



**HAL**  
open science

# Real-Time Simulation For Control Of Soft Robots With Self-Collisions Using Model Order Reduction For Contact Forces

Olivier Goury, Bruno Carrez, Christian Duriez

► **To cite this version:**

Olivier Goury, Bruno Carrez, Christian Duriez. Real-Time Simulation For Control Of Soft Robots With Self-Collisions Using Model Order Reduction For Contact Forces. IEEE Robotics and Automation Letters, 2021, 6 (2), pp.3752 - 3759. 10.1109/LRA.2021.3064247 . hal-03192762

**HAL Id: hal-03192762**

**<https://inria.hal.science/hal-03192762v1>**

Submitted on 8 Apr 2021

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

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

# Real-Time Simulation For Control Of Soft Robots With Self-Collisions Using Model Order Reduction For Contact Forces

Olivier Goury<sup>1,2</sup>, Bruno Carrez<sup>1,2</sup> and Christian Duriez<sup>1,2</sup>

**Abstract**—In rigid robotics, self-collision are usually avoided since it leads to a failure in the robot control and can also cause damage. In soft robotics, the situation is very different, and self-collisions may even be a desirable property, for example to gain artificial stiffness or to provide a natural limitation to the workspace. However, the modeling and simulation of self-collision is very costly as it requires first a collision detection algorithm to detect where collisions occur, and most importantly, it requires solving a constrained problem to avoid interpenetrations. When the number of contact points is large, this computation slows down the simulation dramatically. In this paper, we apply a numerical method to alleviate the contact response computation by reducing the contact space in a low-dimensional positive space obtained from experiments. We show good accuracy while speeding up dramatically the simulation. We apply the method in simulation on a cable-actuated finger and on a continuum manipulator performing exploration. We also show that the reduced contact method proposed can be used for inverse modeling. The method can therefore be used for control or design.

**Index Terms**—Modeling, Control, and Learning for Soft Robots; Model Order Reduction; Self-collision

## I. INTRODUCTION

SOFT robotics has recently emerged as a field of its own within general robotics. Made of soft materials, these robots have all kind of properties that can be beneficial in manipulation task, manoeuvring in constrained environments, etc... One characteristic of soft robots is their interaction with the environment which may not be avoided as is often the case in rigid robotics, but rather used as a benefit for control of force application.

Self-collisions are collisions between different parts of the same robot. In the field of articulated rigid robotics, such as humanoid robotics, self-collisions are banned, typically for safety reasons, since self-collisions can result in damage to themselves or their environment. In some cases, self-collision avoidance is solved easily by applying a limitation in the maximal angle of the motors at each

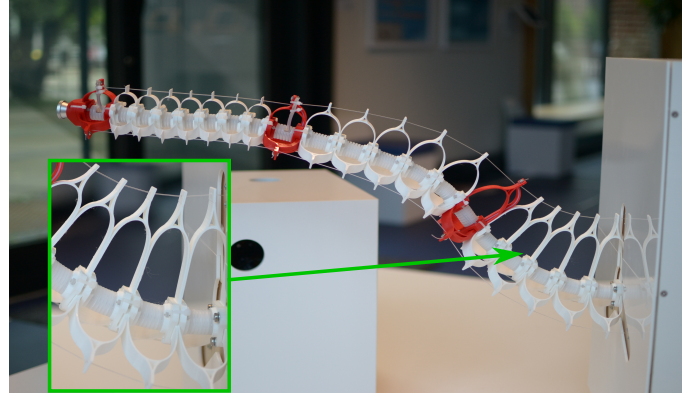


Fig. 1: Flexible manipulator robot where self-collisions play an important role. In this case, they allow the manipulator to stiffen which leads to more stable positioning.

joint. For more complex articulated robots, it is not trivial, and some special strategies have been applied [1], [2], [3], [4].

Conversely, in the field of soft robotics, one may not necessarily want to avoid self-collisions, where they might even be beneficial by allowing to stiffen the robot when it has to lift a load, for instance. However, self-collisions are much harder to detect in soft robots than in rigid robots, because the shape of the robot changes and there are many different scenarios where these collisions may happen.

### A. Contact handling in robotics and soft robotics

In robotics, contacts are particularly important in grasping applications, with rigid or soft fingers [5], [6], [7]. In these examples, an analytical model is used to approximate the contact forces. In particular, the surface of the objects in contacts are either forced to be a hemispheric or planar, or approximated by a quadratic function. See [8] for a review. When dealing with truly soft robots, the robots surface can undergo large deformations, and approximating the surface of contact becomes inaccurate. In [9], the robot was modeled using the FEM, and no assumption was made on the contact geometry. This strategy was good enough to treat the contacts in real-time. However, they are limitations when the number of contact points becomes large in the case of more complex geometries.

Manuscript received: October , 23th, 2020; Revised January, 18th, 2021; Accepted February, 20th, 2021.

This paper was recommended for publication by Editor Cecilia Laschi upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Inria

<sup>1</sup>DEFROST team, Inria Lille-Nord Europe, 40 avenue du Halley, 59000 Lille, France [olivier.goury@inria.fr](mailto:olivier.goury@inria.fr)

<sup>2</sup>University of Lille - CRISTAL UMR CNRS 9189, 59655 Villeneuve d'Ascq Cedex France

Code: <https://github.com/SofaDefrost/ModelOrderReduction>

Digital Object Identifier (DOI): see top of this page.

### B. Projection-based Model order reduction for expensive Finite Element simulations

The finite element method leads to an accurate prediction of material behaviour, at the expense of computationally expensive simulations. This is an issue that has been alleviated in the computational mechanics community using model order reduction in various applications from biomechanics, to aeronautics, fluids, fracture, multiscale simulations [10], [11], [12], [13], [14], [15]. The principle of the method is to define the vector position (or velocity) of the material as a linear combination of elements of a precomputed basis of small dimension that captures all the possible movements of the object up to a certain tolerance. That basis is computed from a database using the Proper Orthogonal Decomposition (POD). Alternative ways to compute the basis using vibrations modes was used in computer graphics, for example in [16].

In the context of soft robotics, it was successfully applied on the framework SOFA in [17]. In [18], it was applied to reduce the state space to a manageable size for dynamic control.

When involving contacts, the computer graphics community has also been very active in developing very efficient reduced models using penalty methods or impulses and sometimes optimising collision detection [19], [20], [21], [22]. In those methods, the contact problem is typically treated by applying penalty forces in contact locations to avoid inter-penetrations, but the contact constraints are not solved using a contact law. In [23], an image-based formulation was proposed, allowing fast volume inter-penetration evaluation to apply a coarse contact constraint. In computational mechanics, methods were developed to reduce the space of the dual variables representing contact forces in [24]. The problem was also considered in [25]. The reduced space is forced to be made of a basis with positive vectors, to enforce the positiveness of the contact force. This basis is computed using a non-negative matrix factorisation (NNMF). We are going to follow a similar approach in this paper.

### C. Contribution

The contribution of this paper is the first real-time model for soft robot self-collision scenarios based on FEM and Linear Complementarity Problem (LCP) frictionless contact. We show that this reduced contact model can be used in inverse simulation on a simple soft finger example. We validate that it can run in real-time and be used for control. Then, we show the application on a real robot involving many self collisions that are captured by the reduced contact model.

## II. ROBOT EQUATION OF MOTION AND CONTACT: FROM RIGID TO SOFT

### A. Rigid robot

Consider a rigid robot in the environment. Its dynamics can be derived from Newton's second law:

$$\mathbf{M}_R(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_R(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_R(\mathbf{q}) = \mathbf{H}^T \boldsymbol{\lambda}, \quad (1)$$

$\mathbf{H}$	Contacts directions	$\boldsymbol{\lambda}$	Contacts magnitude
$\Phi$	Reduced motion matrix	$\boldsymbol{\alpha}$	Reduced motion state
$\Psi$	Reduced contact matrix	$\boldsymbol{\beta}$	Reduced contact state
$\delta$	Gap between objects	$\hat{\delta}$	Gap in the reduced space

TABLE I: Variables listing.

where  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$  is the robot configuration and its derivatives (for example joint angles in case of an articulated robot),  $\mathbf{M}_R(\mathbf{q})$  is the inertia matrix,  $\mathbf{C}_R(\mathbf{q}, \dot{\mathbf{q}})$  the centrifugal and Coriolis term, and  $\mathbf{G}_R(\mathbf{q})$  the gravity term.  $\mathbf{H}^T \boldsymbol{\lambda}$  contains the control inputs, as well as the contacts with the environment.

### B. Soft robot dynamics in a finite element setting

If we now assume the robot is soft, and that we use the finite element method (FEM) to discretise its continuous geometry, the equation (1) is modified in several ways.

- The configuration variables  $\mathbf{q}$  do not represent the rigid joints anymore but the displacement of each node in the FE mesh discretising the robot.
- A new term  $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$  accounting for internal forces arises from the deformation of the soft material in the robot
- The inertia matrix  $\mathbf{M}_R$  becomes the mass matrix accounting for a diffuse effect of mass along the soft material at each node.
- The centrifugal matrix  $\mathbf{C}_R(\mathbf{q}, \dot{\mathbf{q}})$  is removed since its effect is captured by the soft material.

Equation (1) hence takes the form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q})\mathbf{q} = \mathbf{H}^T \boldsymbol{\lambda}, \quad (2)$$

In this setting  $\mathbf{H}^T \boldsymbol{\lambda}$  represents the vector of contact forces, with  $\mathbf{H}^T$  a sparse matrix gathering contact directions, and  $\boldsymbol{\lambda}$  a vector of scalars gathering contact magnitudes. Note that this equation could also directly be obtained from applying Newton's second law on soft objects.

1) *Time discretization*: Let us now discretise the continuous time space in a discrete time interval  $[t_0, t_1, \dots, t_{n_t}]$ , with  $h = t_{n+1} - t_n$ .

Using an implicit Euler integration scheme, setting  $\mathbf{v} = \dot{\mathbf{q}}$ , we obtain:

$$\mathbf{M}(\mathbf{v}_{t_{n+1}} - \mathbf{v}_{t_n}) + h\mathbf{F}(\mathbf{q}_{t_{n+1}}, \mathbf{v}_{t_{n+1}}) + h\mathbf{g}_{t_{n+1}} = h\mathbf{H}^T \boldsymbol{\lambda} \quad (3)$$

$$\mathbf{q}_{t_{n+1}} = \mathbf{q}_{t_n} + h\mathbf{v}_{t_{n+1}}, \quad (4)$$

where  $\mathbf{g}_{t_{n+1}}$  is the body force at time  $t_{n+1}$  known a priori. The vector of internal forces  $\mathbf{F}(\mathbf{q}_{t_{n+1}}, \mathbf{v}_{t_{n+1}})$  is a non-linear function of  $\mathbf{q}$  and  $\mathbf{v}$ . Hence, we perform the following linearisation (which is a decent approximation over a small time interval, involving a small variation of the internal forces):

$$\mathbf{F}(\mathbf{q}_{t_{n+1}}, \mathbf{v}_{t_{n+1}}) = \mathbf{F}(\mathbf{q}_{t_n} + d\mathbf{q}, \mathbf{v}_{t_n} + d\mathbf{v}) \quad (5)$$

$$= \mathbf{f}_{t_n} + \frac{\delta \mathbf{F}}{\delta \mathbf{q}} d\mathbf{q} + \frac{\delta \mathbf{F}}{\delta \mathbf{v}} d\mathbf{v} \quad (6)$$

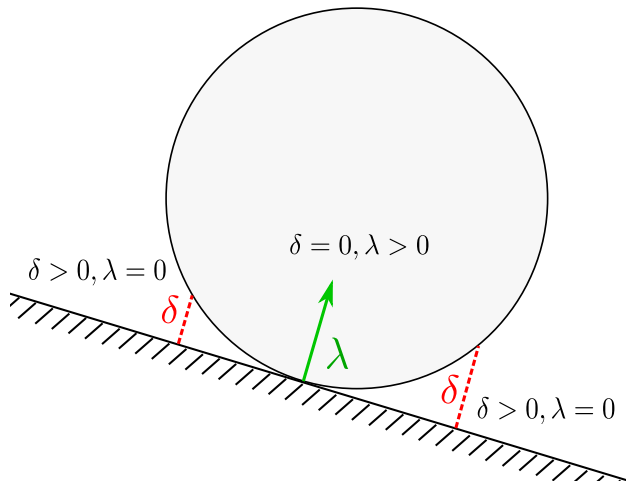


Fig. 2: Complementarity relation between contact force  $\lambda$  and gap  $\delta$  given by Signorini law.

Using  $d\mathbf{q} = \mathbf{q}_{t_{n+1}} - \mathbf{q}_{t_n} = h\mathbf{v}_{t_{n+1}}$  and  $d\mathbf{v} = \mathbf{v}_{t_{n+1}} - \mathbf{v}_{t_n}$ , we obtain:

$$\underbrace{\left( \mathbf{M} + h \frac{\delta \mathbf{F}}{\delta \mathbf{v}} + h^2 \frac{\delta \mathbf{F}}{\delta \mathbf{q}} \right) d\mathbf{v}}_{\mathbf{A}(\mathbf{q}_{t_n}, \mathbf{v}_{t_n})} = \underbrace{-h^2 \frac{\delta \mathbf{F}}{\delta \mathbf{q}} \mathbf{v}_{t_n} - h (\mathbf{g}_{t_{n+1}} + \mathbf{f}_{t_n})}_{\mathbf{b}(\mathbf{q}_{t_n}, \mathbf{v}_{t_n})} + h \mathbf{H}^T \lambda. \quad (7)$$

Finally, at each time step  $t_n$ , the equation can be written:

$$\mathbf{A}_{t_n} d\mathbf{v}_{t_{n+1}} = \mathbf{b}_{t_n} + h \mathbf{H}^T \lambda. \quad (8)$$

### C. Contact law

Now, we are introducing the Signorini law for contact. Consider two points, each on a surface of two distinct objects. The law states that there is a complementarity relation linking the gap  $\delta$  between these two points and the contact force  $\lambda$  they apply on each other (see Fig. 2):

$$0 \leq \delta \perp \lambda \geq 0. \quad (9)$$

In this paper, we consider frictionless contacts.

### D. Solving the constrained problem

At the beginning of each timestep, using the collision detection pipeline in SOFA, we know the value of the gap  $\delta$  for each potential contact point. We now want to solve equation (8) under the constraint of Signorini law (9). This turns out to be a Linear Complementarity Problem (LCP). The LCP can be solved in three steps:

- Solve the free motion ignoring the contact constraint ( $\lambda = 0$ ):

$$\mathbf{A}_{t_n} d\mathbf{v}_{t_{n+1}}^{\text{free}} = \mathbf{b}_{t_n}. \quad (10)$$

- Then the constraint law is linearised, and we write  $d\mathbf{v}_{t_{n+1}} = d\mathbf{v}_{t_{n+1}}^{\text{free}} + d\mathbf{v}_{t_{n+1}}^{\text{cor}}$ :

$$\delta = \delta^{\text{free}} + h \mathbf{H} d\mathbf{v}_{t_{n+1}}^{\text{cor}} \quad (11)$$

$d\mathbf{v}_{t_{n+1}}^{\text{cor}}$  is the corrective motion and is found by solving (8) with  $\mathbf{b}_{t_n} = 0$ , which leads to:

$$\delta = \delta^{\text{free}} + \underbrace{h^2 \mathbf{H} \mathbf{A}_t^{-1} \mathbf{H}^T}_{\mathbf{W}} \lambda. \quad (12)$$

Then an algorithm of Gauss-Seidel is used to solve the contacts. Signorini's law is solved for each contact one after the other by integrating the coupling between them at each step.

$$\delta_i - \mathbf{W}_{ii} \lambda_i = \left( \sum_{j \neq i} \mathbf{W}_{ij} \lambda_j \right) + \delta_i^{\text{free}} \quad (13)$$

- After convergence, the value of  $\lambda$  is known and the corrective motion is applied.

As the number of contact increases, the dimension of  $\lambda$  increases, and solving the LCP takes longer. This can become a bottleneck when dealing with a high number of contacts. In the following, we propose a method to reduce the dimension of the contact space using pre-computed tests.

## III. REDUCTION BY PROJECTION

In this section, we will describe the method to reduce the dimension of the dual space of contact using a basis made of positive vectors. Before that, we show what is the effect of reducing the state variables (displacement or velocity) using POD (like it is done in [17] for example), on the LCP.

### A. Effect of POD-Reduction of the state variables

In traditional POD-based model order reduction methods, the state variables are expressed in a reduced space:

$$\mathbf{q}(t) \approx \mathbf{q}(0) + \sum_i \phi_i \alpha_i(t) = \mathbf{q}(0) + \Phi \boldsymbol{\alpha}(t) \quad (14)$$

This leads to the reduced version of equation (8):

$$\Phi^T \mathbf{A}_t \Phi d\boldsymbol{\alpha}_{t+1} = \Phi^T \mathbf{b}_t + \Phi^T \mathbf{H}^T \lambda, \quad (15)$$

From there, we get the reduced LCP:

$$\delta = \delta^{\text{free}} + \underbrace{h^2 \mathbf{H} \Phi \left( \Phi^T \mathbf{A}_t \Phi \right)^{-1} \Phi^T \mathbf{H}^T}_{\mathbf{W}_{\text{POD}}} \lambda, \quad (16)$$

We can see that the dimension of the contact problem has not changed, and in principle, there is no reason why performing the reduction of the state variables would translate into a reduction of the contact space. However, to some extent, it does speed up the computation of the matrix  $\mathbf{W}$  from (12) since the term  $\left( \Phi^T \mathbf{A}_t \Phi \right)^{-1}$ , which requires to invert a matrix is easier to compute since its dimension is reduced. With no reduction, we have to compute the inverse of the matrix  $\mathbf{A}_t$  (or at least its factorization) which can take a significant amount of time.

We see that this POD-reduction of state variables has the effect of reducing the number of degrees of freedom of the robot models, but not the collision degrees of freedom. In particular, if there are a lot of contacts, the reduction will not be sufficient to reach a real-time constraint.

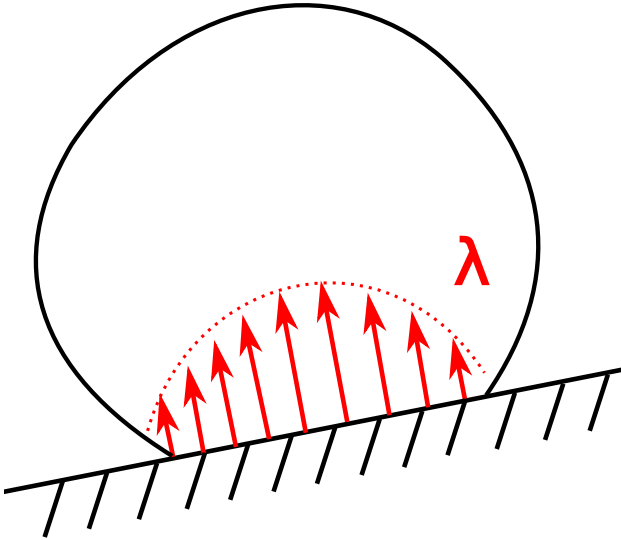


Fig. 3: Typical profile of the contact force of a soft object in contact with a plane. The contact force could be represented by simple bell with a magnitude encoded by a single scalar, rather than a detailed pointwise vector.

### B. Reduction of contact forces

In this section we will reduce the space of contact forces by using a projection of the contact force onto a reduced positive space. The idea is that we expect the contact forces to have some pattern that can be captured by a basis of small dimension. See Fig. 3. A reduction of the state variables using POD as presented in the previous section may or may not have been performed. Here, for simplicity, we will only reduce the contact space. We enforce the Lagrange multipliers  $\lambda$  to be a positive linear combination of fixed pre-computed positive vectors:

$$\lambda \approx \sum_i \psi_i \beta_i(t) = \Psi \beta(t) \quad (17)$$

with  $\Psi \geq 0$  and  $\beta(t) \geq 0$ . The formulation is similar to what we have for a POD-based reduction, except that here everything has to be positive, the scalar coefficients  $\beta_i$  and the basis  $\Psi$ , to ensure the positiveness of the reduced contact force.

Using this formulation, equation (12) becomes:

$$\delta = \delta^{\text{free}} + h^2 \underbrace{\mathbf{H} \mathbf{A}_t^{-1} \mathbf{H}^T}_{\mathbf{w}} \Psi \beta, \quad (18)$$

With a projection onto the reduced contact space, we get a reduced LCP:

$$\underbrace{\Psi^T \delta}_{\delta} = \underbrace{\Psi^T \delta^{\text{free}}}_{\delta^{\text{free}}} + h^2 \underbrace{(\Psi^T \mathbf{H}) \mathbf{A}_t (\Psi^T \mathbf{H})^T}_{\tilde{\mathbf{w}}} \beta, \quad (19)$$

The Signorini's law is still verified on the reduced space:

$$0 \leq \delta \perp \Psi \beta \geq 0 \quad (20)$$

$$\Leftrightarrow 0 \leq \hat{\delta} \perp \beta \geq 0 \quad (21)$$

Gauss-Seidel can be used, just like for the non-reduced case, to solve the reduced contacts together with the reduced Signorini law on the reduced variable  $\beta$ :

$$\hat{\delta}_i - \hat{\mathbf{W}}_{ii} \beta_i = \left( \sum_{j \neq i} \hat{\mathbf{W}}_{ij} \beta_j \right) + \hat{\delta}_i^{\text{free}} \quad (22)$$

Note that the reduction method would work with other LCP solvers, what matters is the way we build the LCP. With this reduced formulation, the number of contacts is fixed to the small dimension of the reduced contact basis  $\Psi$ , but the algorithm to solve the contact problem do not have to be changed.

### C. Construction of the reduced contact basis using snapshots and NNMF.

In this section we will explain how a positive basis representative of the contact space can be created. We proceed in a similar way to [24].

Similarly to a POD-based procedure, data will be collected in an offline stage: a snapshot  $\mathbf{S} \in \mathbb{R}^{n_c \times n_s}$  of the contact forces will be collected by simulating the robot in all possible postures (we assume there will be  $n_s$  postures), or at least its most expected deformations involving contacts within the computational power at hand. Here,  $n_c$  is the number of potential contact constraints, that depends on the level of discretisation of the surfaces of the objects. In practice, only a subset of these contact constraints are actually active at the same time. Since the simulations will be done using the a fine discretisation of the potential contact surfaces to guarantee accuracy,  $n_c$  may be large.

Now, in traditional POD based reduction, a singular value decomposition is used to extract the essence of the snapshot and output a set of basis vectors. Even if the snapshot contains only positive-valued vectors, there is absolutely no guarantee that the reduced basis will also be all-positive. Actually, it is almost always certain that the basis contains negative values. Hence, to generate the positive basis  $\Psi$ , we can not use SVD, but instead a non-negative matrix factorisation (NNMF). The procedure looks for a *non-negative* matrix  $\Psi \in \mathbb{R}^{n_c \times k}$  with  $k \ll n_c$  and a *non-negative* matrix  $\mathbf{B} \in \mathbb{R}^{k \times n_s}$  that solve the minimisation problem:

$$(\Psi, \mathbf{B}) = \arg \min_{(\Psi^*, \mathbf{B}^*)} \|\mathbf{S} - \Psi^* \mathbf{B}^*\|_F^2, \quad (23)$$

where  $\|\cdot\|_F$  stands for the Frobenius norm.

This minimisation problem is solved using a heuristic algorithm called alternating least-square (ALS) [26]. The procedure is described in algorithm 1.

This algorithm gives a sub-optimal solution that in practice is sufficient to obtain a decent basis. It is sub-optimal in the sense there is no guarantee it will find the best solution, but it reaches a local optimum. The fact the basis is positive forces the approximation of the contact forces to be an addition of contributions and this tends to give solutions with mutually exclusive support since no contributions can be subtracted.

**Algorithm 1** ALS algorithm

---

```

1: procedure ALS( $\mathbf{S}, n_C, k$ )  $\triangleright$  The NNMF of  $\mathbf{S}$  of size  $k$ 
2:    $\Psi \leftarrow \text{rand}(n_C, k)$ 
3:   while  $n_{iter} < \text{max}_{iter}$  do
4:     Find  $\mathbf{B}$  solving  $\Psi^T \Psi \mathbf{B} = \Psi^T \mathbf{S}$ .
5:     Set all negative values in  $\mathbf{B}$  to 0.
6:     Find  $\Psi$  solving  $\mathbf{B} \mathbf{B}^T \Psi^T = \mathbf{B} \mathbf{S}^T$ .
7:     Set all negative values in  $\Psi$  to 0.
8:      $err \leftarrow \|\mathbf{S} - \Psi^* \mathbf{B}^*\|_F^2$ 
9:   end while
10:  Normalise  $\Psi$ 
11: end procedure

```

---

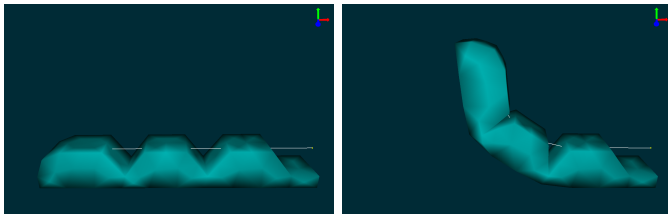


Fig. 4: Finger actuated by a single cable. (Left) Initial Position (Right) Position after a small actuation

#### IV. APPLICATIONS

In this section, we will present two examples:

- a simple soft finger on which we will show the reduced contact method is compatible with inverse modelling
- a real soft manipulator robot which undergoes self-collisions useful to stiffen its behaviour

In both cases, the simulations are implemented in the SOFA framework<sup>1</sup>. Our goal in this section is to validate the accuracy and the speed-up of the simulation using the reduced contact model versus using the full contact model. In the case of the real manipulator, we assume a good concordance between the real robot and the detailed FE model with full contact description.

##### A. Cable-driven Soft Finger

We consider a toy example with soft finger actuated by a single cable. When pulling on the cable, the finger bends. When actuating by a large quantity, self-collisions have to be considered to obtain a correct deformation in the FE model. See Fig. 4 to see the deformation in various cases.

The behaviour of the robot is modeled using FEM with tetrahedral elements. We use a collision model based on points and triangles. Forces from the collision model are transferred to the FE model using a barycentric mapping. The collision model is displayed in orange in Fig. 6. Note that in this case, the collision model we chose is artificially fine, and would in principle be able to detect very small contact features. However on the application shown here, a coarser model would also work. For collision detection, which is activated when one triangle gets too close to an

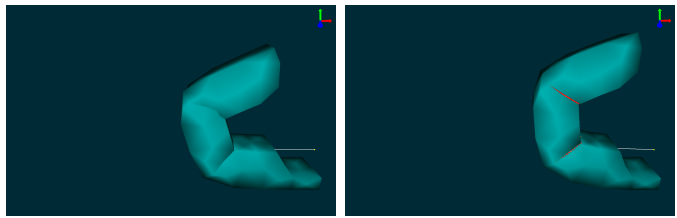


Fig. 5: When pulling by a large amount, self collisions have to be taken into account to model a correct behaviour. (Left) Without collision detection (Right) With Collision Detection

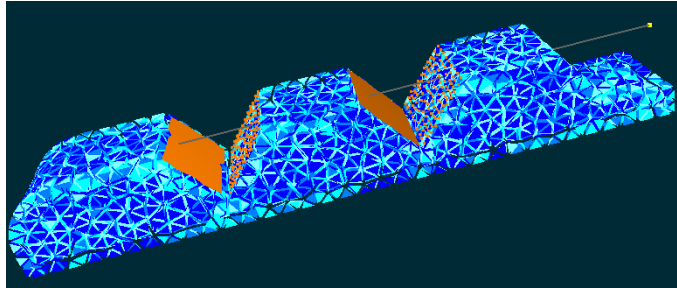


Fig. 6: Finger Self-collision model in orange. We consider the collision between points on one side of the soft articulation and triangles on the other. Here we use a refine model with 495 points. In blue is displayed the FE mesh.

opposite point, we use the pipeline provided by the SOFA framework.

*a) Direct model:* We proceed to apply the reduction method described in section III-B to reduce the contact forces in this example. The snapshot is generated by incrementally pulling on the cable by steps of 0.05 mm up to a pull of 40 mm, storing contact forces  $\lambda$  along the way, which leads to a snapshot with size 800. We then apply the ALS algorithm (described in section III-C) to select a positive basis of dimension 5 (dimension chosen arbitrarily). See in Fig. 7 the first four modes.

*b) Inverse model:* The method is implemented in the SOFA framework, and can also be used for the inverse model using the softRobots. See Fig 8. The effector is defined as the tip of the finger and the inverse problem is defined as finding the actuation that moves the effector to reach a target 3D position, while at the same time respecting the constraint of collision. The inverse problem is solved using a Quadratic Programming with Complementarity Constraints (QPCC) solver implemented in the soft robotics plugin for SOFA (More details can be found in [9]), and the collision treatment is managed using the reduced model. We can see that the method allows for solving the inverse problem involving self-collisions in real-time.

##### B. Real robot: deformable manipulator robot with a compliant spine

We are considering the deformable manipulator pictured in Fig. 9, and presented in [27]. The manipulator is made of

<sup>1</sup>www.sofa-framework.org

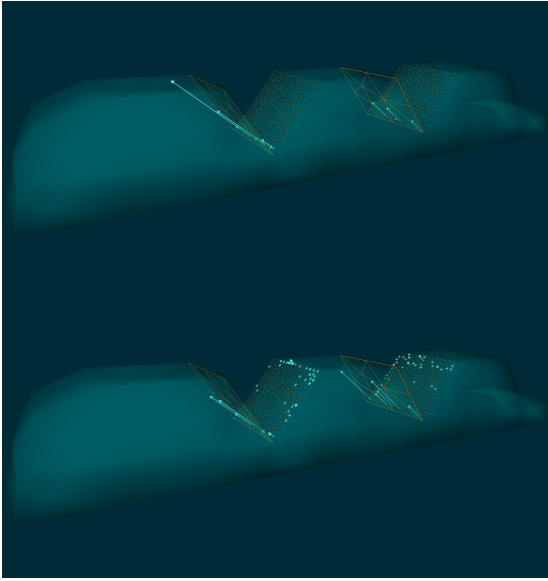


Fig. 7: Two contact modes displayed with arrows. The modes cater for the progressive evolution of the contact surface when pulling on the cable. At the start, contact occurs on a small support in the valley of the finger soft articulation, before spreading along.

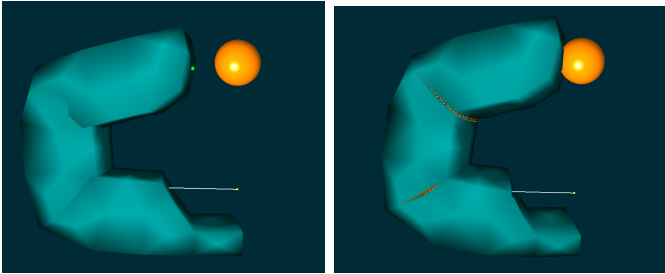


Fig. 8: Finger model not taking self-collision into account (Left) versus Finger using a reduced contact model (Right). The effector is the finger tip and the goal is materialised by the sphere. Without self-collision treatment, the finger self-penetrates leading to a unrealistic behaviour. When using the reduced contact model, the inverse problem can be solved in real time while giving a realistic prediction of the finger deformation.

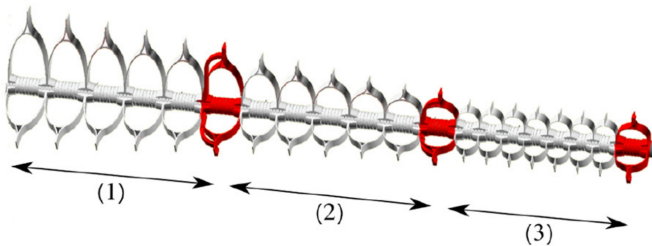


Fig. 9: Deformable manipulator robot with a skeletal compliant spine inspired by vertebrate animals. The robot is made of 3 sections each actuated by 3 tendons.

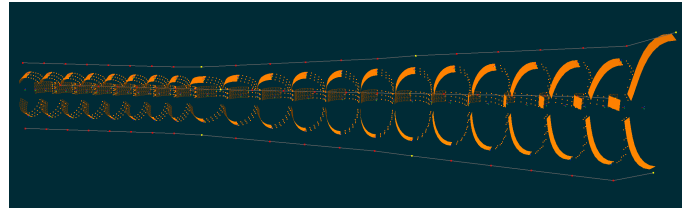


Fig. 10: Collision Model of the manipulator made of triangle and points. This model has a dimension of size 1290.

3D-printed vertebrae and is actuated using tendons. This robot tends to undergo self-collisions of the ribs on each other by design as the cables are pulled and the arm bends. These collisions create some artificial stiffness all along the manipulator, and the more the manipulator is bent, the stiffer it gets since the surface of the robots enduring self-collisions increases. See Fig. 1.

In [27], a kinematic model was made using real-time finite element simulation defined by a beam model for the ribs and the backbone of the manipulator, but self-collisions were neglected for computational reasons. Thanks to the information provided by the sensors and the error compensation of a closed loop, the robot could be precisely controlled in position. Another possibility would be to fit a collisionless model with equivalent mechanical parameters, for example following the method from [28]. However, it would tend to be over-stiff to compensate for the lack of collision response treatment. In the following, we will improve that model by including the self-collision response. First, we will show how the behaviour of the robot is changed, and then, to reach real-time performance, we will then apply the reduced contact technique presented in section III-B.

The collision model is set on each rib by facing triangles versus points, in a similar way to what was done in the previous example with the soft finger. This leads to a contact space of dimension 1290. It is displayed in Fig 10.

To generate the snapshot space of contact forces, we do not choose to use an exhaustive sampling to cover all possible actuations since with 9 actuators, the number of possible actuation combination is quite high (at least  $2^9 = 512$  if we only store the maximal actuation for each cable). Instead, we use a Plackett-Burman design [29], which leads to only 12 positions but is sufficient to capture most of the contact scenarios. To generate the data, we reach successively these positions in 120 steps, allowing to capture the progressive increase of the contact supports along the actuation. This leads to a snapshot of size 1440.

*a) Results:* In Fig. 11 is displayed the different behaviour in simulation between the model with and without collision when a certain force emulating the position of the real robot in Fig 1 is given as input. It can be seen that a good part of the cable forces is taken by the self-collision on the ribs that as a response tend to bend, and this leads to a realistic position of the effector much lower than in

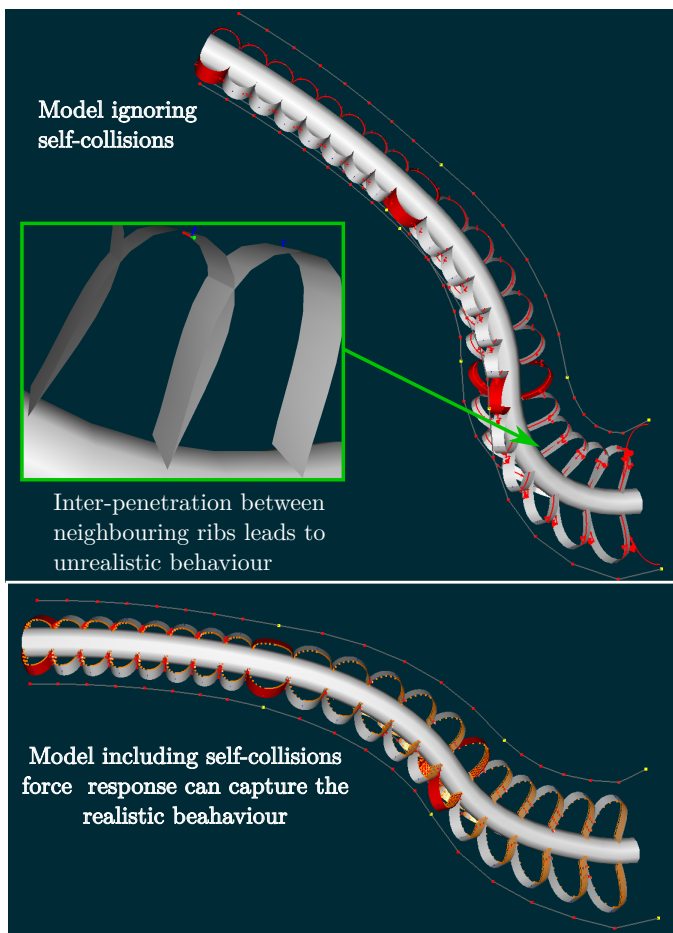


Fig. 11: Comparison between the model ignoring self-collisions (Top) and the model capturing self-collisions (Bottom), in a situation where the actuators are pulled with an identical force. Without considering collisions, the manipulator tends to bend in an unrealistic way, while when considering self-collisions, the amount of bending is limited by the stiffness created by the collision response. This behaviour is similar to the position obtained by the real robot where we can see that self-collisions do occur in an important manner in Fig. 1.

the case self-collisions are ignored.

In Table II, are displayed computational times of the full order model versus the reduced version, in cases with and without contact. We can see that quite unsurprisingly, in the case where contacts are not involved, the computation time between the full model and the reduced model is identical. However, as soon as contacts occur, the reduced contact model can enter in action, and while the full model has a dramatic slowdown, the reduced contact model mitigate very much this slowdown by a factor of about 85 in this case.

In Fig. 12, we can see the accuracy of the reduced contact model depending on the dimension of the reduced contact basis. When that basis is too small, the self-collisions are missed and the error is similar to the case without collisions. The error then decreases as the basis

Contact Model	Contact Model size	Timesteps/seconds
Full /no contact	4500	18
Full / contact	4500	0.2
MOR /no contact	23	18
MOR / contact	23	17

TABLE II: Timestep rate for the full order simulation and the contact reduced model (MOR) using 23 contact modes, in the case the manipulator is straight (so there are no collisions yet occurring) or fully bent (so many collisions occur). The reduced model outperforms the full model by a factor up to around 85 when many self-collisions happen. The benchmark was made on an HP laptop with 4 Intel Core(TM) i7-7820HQ CPU with 2.90GHz

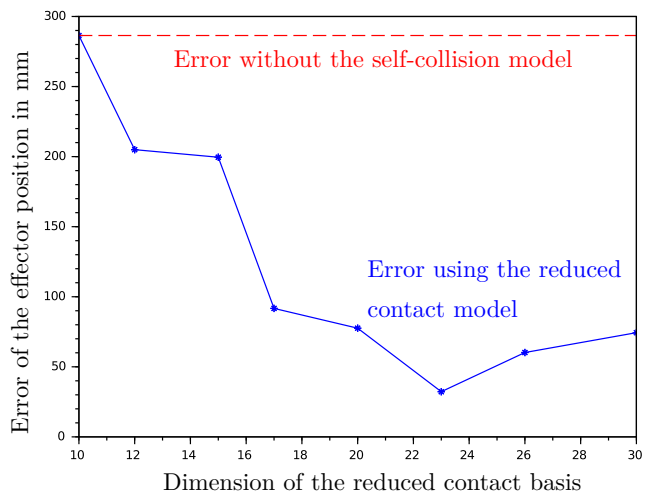


Fig. 12: Error in position between the model with reduced contacts using bases with increasing dimension and the full order contact model. We can see that using a reduced contact basis with dimension 23 leads to an error of about 30 mm, to be compared with an error of about 280 mm for a model ignoring self-collisions.

grows, but at some point, the error rises again. This is due to the ALS algorithm that is a heuristic algorithm and tends to fail in its purpose when the problem becomes too large in proportion with the amount of data collected. However, with a good guess of the size of the basis at around 25, we can see that the reduced model captures the full order contact model by an error of about 3cm, to compare with the 28cm error when ignoring self-collisions (note that the arm is about 80 cm long). With this reduced model we can have a refresh rate of the full mechanical model at 18 FPS on a standard laptop.

## V. DISCUSSION

Despite promising results, the method proposed in this letter has some limitations. The heuristic algorithm (ALS) used to find the contact basis may fail in some cases and it is up to the user to make sure the computed basis gives good results. In particular, the dimension of the basis has to be chosen a priori, and we have shown that



this can affect the results. Also the deformation model used to compute the behaviour of the robot needs to be able to handle contacts. For example, the manipulator was modeled with a beam model, and a mapping was necessary to transfer the forces from the beam to the surface of the robot and vice versa.

For the sampling phase, it could be improved by taking advantage of possible symmetries of the robot.

Finally, we only considered frictionless contacts in this paper. Coulomb friction, for example, is harder to take into account since the stick/slip condition makes the contact behaviour non-smooth.

## VI. CONCLUSION AND PERSPECTIVES

In this paper, we develop a reduced contact model to simulate the behaviour of soft robots with non-trivial self-collision scenarios in real time. The method is compatible with inverse modelling allowing this formulation to be used for control. The method is generic and can in principle be used with any type of robot geometry. We think the method could be used for designing robot kinematics with internal collisions. In principle, the method is applicable beyond self-collisions and it is possible to take into account any kind of collisions that may happen with the environment. However, unlike self-collisions which tend to occur in a repeated and predictable way and are therefore representable by a reduced contact basis of small dimension, general collisions scenarios would need to be carefully listed to make sure they are not too numerous to be captured by a reasonably small basis.

## REFERENCES

- [1] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Self-collision detection and prevention for humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 3. IEEE, 2002, pp. 2265–2270.
- [2] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, and A. Kheddar, "Real-time (self)-collision avoidance task on a hrp-2 humanoid robot," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3200–3205.
- [3] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.
- [4] M. Selvaggio, S. Grazioso, G. Notomista, and F. Chen, "Towards a self-collision aware teleoperation framework for compound robots," in *2017 IEEE World Haptics Conference (WHC)*. IEEE, 2017, pp. 460–465.
- [5] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [6] M. Ciocarlie, C. Lackner, and P. Allen, "Soft finger model with adaptive contact geometry for grasping and manipulation tasks," in *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*. IEEE, 2007, pp. 219–224.
- [7] H. Dang, J. Weisz, and P. K. Allen, "Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 5917–5922.
- [8] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.
- [9] E. Coevoet, A. Escande, and C. Duriez, "Optimization-based inverse model of soft robots with contact handling," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1413–1419, 2017.
- [10] N. Verdon, C. A. and C. Béghein, A. Hamdouni, and D. Ryckelynck, "Reduced-order modelling for solving linear and non-linear equations," *International Journal for Numerical Methods in Biomedical Engineering*, vol. doi:10.1002/cnm.1286, 2009.
- [11] D. Amsallem and C. Farhat, "An Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity," *AIAA Journal*, vol. 46, no. 7, pp. 1803–1813, 2008.
- [12] J.-C. J. T. Braconnier, M. Ferrier and P. Sagaut, "Towards an adaptive pod/svd surrogate model for aeronautic design," *Computers and Fluids*, vol. 40, no. 1, pp. 195–209, 2011.
- [13] K. Kunisch and S. Volkwein, "Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics," *SIAM Journal on Numerical analysis*, vol. 40, no. 2, pp. 492–515, 2003. [Online]. Available: <http://www.jstor.org/pss/4100966>
- [14] P. Kerfriden, O. Gouy, T. Rabczuk, and S. P.-A. Bordas, "A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics," *Computer methods in applied mechanics and engineering*, 2012.
- [15] O. Gouy, D. Amsallem, S. P. A. Bordas, W. K. Liu, and P. Kerfriden, "Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to bayesian optimization," *Computational Mechanics*, vol. 58, no. 2, pp. 213–234, 2016.
- [16] J. Barbič, "Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. part 2: Model reduction," in *SIGGRAPH 2012 Course Notes*. Citeseer, 2012.
- [17] O. Gouy and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.
- [18] M. Thieffry, A. Kruszewski, O. Gouy, T. Guerra, and C. Duriez, "Dynamic control of soft robots," in *Preprints of International Conference on Automatic Control (IFAC)*, 2017.
- [19] Y. Teng, M. A. Otaduy, and T. Kim, "Simulating articulated subspace self-contact," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–9, 2014.
- [20] D. L. James and D. K. Pai, "Bd-tree: output-sensitive collision detection for reduced deformable models," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 393–398.
- [21] D. L. James and K. Fatahalian, "Precomputing interactive dynamic deformable scenes," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 879–887, 2003.
- [22] Y. Teng, M. Meyer, T. DeRose, and T. Kim, "Subspace condensation: full space adaptivity for subspace deformations," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–9, 2015.
- [23] J. Allard, F. Faure, H. Courtecuisse, F. Falipou, C. Duriez, and P. Kry, "Volume Contact Constraints at Arbitrary Resolution," *ACM Transactions on Graphics*, vol. 29, no. 3, Aug. 2010. [Online]. Available: <https://hal.inria.fr/inria-00502446>
- [24] M. Balajewicz, D. Amsallem, and C. Farhat, "Projection-based model reduction for contact problems," *International Journal for Numerical Methods in Engineering*, vol. 106, no. 8, pp. 644–663, 2016.
- [25] J. Fauque, I. Ramière, and D. Ryckelynck, "Hybrid hyper-reduced modeling for contact mechanics problems," *International Journal for Numerical Methods in Engineering*, vol. 115, no. 1, pp. 117–139, 2018.
- [26] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational statistics & data analysis*, vol. 52, no. 1, pp. 155–173, 2007.
- [27] T. Bieze, A. Kruszewski, B. Carrez, and C. Duriez, "Design, implementation, and control of a deformable manipulator robot based on a compliant spine," *The International Journal of Robotics Research*, 2020.
- [28] D. Hahn, P. Banzet, J. M. Bern, and S. Coros, "Real2sim: Viscoelastic parameter estimation from dynamic motion," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.
- [29] R. L. Plackett and J. P. Burman, "The design of optimum multifactorial experiments," *Biometrika*, vol. 33, no. 4, pp. 305–325, 1946.