosserat rod theory* [35] accounts for elastic deformations in the form of both shearing, stretching, bending and torsion.

The elastic potential energy density of a rod is composed of quadratic forms in terms of frame invariant *strain measures*. The shearing/stretching strain $\Gamma = (\Gamma_1, \Gamma_2, \Gamma_3)$ and curvature/torsion strain $\Omega = (\Omega_1, \Omega_2, \Omega_3)$ in material coordinates read

$$\Gamma(s) = R(s)^T \partial_s \varphi(s) - e_3, \tag{14}$$

$$\hat{\Omega}(s) = R(s)^T \partial_s R(s). \tag{15}$$

Here, $\Gamma_{1,2}$ are the shearing strain components in the d_1 and d_2 directions and Γ_3 is the tension strain, whereas $\Omega_{1,2}$ are the bending curvature strain components and Ω_3 is the torsion strain. Let Γ_0 and Ω_0 be the strain vectors for the nominal configuration q_0. With a hyper-elastic constitutive law (which can be justified since locally $\Gamma - \Gamma_0$ and $\Omega - \Omega_0$ are small), the stored shearing/stretching and bending/torsion energy densities read

$$w^{\Gamma}(s) = \frac{1}{2}(\Gamma(s) - \Gamma_0(s))^T K^{\Gamma}(\Gamma(s) - \Gamma_0(s)), \tag{16}$$

$$w^{\Omega}(s) = \frac{1}{2}(\Omega(s) - \Omega_0(s))^T K^{\Omega}(\Omega(s) - \Omega_0(s)), \tag{17}$$

for some effective stiffness tensors K^{Γ} and K^{Ω}.

In the presence of a gravitational force field with field constant g, the total potential energy is then

$$W = \int_{s=0}^{L} w^{\Gamma}(s) + w^{\Omega}(s) - K^{\rho} g^T \varphi(s) ds. \tag{18}$$

According to the Hamiltonian principle, the static mechanical equilibrium is characterized as a stationary point to the total potential energy,

$$\delta W = 0. \tag{19}$$

4.1.1. Contact forces

Friction-less contact forces can be included in the model by adding a repelling potential energy W^C. Hertz contact theory [36] suggests a potential energy density of the form $w^C(s) = K^C d(s)^{5/2}$, where d is the penetration depth of $S(q)$ at s with respect to $\mathcal{W}$, so that $W^C = \int_{s=0}^{L} w^C(s) ds$. Eq. (18) is then extended to

$$\delta \left(W + W^C \right) = 0. \tag{20}$$

4.2. Problem description

The local refinement problem is to ensure that, when a component with a nominal configuration q_0 of the form (Eq. (10)) is held at Q_S and Q_G and influenced by gravity, the deformed configuration q is still collision free,

$$\mathcal{S}(q) \cap \mathcal{W} = \emptyset, \tag{21}$$

and feasible with respect to the geometric functional constraints,

$$h_f(q) := (r - r_{\min}) \geq 0. \tag{22}$$

It is also desirable that q is still good with respect to the cost functional $\pi(q)$ (Eq. (8)).

The tunable design variables x can in principle be any design property of the component. In this article, a scaling[4] of the nominal configuration length L and the twist orientation at the connection points Q_S and Q_G are considered. Other choices of x could be the cross section radius, the individual construction segment arc lengths or even the nominal configuration strains (Γ_0, Ω_0) and the material parameters $(K^{\Gamma}, K^{\Omega}, K^{\rho})$.

Given an initial configuration q_0 and a set of design variables x, the simulation model in Section 4.1 can now be used to compute the corresponding deformed configuration $q(x)$ in mechanical equilibrium (with $\Gamma_0 = \Gamma(q_0)$ and $\Omega_0 = \Omega(q_0)$, if q_0 is treated as the nominal configuration of the component).

4.3. Algorithm description

The LocalRefiner algorithm seeks a set of design variables x that makes the deformed configuration $q(x)$ feasible and (locally) optimal with respect to the weighted cost functional π. In order to treat it as an unconstrained minimization problem, the geometric functional constraints h_f are enforced by adding a Lagrangian term penalizing deformed configurations with infeasible bending radii with a corresponding non-negative weight w_r. The objective functional f to minimize is then written

$$f(x; w_L, w_\phi, w_r) = \pi(q(x); w_L, w_\phi) + w_r \int_{s=0}^{L(x)} \min\{0, h_f(q(x))\}^2 ds. \tag{23}$$

To minimize f, we employ the *Nelder–Mead algorithm* [37]. It is a gradient-free iterative solver for unconstrained non-linear optimization problems and thus suitable for coupling with the simulation model. Also, note that f is not continuously differentiable due to the last penalty term in Eq. (23) and the distance field ϕ in Eq. (9), further motivating the choice of a gradient-free solver.

Finally, we observe that a nominal configuration q_0 is already in a stress-free mechanical equilibrium in the absence of gravity. By initially turning off the gravity field and then gradually increasing it during the optimization, the distance field ϕ is given more room to influence the solution. Also, the solution q_0 from NominalRouter can give a qualified hint on where to introduce a single or multiple supplementary clips to further constrain the component.

4.3.1. Implementation notes

For an efficient computation of the mechanical equilibrium q in each evaluation of f in the Nelder–Mead algorithm, the discrete strain measures in the simulation model are based on non-linear geometric finite differences and an efficient quasi-Newton solver is used to find a stationary point to the discrete energy functional.[5] Furthermore, since the cross section of the component is assumed to be symmetric, the penetration depth in the contact force potential (Eq. (20)) can be computed from the shortest distance to the centre curve, which in turn can be computed as in [40].

4.4. Feedback

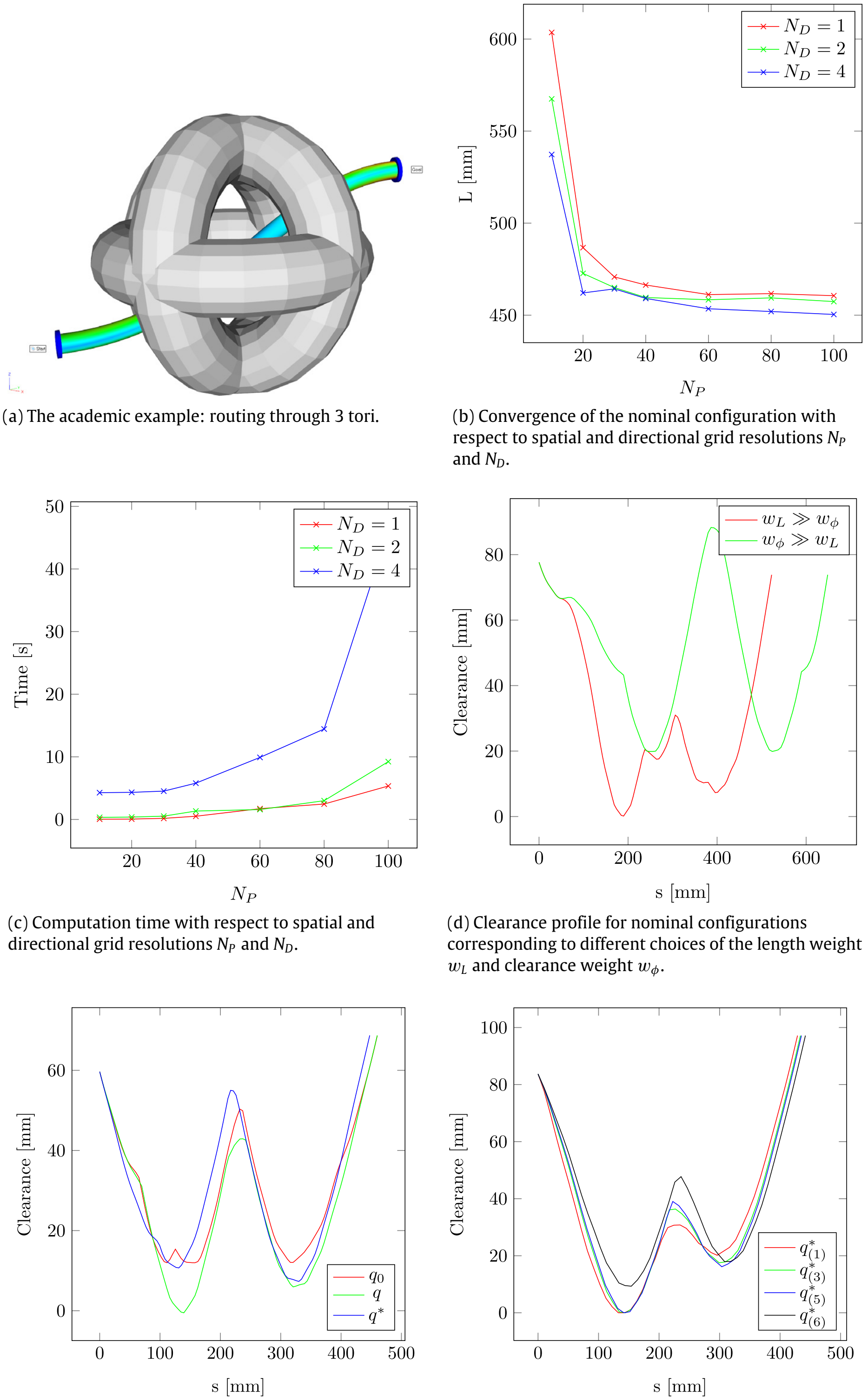
After LocalRefiner, it is still possible that the flexible component violates the functional constraints h_f and is held collision-free by undesired contact forces. This can be resolved by restarting NominalRouter with a revised set of routing parameters; for instance, if the component is in contact with $\mathcal{W}$ the clearance weight w_ϕ can be tuned up; if the minimum bending radius is too small, the length weight w_L can be tuned down and the bending weight w_B and the minimum allowed bending radius $r_{\min}$ can be increased. In this article, we suggest the simple update scheme in Algorithm 2 with a given weight increment/decrement parameter $\epsilon > 0$. The update is run until the incremented weights become larger than a given threshold of 1.

Algorithm 2 The feedback update

```
if $h_f(q^*) \ngeq 0$ then
    $w_r \leftarrow w_r + \epsilon$
    $w_L \leftarrow \max\{0, w_L - \epsilon\}$
else if $W^C(q^*) \neq 0$ then
    $w_\phi \leftarrow w_\phi + \epsilon$
else
    $q^*$ acceptable
end if
```

[4] A scaling λL by a factor $\lambda > 0$ is assigned by scaling the arc lengths of the individual construction segments by λ.

[5] See e.g. [38,39] for details concerning an efficient implementation of such a simulation model.



(a) The academic example: routing through 3 tori.

(b) Convergence of the nominal configuration with respect to spatial and directional grid resolutions N_P and N_D.

(c) Computation time with respect to spatial and directional grid resolutions N_P and N_D.

(d) Clearance profile for nominal configurations corresponding to different choices of the length weight w_L and clearance weight w_ϕ.

(e) Clearance profile for the nominal configuration and the configuration of the component before and after local refinement (q_0, q and q^*).

(f) Clearance profile for the deformed configuration of the component after feedback loop iterations 1, 3, 5 and 6.

Fig. 6. The academic example: routing of a flexible 1D component through three tori.

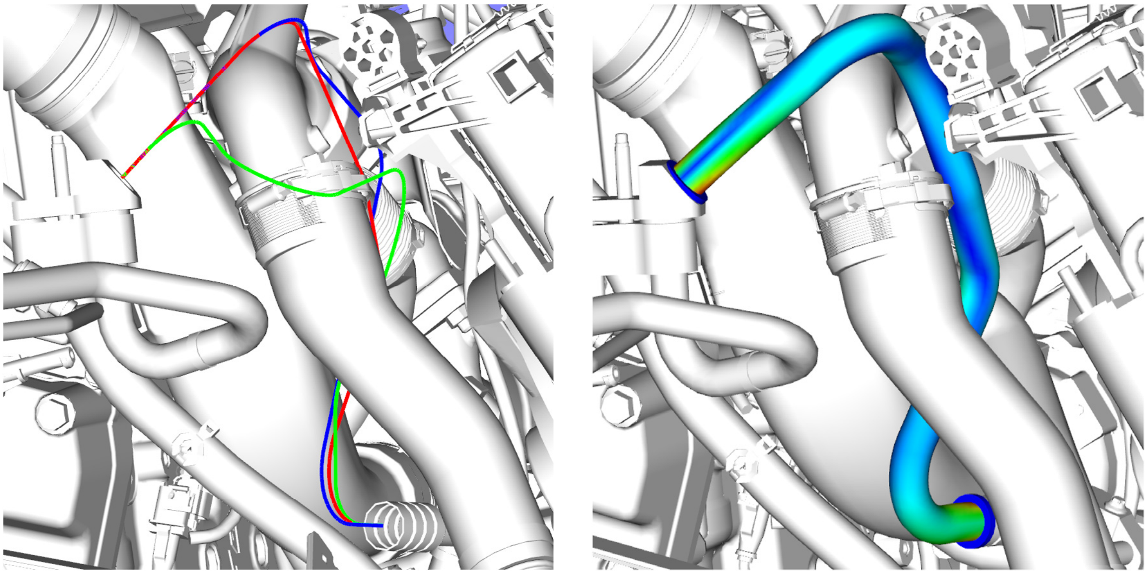
5. Results

The method is tested and validated on two scenarios: an academic example that involved routing a flexible 1D component through 3 tori (Fig. 6(a)) and an industrial example that involves routing an aluminium ducting hose (Fig. 7(b)). The tests are structured as follows:

First, NominalRouter is evaluated with respect to different roadmap search strategies and the influence of the $\mathbb{R}^3$ sweep is investigated. Secondly, nominal configurations are generated for

Table 1
NominalRouter with different roadmap search strategies. N_E and N_I are the number of heap extractions and the heap size at termination respectively. The time for the sweep procedure was measured to 0.247 s.

Strategy	Sweep	Heuristic function	Total time (s)	Arc length (mm)	N_E	N_I
Dijkstra	No	–	89.232	465.94	115 071	28 728
Dijkstra	Yes	–	10.312	465.94	21 131	4 502
Bi-Dijkstra	Yes	–	6.255	465.94	14 176	16 254
A^*	No	$\|\cdot\|$	0.967	465.94	358	11 583
A^*	Yes	$\|\cdot\|$	0.515	465.94	341	6 170
A^*	Yes	π_G	0.353	466.43	514	4 621
Bi-A^*	Yes	π_S, π_G	0.578	468.68	768	11 687



(a) Nominal configurations of an AC hose routed with different priorities (unspecified weights are 0). (red): length ($w_L > 0$), (green): clearance ($w_\phi > 0$), (blue): number of turns ($w_N > 0$).

(b) Locally refined flexible AC hose routed with length priority ($w_L > 0$). The total computation time was 7.4 s.

Fig. 7. The industrial example: routing of an aluminium ducting hose. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Source: Courtesy of NEVS.

different sets of weights. Finally, the effect of LocalRefiner is analysed with respect to clearance. All computations are done on a standard desktop PC with an Intel Core i5 2.50 GHz CPU.

Table 1 shows results from solving an instance of the academic example with different roadmap search strategies. Each strategy is evaluated with and without the $\mathbb{R}^3$ sweep. The heuristic functions considered in the A^* search strategies are the Euclidean distance and the $\mathbb{R}^3$ sweep cost fields. In all encountered test instances, roadmap exclusion combined with the A^* strategy and heuristic function π_G is superior in terms of computational performance. This comes, as indicated Table 1, with a small trade-off in terms of optimality, since the heuristic function is monotonic but not always admissible (see Section 3.6.4).

The convergence of the nominal configuration produced by NominalRouter with respect to grid resolution is indicated in Fig. 6(b) by measuring the configuration arc length L. The corresponding computation times are shown in Fig. 6(c). In Fig. 6(d), the clearance profile along the nominal configuration is plotted for two different sets of cost function weights, showcasing that a higher desired clearance comes at the expense of a longer component. In Fig. 6(e), the clearance profile is plotted along the nominal configuration before and after gravitational influence and after local refinement. By automatically refining the component length and the twist at Q_G, the small clearance at $s = 120$ mm due to elastic deformation can be fended off. Fig. 6(f) reflects an instance of the problem where multiple feedback updates are required in order to resolve the undesired contact point introduced in local refinement. With $\epsilon = 0.1$ in Algorithm 2, a deformed configuration with clearance 9.32 mm is found in 6 iterations.

The industrial example consists of routing an aluminium ducting hose for the air conditioner (AC) in the engine compartment of a car, courtesy of NEVS [41]. The surrounding consists of 1.2 million triangles. In Fig. 7(a), it can be seen that different sets of cost function weights yield different global nominal configurations. In Fig. 7(b) the hose corresponding to length priority is simulated in mechanical equilibrium. The computation time was 7.4 s for the routing algorithm and an additional 4.3 s for the preparation of the collision structures.

6. Conclusion and outlook

This article has presented a unifying method for routing of flexible one-dimensional components with geometric design constraints. In a segregated approach, a resolution complete A^* based grid search combined with a lower dimensional sweep procedure first generates feasible nominal configurations. Secondly, a simulation model based on Cosserat rod theory is coupled to the Nelder–Mead algorithm to ensure that the flexible component is still feasible in mechanical equilibrium by locally adjusting a set of tunable design variables. Test results show that the method is able to efficiently generate global solutions for different objectives for industrial scenarios involving complex geometries and real constraints.

To extend upon this work in the future there are several possibilities:

- The method should be extended to flexible 1D components with multiple branching and break out locations and non-circular cross sections.
- The design of the feedback update in Section 4.4 should be investigated more in detail and the simple scheme in Algorithm 2 could potentially be improved.
- As mentioned in Section 4.3, it should be looked into how to gradually increase the gravity field during local refinement to avoid local minima where the configuration is in contact.
- The cost field ϕ defined in Eq. (9) should be extended to give preference to more specific areas of space, e.g. favouring closeness to a certain surface. Also, additional via points could be introduced in order to further help the designer guide the routing through preferred regions.
- The structure of the uniform grid sometimes permits optimal nominal configurations with an unnecessary high number of turns if the grid is too coarse. This issue can certainly be resolved by increasing the grid resolution, however a smoothing of the nominal configuration in a post-processing step could be an alternative.

Acknowledgements

This work was carried out within the FFI Sustainable Production Technology program, the Wingquist Laboratory VINN Excellence Centre, the Sustainable Production Initiative and the Production Area of Advance at Chalmers University of Technology. It was supported by the Swedish Governmental Agency for Innovation Systems grant number 2009-01594.

References

[1] Lee CY. An algorithm for path connections and its application. IRE Trans Electron Comput 1961;10:346–64.
[2] Ito T. A genetic algorithm approach to pipe route path planning. J Intell Manuf 1999;10(1):103–14.
[3] Zhu D, Latombe J. Pipe routing—path planning (with many constraints). In: Proceedings—1991 IEEE international conference on robotics and automation. 1991. p. 1940–7.
[4] Guirardello R, Swaney R. Optimization of process plant layout with pipe routing. Comput Chem Eng 2005;30:99–114.
[5] Wang C, Liu Q. Projection and geodesic based pipe routing algorithm. IEEE Trans Autom Sci Eng 2011;8(3):641–5.
[6] Ng FM, Ritchie JM, Simmons JEL, Dewar RG. Designing cable harness assemblies in virtual environments. J Mater Process Technol 2000;107(1–3):37–43.
[7] Srinivasan R, Patnaik LM. Two algorithms for three-layer channel routing. Comput-Aided Des 1984;16(5):264–71.
[8] West NA, Dwolatzky B, Meyer AS. Terrain based routing of distribution cables. IEEE Comput Appl Power 1997;10(1):42–6.
[9] Conru AB. A genetic approach to the cable harness routing problem. In: Proceedings—first IEEE conference on evolutionary computation. 1994. p. 200–5.
[10] Canny JF. The complexity of robot motion planning. Cambridge, MA: MIT Press; 1988.
[11] Kavraki L, Svestka P, Latombe J-C, Overmars MH. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans Robot Autom 1996;12(4):566–80.
[12] Bohlin R, Kavraki L. Path planning using lazy prm. In: Proceedings—IEEE international conference on robotics and automation. 2000. p. 521–8.
[13] Lavalle S. Rapidly-exploring random trees: a new tool for path planning. Technical report. 1998.
[14] Barraquand J, Latombe JC. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. Algorithmica 1993;10(2–4):121–55.
[15] Bohlin R. Path planning in practice: Lazy evaluation on a multi-resolution grid. In: Proceedings—IEEE international conference on intelligent robots and systems. 2001. p. 49–54.
[16] LaValle S. On the relationship between classical grid search and probabilistic roadmaps. Int J Robot Res 2004;23(7–8):673–92.
[17] Likhachev M, Ferguson DI, Gordon GJ, Stentz A, Thrun S. Anytime dynamic a*: An anytime, replanning algorithm. In: Proceedings—2005 international conference on automated planning and scheduling, ICAPS. 2005. p. 262–71.
[18] Dubin LE. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. Amer J Math 1957;79(3):497–516.
[19] Reeds JA, Shepp LA. Optimal paths for a car that goes both forwards and backwards. Pacific J Math 1990;145(2):367–93.
[20] Lamiraux F, Laumond JP. Smooth motion planning for car-like vehicles. IEEE Trans Robot Autom 2001;17(4):498–501.
[21] LaValle S. Randomized kinodynamic planning. Int J Robot Res 2000;20(5):378–400.
[22] Karaman S, Frazzoli E. Sampling-based optimal motion planning for non-holonomic dynamical systems. In: 2013 IEEE international conference on robotics and automation, ICRA. 2013. p. 5041–7.
[23] Kuffner J, LaValle S. Space-filling trees: A new perspective on motion planning via incremental search. In: Proceedings—2011 IEEE international conference on intelligent robots and systems, 2011.
[24] Webster RJ. Nonholonomic modeling of needle steering. Int J Robot Res 2006;25(5–6):566–80.
[25] Patil S, Alterovitz R. Interactive motion planning for steerable needles in 3D environments with obstacles. In: 2010 3rd IEEE RAS and EMBS international conference on biomedical robotics and biomechatronics, BioRob. 2010. p. 893–9.
[26] Latombe JC. Robot motion planning. Springer; 1991.
[27] Laumond JP, Sekhavat S, Lamiraux F. Guidelines in nonholonomic motion planning for mobile robots. Springer; 1998.
[28] Choset H, Lynch K, Hutchinson S, Kantor G, Burgard W, Kavraki L, et al. Principles of robot motion: theory, algorithms, and implementations. 2005.
[29] LaValle S. Planning algorithms. New York, NY: Cambridge University Press; 2006.
[30] Bretl T, McCarthy Z. Equilibrium configurations of a kirchhoff elastic rod under quasi-static manipulation. In: Algorithmic foundations of robotics X, Vol. 88. 2013. p. 71–87.
[31] Moll M, Kavraki L. Path planning for deformable linear objects. IEEE Trans Robot 2006;22(3):625–36.
[32] Kabul I, Gayle R, Lin M. Cable route planning in complex environments using constrained sampling. In: Proceedings—2007 ACM symposium on solid and physical modeling. 2007. p. 385–402.
[33] Hermansson T, Bohlin R, Carlson JS, Söderberg R. Automatic assembly path planning for wiring harness installations. J Manuf Syst 2013;32(3):417–22.
[34] Larsen E, Gottschalk S, Lin MC, Manocha D. Fast distance queries with rectangular swept sphere volumes. In: Proceedings—2000 international conference on robotics and automation, ICRA, Vol. 4. 2000. p. 3719–26.
[35] Simo JC. A finite strain beam formulation, the three-dimensional dynamic problem, part i. Comput Methods Appl Mech Engrg 1985;49:55–70.
[36] Hertz H. Miscellaneous papers. London: McMillan; 1896.
[37] Nelder JA, Mead R. A simplex method for function minimization. Comput J 1965;7(4):308–13.
[38] Gregoire M, Schömer E. Interactive simulation of one-dimensional flexible parts. Comput-Aided Des 2007;39(8):694–707.
[39] Linn J, Stephan T, Carlson JS, Bohlin R. Fast simulation of quasistatic rod deformations for vr applications. In: Progress in industrial mathematics at ECMI 2006, Vol. 12. 2008. p. 247–53.
[40] Shellshear E, Bitar F, Assarsson U. Pdq: Parallel distance queries for deformable meshes. Graph Models 2013;75(2):69–78.
[41] National Electric Vehicle Sweden AB, 2015. http://www.saabcars.com/ [Online; accessed 15.04.15].