2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Reduced Order Control of Soft Robots with Guaranteed Stability

Maxime Thieffry[1,2], Alexandre Kruszewski[2], Thierry-Marie Guerra[1] and Christian Duriez[2]

Abstract— **This work offers the ability to design a closed-loop strategy to control the dynamics of soft robots. A numerical model of a robot is obtained using the Finite Element Method, which leads to work with large-scale systems that are difficult to control. The main contribution is a reduced order model-based control law, that consists in two main features: a reduced state feedback tunes the performance while a Lyapunov function guarantees the stability of the large-scale closed-loop systems. The method is generic and usable for any soft robot, as long as a FEM model is obtained. Simulation results show that we can control and reduce the settling time of the soft robot and make it converge faster without oscillations to a desired position.**

I. INTRODUCTION

Soft robots - robots made of deformable materials - promise disruptive advances in many areas and bring transversal challenges, among which are dynamical modelling and control, see [1], [2] and [3]. Being lighter and more compliant than rigid ones, vibrations issues arise when dealing with soft robots dynamics. We propose a feedback control design to handle these issues. Designing such a feedback law brings different challenges such as dynamical modelling, large scale control and stability preservation.

Modelling soft robots analytically is hard to achieve as this new type of robot has a theoretical infinite number of degrees of freedoms. Several approaches has been proposed so far: piece-wise constant curvature (PCC), Cosserat rod theory and finite element method (FEM). Depending on the robot geometry, the constant curvature is not always valid. Moreover, the equations coming from the Cosserat model are not often suitable for controller design. Recent work has been done to deal with these issues in [4] and [5] to study continuum manipulators or beam-like soft robots. The goal of the present method is to be as generic as possible concerning the geometry of the robot, therefore, the finite element method is used.

This spatial discretization gives rise to large scale systems, which are abounding in many fields of research, such as control theory. Standard control theory tools, like Linear Matrix Inequalities (LMI) or Lyapunov equation solvers, can not deal with a too large number of decision variables. Numerical efficiency for control applications is also an active field of research, see [6], [7] or [8].

If classical tools of automatic control have to be applied, model order reduction must be considered. The methods used in this work are rather standard and we refer the reader to [9] for more details.

Recent work has been done on soft robots control: [10] provides an open-loop stategy for dynamics and trajectory optimization, [11] provides an open-loop trajectory planning approach based on FEM model and [12] provides a closed-loop controller based on FEM model but restricted to kinematics. Recently, space reduction has also been used to study continuum manipulators in [13].

This paper aims at providing a closed-loop controller that fixes the dynamics of a reduced order system while guaranteeing the stability of the full order model using Lyapunov framework. The remainder of this document is organised as follows. Section II presents a dynamic model of the robot while the main contribution of this work is presented in section III and section IV presents tracks to improve this result. Simulation results are provided along with the theoretical results to illustrate and show the effectiveness of the methodology.

II. SOFT ROBOT MODELLING

A. FEM model

Modelling soft robots relies on both continuum mechanics theory and numerical approaches to solve the underlying equations. Here, a corotated FEM allows us to define position and velocity vectors, respectively $q(t) \in \mathbb{R}^n$ and $v(t) \in \mathbb{R}^n$ whose dimension n is proportional to the size of the FEM mesh used to model the robot. The more nodes the mesh has, the more the model tends to be accurate and, for soft robots application, the size $\mathcal{N}$ of the FEM mesh goes from hundreds to thousands of variables. The dimension n of the previous vectors is made of $3 \times \mathcal{N}$ variables, as the position and velocity vectors are given in the 3 dimension of space.

Using the SOFA framework, the method proposed in [14] describes concretely how to use FEM to model soft robots, in the particular context of real-time simulation.

The non-linear equation of motion of the robots is given by the second law of Newton:

$$M(q)\dot{v} = P(q) - F(q,v) + H^T(q)u(t) \tag{1}$$

where $M(q)$ is the mass matrix, $H^T(q)u$ is the actuators contribution : $H^T(q)$ contains the direction of the actuators forces and u their amplitude. The matrix $F(q,v)$ represents the internal forces and $P(q)$ gathers all the known external forces. As we consider only the gravity field, $P(q)$ is constant and $P(q) = P$.

Let $q_0 \in \mathbb{R}^n$ be a stable equilibrium point induced by P and $u(t) = u_0$, i.e. q_0 is solution to

$$0 = P - F(q_0,0) + H^T(q_0)u_0 \tag{2}$$

[1]Univ. Valenciennes, UMR 8201 - LAMIH F-59313 Valenciennes, France

[2]Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

978-3-9524-2699-9 ©2018 EUCA

Equation (1) can also be written as:

$$\begin{aligned} &M(q)\dot{v} = P - F(q,v) + H^T(q)u(t) - P + F(q_0,0) - H^T(q_0)u_0 \\ &\quad\Leftrightarrow \\ &M(q)\dot{v} = F(q_0,0) - F(q,v) + H^T(q)u(t) - H^T(q_0)u_0 \end{aligned} \tag{3}$$

We can approximate the internal forces F with a first order Taylor expansion around this equilibrium point:

$$F(q,v) \approx f(q_0,0) + \left.\frac{\partial F(q,v)}{\partial q}\right|_{q=q_0}(q-q_0) + \left.\frac{\partial F(q,v)}{\partial v}\right|_{v=0} v \tag{4}$$

where $\frac{\partial F(q,v)}{\partial q} = K(q,v)$ is the compliance matrix, and $\frac{\partial F(q,v)}{\partial v} = D(q,v)$ is the damping matrix. By definition, mass, compliance and damping matrices are positive definite. With these notations, equation (3) becomes:

$$M(q)\dot{v} \approx -K(q_0,0)(q-q_0) - D(q_0,0)v + H^T(q)u(t) - H^T(q_0)u_0 \tag{5}$$

Let d be the displacement vector defined by:

$$d = q - q_0 \tag{6}$$

The equation of motion around an equilibrium point q_0 is thus given by the following relation:

$$M(q)\dot{v} \approx -K(q_0,0)d - D(q_0,0)v + H^T(q)u(t) - H^T(q_0)u_0 \tag{7}$$

B. State-space Equation

Without loss of generality, considering $u_0 = 0$ allows us to define the following non-linear state-space equation:

$$\begin{cases} \dot{x} = \underbrace{\begin{pmatrix} -M(x)^{-1}D(x) & -M(x)^{-1}K(x) \\ I & 0 \end{pmatrix}}_{A(x)} x + \underbrace{\begin{pmatrix} M(x)^{-1}H^T(x) \\ 0 \end{pmatrix}}_{B(x)} u \\ y = Cx \end{cases} \tag{8}$$

where $x = \begin{pmatrix} v \\ d \end{pmatrix}$, $x \in \mathbb{R}^{2n}$ and where system matrices are large-scale sparse non-linear matrices, i.e. $A(x) \in \mathbb{R}^{2n \times 2n}, B(x) \in \mathbb{R}^{2n \times m}, C \in \mathbb{R}^{p \times 2n}$, where m is the number of actuators and p the number of outputs.

The results showed in this paper are obtained on simulation experiments, where the non-linear model is used to simulate the robot. For control application, it is more direct to work on a linear model to design the control law and around the equilibrium point q_0, system (8) can be approximated by the following linear representation:

$$\begin{cases} \dot{x} = \begin{pmatrix} -M^{-1}D & -M^{-1}K \\ I & 0 \end{pmatrix} x + \begin{pmatrix} M^{-1}H^T \\ 0 \end{pmatrix} u \\ y = Cx \end{cases} \tag{9}$$

where $M = M(q_0, 0)$, $D = D(q_0, 0)$ and $K = K(q_0, 0)$.

Remark 1: The FEM model of the robot allows us to compute its energy in real-time. The kinetic energy of a soft robot is defined as:

$$E_k(x) = \frac{1}{2} v^T M v \tag{10}$$

and its potential energy:

$$E_p(x) = \frac{1}{2} d^T K d \tag{11}$$

The total energy of the robot is then:

$$E(x) = \frac{1}{2}\begin{pmatrix} v \\ d \end{pmatrix}^T \begin{pmatrix} M & 0 \\ 0 & K \end{pmatrix} \begin{pmatrix} v \\ d \end{pmatrix} \tag{12}$$

As M and K are positive definite, i.e. $M > 0$ and $K > 0$, this energy function is positive definite:

$$E(x) > 0 \tag{13}$$

and

$$E(x) = 0 \Leftrightarrow (q,v) = (0,0) \tag{14}$$

III. LARGE SCALE CONTROL DESIGN

The general objective of this paper is to compute a state feedback control law

$$u = Lx \tag{15}$$

that guarantees the performance of the system (9) in closed-loop. Yet, using pole placement or LMI approaches, the dimension of x implies that the computation of the matrix L cannot be done with numerical efficiency. Moreover, this controller could guarantee the closed-loop stability of the large scale model, but to be usable it would require the measurement of the whole state, which is of course impossible in practice due to the large number of variables to measure.

The objective of this work is also to reduce the number of parameters in the controller. The use of Lyapunov stability and LMI constraints to study the stability and design the controller permits to optimize the performance of the system.

In many cases, energy functions can be used as Lyapunov functions. Without actuation the studied soft robot converges to a natural equilibrium point where its energy is zero, this energy function is also a Lyapunov function for the system in open-loop. This allows us to design a Lyapunov function based on the system matrices and limits the complexity in the choice of the Lyapunov function. However, this does not reduce the complexity of the controller design, as the matrix L still contains a lot of variables to tune.

The contribution of this paper is a method to deal with this issue: relying on model reduction techniques, a reduced order state feedback control law

$$u = L_r \xi_r \tag{16}$$

is computed and a proof of stability for the original large scale system is given, using large-scale Lyapunov functions.

A. Model Order Reduction

Before developping the control part, this subsection presents notion of model order reduction that are required to develop the main results. There are two main branches of model order reduction methods: the first one based on optimisation [15] and the other based on projection [16]. As the implementation of optimisation based model reduction is still challenging for very large scale systems, only projection methods are used here . These methods aim at computing two projectors $\mathcal{W}_r \in \mathbb{R}^{n \times r}$ and $\mathcal{V}_r \in \mathbb{R}^{n \times r}$, with $\mathcal{W}_r^T \mathcal{V}_r = I_r$, to approximate a large scale system

$$\dot{x} = f(x,u)\ , x \in \mathbb{R}^n \tag{17}$$

with a reduced order one:

$$\dot{\xi}_r = \mathcal{W}_r^T f(\mathcal{V}_r \xi_r, u), \xi_r \in \mathbb{R}^r,\ r \ll n \tag{18}$$

where

$$\xi_r = \mathcal{W}_r^T x \tag{19}$$

is the reduced order state and $\xi_{\bar{r}}$ is the neglected one:

$$\xi_{\bar{r}} = \mathcal{W}_{\bar{r}}^T x,\ \xi_{\bar{r}} \in \mathbb{R}^{\bar{r}},\ \bar{r} = n - r \tag{20}$$

such that:

$$x = \mathcal{V}_r \xi_r + \mathcal{V}_{\bar{r}} \xi_{\bar{r}} \tag{21}$$

For soft robotics applications, to compute an approximation of the full-order state $x = \begin{pmatrix} v \\ d \end{pmatrix}$, it is interesting to use *structure preserving* model order reduction. The reduced and neglected states will also keep their initial structure:

$$\xi_r = \begin{pmatrix} \xi_{rv} \\ \xi_{rd} \end{pmatrix}\ ;\ \xi_{\bar{r}} = \begin{pmatrix} \xi_{\bar{r}v} \\ \xi_{\bar{r}d} \end{pmatrix}\ ; \tag{22}$$

Concretely, this requires to find projectors for the velocity and the displacement vectors such that equation (21) writes:

$$\begin{aligned} \begin{pmatrix} v \\ d \end{pmatrix} &= \begin{pmatrix} \mathcal{V}_{rv} & 0 & \mathcal{V}_{\bar{r}v} & 0 \\ 0 & \mathcal{V}_{rd} & 0 & \mathcal{V}_{\bar{r}d} \end{pmatrix} \begin{pmatrix} \xi_{rv} \\ \xi_{rd} \\ \xi_{\bar{r}v} \\ \xi_{\bar{r}d} \end{pmatrix} \\ \Leftrightarrow x &= \mathcal{V}\Xi \end{aligned} \tag{23}$$

where $\Xi \in \mathbb{R}^n$ is the projected state. To find the projectors $\mathcal{V}$ and $\mathcal{W}$, state-of-the-art model reduction methods are available, such as *Improved Balanced Truncation* [17], *Iterative Interpolation Methods* [18] and *Proper Orthogonal Decomposition (POD)*. The first two methods are for now restricted to the linear case whereas the POD is a projection based method adapted to non-linear systems.

B. Reduced Order Model-Based Controller

To get rid of the whole state measurement to control the large scale system using equation (15), we aim at computing a *reduced order state feedback controller*. Starting from the Lyapunov function defined previously:

$$E(x) = x^T \begin{pmatrix} M & 0 \\ 0 & K \end{pmatrix} x = \Xi^T \mathcal{V}^T \begin{pmatrix} M & 0 \\ 0 & K \end{pmatrix} \mathcal{V}\Xi \tag{24}$$

In open-loop, with $\dot{x}$ defined in eq (9), the derivative of this function is:

$$\dot{E}(x) = x^T \begin{pmatrix} -D & 0 \\ 0 & 0 \end{pmatrix} x + (*) = -2v^T D v \tag{25}$$

where $(*)$ represents the transpose of the matrix preceding it.

The reduced order state ξ_r is of reasonable dimension and we can design a reduced order feedback

$$u = L_r \xi_r = \begin{pmatrix} L_{rv} & L_{rd} \end{pmatrix} \begin{pmatrix} \xi_{rv} \\ \xi_{rd} \end{pmatrix} \tag{26}$$

that fixes the performance of the closed loop. With this control law, the derivative of the Lyapunov function becomes:

$$\dot{E}(x) = x^T \begin{pmatrix} -D & 0 \\ 0 & 0 \end{pmatrix} x + x^T \begin{pmatrix} H^T \\ 0 \end{pmatrix} u + (*) \tag{27}$$

which, in the projected space, is equivalent to:

$$\begin{aligned} \dot{E}(\Xi) &= \Xi^T \mathcal{V}^T \begin{pmatrix} -D & 0 \\ 0 & 0 \end{pmatrix} \mathcal{V}\Xi + \Xi^T \mathcal{V}^T \begin{pmatrix} H^T \\ 0 \end{pmatrix} u + (*) \\ &= \Xi^T \mathbf{Q} \Xi + \begin{pmatrix} \xi_r \\ \xi_{\bar{r}} \end{pmatrix}^T \mathcal{V}^T \begin{pmatrix} H^T \\ 0 \end{pmatrix} \begin{pmatrix} L_r & 0 \end{pmatrix} \begin{pmatrix} \xi_r \\ \xi_{\bar{r}} \end{pmatrix} + (*) \\ &= \Xi^T (\mathbf{Q} + \mathbf{R}) \Xi \end{aligned} \tag{28}$$

where

$$\mathbf{Q} = \begin{pmatrix} -\mathcal{V}_{rv}^T D \mathcal{V}_{rv} & 0 & -\mathcal{V}_{rv}^T D \mathcal{V}_{\bar{r}v} & 0 \\ 0 & 0 & 0 & 0 \\ -\mathcal{V}_{\bar{r}v}^T D \mathcal{V}_{rv} & 0 & -\mathcal{V}_{\bar{r}v}^T D \mathcal{V}_{\bar{r}v} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + (*) \tag{29}$$

and

$$\mathbf{R} = \begin{pmatrix} \mathcal{V}_{rv}^T H^T L_v & \mathcal{V}_{rv}^T H^T L_d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mathcal{V}_{\bar{r}v}^T H^T L_v & \mathcal{V}_{\bar{r}v}^T H^T L_d & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + (*) \tag{30}$$

with $\mathbf{Q}, \mathbf{R} \in \mathbb{R}^{2n \times 2n}$. The computation of the matrix L_r is done using the framework of LMI constraints. The LMI is defined as:

$$\dot{E}(x) < -\lambda E(x) \Leftrightarrow (\mathbf{Q} + \mathbf{R}) < -\lambda \begin{pmatrix} M & 0 \\ 0 & K \end{pmatrix} \tag{31}$$

where λ is a positive scalar that fixes the decay rate of the function $E(x)$. The stability is proven for the large-scale system, even if the number of variables in the LMI is equal to $m \times r$, where m is the number of input and r is the reduced dimension.

C. Simulation Results

The method is tested on a simulated model of a soft robot made of silicone, shown on the following picture:

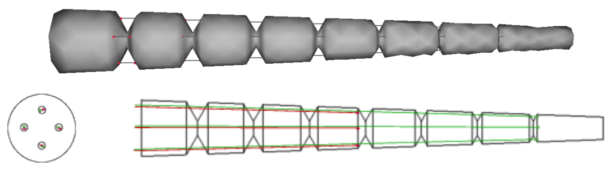

Fig. 1. Top: Visual model of the robot
Bottom: Design of the robot: slice view (left) and side view (right).
The robot is actuated with 8 cables: 4 short cables in red, and 4 long ones in green.

This robot is actuated with 8 cables mounted on the structure as shown on figure 1 so that the robot can deform in any direction of space. Friction between the robot and its cables is neglectable thanks to flexible tubes that guide the cables and that are added in the simulation.

SOFA integrates the CGAL library to compute a FEM mesh from a visual model. In this case, the mesh is made

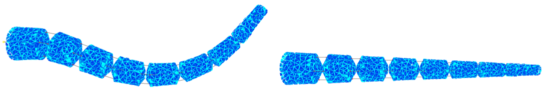

Fig. 2. FEM mesh of the Trunk-like robot, made of 1557 nodes.
Left : Deformed position; Right : Rest position.

of 1557 nodes and 5157 tetrahedron elements, as shown on figure 2. The size n of both velocity and position vectors is equal to $n = 3 \times 1557 = 4671$, and thus system (9) is of dimension 9342. Computing a full-order state feedback, as in (15) would have implied the computation of a matrix L of dimension 8×9342, with 74 736 variables to compute. Instead of that, the model reduction step provide us with a reduced system of dimension 6, leading to a feedback matrix $L_r \in \mathbb{R}^{m \times r}$ where $m = 8$ and $r = 6$. Solving LMI defined in equation (31) needs the computation of 48 decision variables.

In simulation experiments the state vector, i.e. both velocity and position vectors, is directly available and we use POD reduction method to obtain the reduced order state. The simulation results are given for the following example: the robot starts from its initial shape (left of figure 2) and converges to its rest position (right of figure 2), where the model has been linearized.

The behaviour of the robot in open-loop is illustrated on figure 3 and 4, it shows that the robot converges to its equilibrium point after some oscillations. The goal of the proposed control method is to reduce, or suppress, this oscillations and make the robot converge faster to the desired position.

The control law defined in (26) is directly usable in practice, as it only requires the measurement of the state of the robot, i.e. displacement and velocity vectors. In simulation, they are directly available and we reconstruct the reduced state by applying equation (19) at each time step. (To test on real robots, an observer should be used to compute the reduced order state from the sensor measurement, this work is not conducted in this paper.)

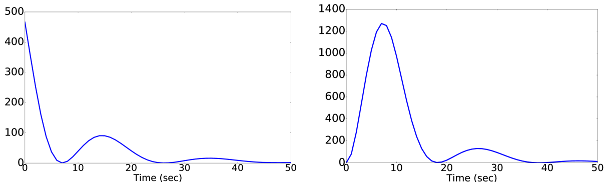

Fig. 3. Left : Norm of the displacement (cm) in open-loop
Right: Norm of velocity (cm/s^2) in open-loop.

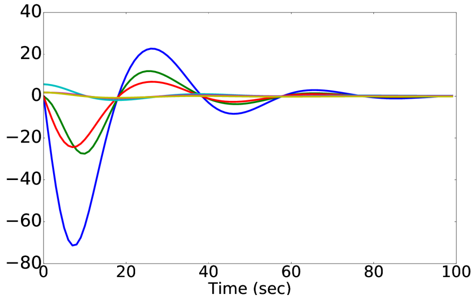

Fig. 4. Reduced order state in open-loop.

With this feedback controller, the oscillations vanish, as shown on figure 5 and 6. This control law allows us to suppress the oscillations in the robot behaviour and the convergence time of the robot is decreased. However, the gain L_{rd} is near zero and does not seem to have an impact on the closed-loop performance. Looking at the definition of matrices $\mathbf{Q}$ and $\mathbf{R}$ in (29) and (30), one can see zero entries on the diagonal corresponding to the displacement quadratic term. This brings conservatism in the choice of matrix L_{rd}, next section proposes an extension of the Lyapunov function (24) to handle this issue.

IV. REDUCTION OF CONSERVATISM

A. *Choice of Lyapunov function*

Adding parameters in the Lyapunov function reduces the conservatism of the results obtained using the previous Lyapunov function. The following result holds:

Theorem 1: *The following functions are Lyapunov functions for system* (9)*:*

$$\mathbf{V}(x) = x^T \begin{pmatrix} (1+\epsilon)M & \epsilon M \\ \epsilon M & (1+\epsilon)K + \epsilon D \end{pmatrix} x \tag{32}$$

for all scalar ϵ such that:

$$0 < \epsilon < \frac{\alpha}{1-\alpha} \tag{33}$$

where α is the mass-damping coefficient of the material (see appendix for details).

In open-loop, the derivative of $\mathbf{V}(x)$ becomes:

$$\dot{\mathbf{V}}(x) = 2x^T \begin{pmatrix} -(1+\epsilon)D + \epsilon M & 0 \\ 0 & -\epsilon K \end{pmatrix} x \tag{34}$$

By adding parameter ϵ in the Lyapunov function, we add a non-zero entry on the diagonal of its derivative.

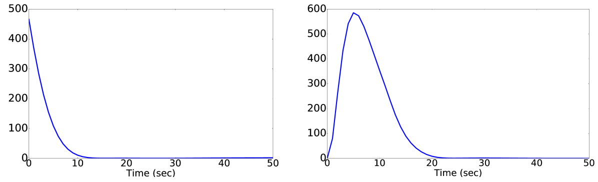

Fig. 5. Results in closed-loop using feedback gains computed thanks to eq (31).
Left : Norm of the displacement (cm); Right: Norm of velocity (cm/s^2).

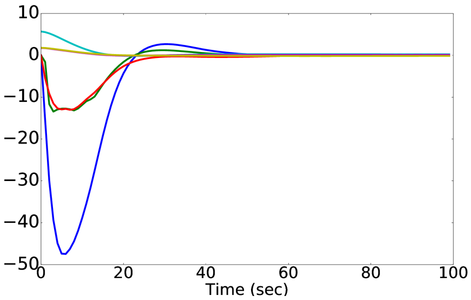

Fig. 6. Reduced order state in closed-loop using eq (31)

B. Closed-loop algorithm

The derivative of the Lyapunov function according to the trajectories of the closed-loop writes:

$$\dot{\mathbf{V}}(x) = x^T \left(\begin{pmatrix} -(1+\epsilon)D + \epsilon M & 0 \\ 0 & -\epsilon K \end{pmatrix} + \begin{pmatrix} (1+\epsilon)H^T \\ \epsilon H^T \end{pmatrix} L \right) x + (*) \tag{35}$$

which in the projected space is equivalent to:

$$\Xi^T \mathcal{V}^T \left(\begin{pmatrix} -(1+\epsilon)D + \epsilon M & 0 \\ 0 & -\epsilon K \end{pmatrix} + \begin{pmatrix} (1+\epsilon)H^T \\ \epsilon H^T \end{pmatrix} L \right) \mathcal{V}\Xi + (*) \tag{36}$$

from which the design of the controller is made possible thanks to the following theorem:

Theorem 2: *System* (9) *with feedback* (26) *is stable with a decay rate* λ *if:*

$$(\mathbf{S} + \mathbf{T}) < -\lambda \begin{pmatrix} (1+\epsilon)M & \epsilon M \\ \epsilon M & (1+\epsilon)K + \epsilon D \end{pmatrix} \tag{37}$$

with

$$\mathbf{S} = \mathcal{V}^T \begin{pmatrix} -(1+\epsilon)D + \epsilon M & 0 \\ 0 & -\epsilon K \end{pmatrix} \mathcal{V} + (*) \tag{38}$$

which also writes

$$S = \begin{pmatrix} -(1+\epsilon)D_{rv}^{rv} + \epsilon M_{rv}^{rv} & 0 & -(1+\epsilon)D_{rv}^{\bar{r}v} + \epsilon M_{rv}^{\bar{r}v} & 0 \\ 0 & -\epsilon K_{rv}^{rv} & 0 & -\epsilon K_{rv}^{\bar{r}v} \\ -(1+\epsilon)D_{\bar{r}v}^{rv} + \epsilon M_{\bar{r}v}^{rv} & 0 & -(1+\epsilon)D_{\bar{r}v}^{\bar{r}v} + \epsilon M_{\bar{r}v}^{\bar{r}v} & 0 \\ 0 & -\epsilon K_{\bar{r}v}^{rv} & 0 & -\epsilon K_{\bar{r}v}^{\bar{r}v} \end{pmatrix}$$

where $D_{rv}^{rv} = \mathcal{V}_{rv}^T D \mathcal{V}_{rv}$, $D_{rv}^{\bar{r}v} = \mathcal{V}_{rv}^T D \mathcal{V}_{\bar{r}v}$ *and*

$$\begin{aligned} \mathbf{T} &= \mathcal{V}^T \begin{pmatrix} (1+\epsilon)H^T \\ \epsilon H^T \end{pmatrix} L\mathcal{V} + (*) \\ &= \begin{pmatrix} (1+\epsilon)\mathcal{V}_{rv}^T H^T L_{rv} & (1+\epsilon)\mathcal{V}_{rv}^T H^T L_{rd} & 0 & 0 \\ \epsilon \mathcal{V}_{rd}^T H^T L_{rv} & \epsilon \mathcal{V}_{rd}^T H^T L_{rd} & 0 & 0 \\ (1+\epsilon)\mathcal{V}_{\bar{r}v}^T H^T L_{rv} & (1+\epsilon)\mathcal{V}_{\bar{r}v}^T H^T L_{rd} & 0 & 0 \\ \epsilon \mathcal{V}_{\bar{r}d}^T H^T L_{rv} & \epsilon \mathcal{V}_{\bar{r}d}^T H^T L_{rd} & 0 & 0 \end{pmatrix} + (*) \end{aligned} \tag{39}$$

Remark 2: Sketch of the proof.
$V(x)$ is a Lyapunov function defined in Theorem 1 and it holds:

$$(37) \Leftrightarrow \dot{\mathbf{V}}(x) < -\lambda \mathbf{V}(x) \tag{40}$$

Moreover, $L_r = 0$ is solution of the previous LMI for $\lambda = 0$ which makes it possible to use optimization algorithms. □

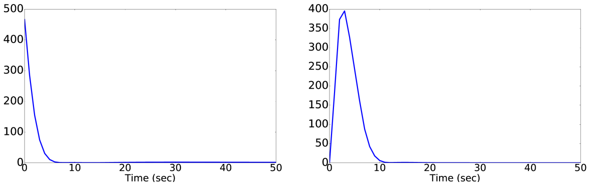

Fig. 7. Results in closed-loop using feedback gains computed thanks to eq (37).
Left : Norm of the displacement (cm); Right: Norm of velocity (cm/s^2).

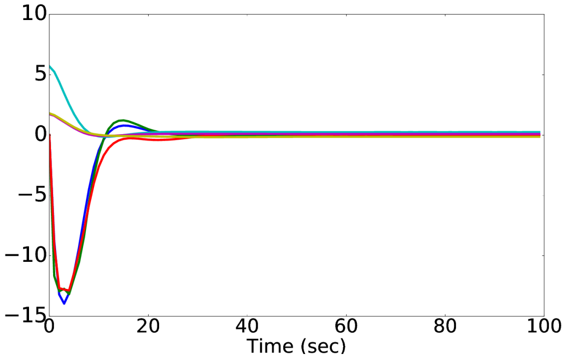

Fig. 8. Reduced order state in closed-loop using eq (37)

Using the Lyapunov function $\mathbf{V}(x)$, more flexibility is given in the choice of the matrix L_r than in (31) with the same number of decision variables.

C. Simulation Experiments

The same experiment is done with the robot presented on figure 1 but with the control method of this section, i.e. Theorem 2. The objective remains the same, the oscillations of the open-loop behaviour of figure 3 and 4 should be attenuated by the controller. The closed-loop results are presented on figures 7 and 8.

It is clear that the oscillations are removed with this control law, and in this case the tuning of the decay rate of the Lypunov function is easier than with the controller defined in (31). During firsts experiments whose results are shown on figure 5, the displacement vector converges after $t = 11s$ whereas on figure 7, it converges at $t = 6s$. A more complete control of the robots dynamics is also made possible using the method presented in this section. The large scale Lyapunov function $\mathbf{V}(x)$ guarantees the stability of the large scale model but here both reduced velocity and displacement have a direct impact on the closed-loop performance. We also have more flexibility in the tuning of the controller.

Remark 3: The resolution of the LMI (37), with 48 variables and 9342×9342 constraints took 75 minutes on a Intel Core i7 CPU.

V. CONCLUSIONS

We presented a generic method that offers the possibility to control the dynamical behaviour of soft robots. The first benefit of this work is the use of model order reduction methods to provide the user with a reduced order system that models a large-scale accurate model of the robot. Thanks to the reduction of the state variable, the design of a reduced order state feedback is made possible. The second advantage of our work is the use of Lyapunov function to prove the stability of the large-scale closed-loop system. Simulation results provided in this paper show the performance of our approach, which is generic in the sense that it is applicable to any robot with a stable equilibrium point, as long as a FEM mesh of the robot is obtained.

A few steps are still required before testing on real prototypes, as the one presented on figure 1. Our next move is the integration of an observer in the control design to make it possible to reconstruct the reduced state from measurements.

For now, the control design allows the user to control a studied soft robot from any initial shape to a desired position, where the model has been linearized. An extension of the approach would be interesting to be able to control the robot from any initial shape to any desired position. Further research could focus on the linearization assumption required in the method, removing the linearization step to design a controller for the initial non-linear system (8) could also lead to a controller where the region of convergence is guaranteed.

APPENDIX : PROOF OF THEOREM 1

The continuous function $\mathbf{V}(x)$ is a Lyapunov function in open-loop for system (9) if:

1) $V(x) > 0$
2) $V(x)$ is radially unbounded
3) $\dot{V}(x) < 0$

- Proof of 1)

$$\mathbf{V}(x) > 0 \Leftrightarrow \begin{pmatrix} (1+\epsilon)M & \epsilon M \\ \epsilon M & (1+\epsilon)K + \epsilon D \end{pmatrix} > 0$$

Using Schur complement, it is equivalent to:

$$(1+\epsilon)M > 0 \;;\; (1+\epsilon)K + \epsilon D - \frac{\epsilon^2}{1+\epsilon}M > 0$$

By definition, matrices M, K and D are positive definite. The damping matrix is defined using Rayleigh definition:

$$D = \alpha M + \beta K$$

where α and β are respectively the mass-proportional and the stiffness-proportional damping coefficients of the material. Both are positive scalars lower than one. It holds:

$$\mathbf{V}(x) > 0$$

$$\Leftrightarrow \epsilon > -1;\; (1+\epsilon)K + \epsilon(\alpha M + \beta K) - \frac{\epsilon^2}{1+\epsilon}M > 0$$

$$\Leftrightarrow \epsilon > -1;\; (1+\epsilon+\epsilon\beta)K + \epsilon(\alpha - \frac{\epsilon}{1+\epsilon})M > 0$$

Sufficient conditions are:

$$\begin{cases} \epsilon > -\dfrac{1}{1+\beta} \Rightarrow (1+\epsilon+\epsilon\beta)K > 0 \\ 0 < \epsilon < \dfrac{\alpha}{1-\alpha} \Rightarrow \epsilon(\alpha - \dfrac{\epsilon}{1+\epsilon}) > 0 \end{cases}$$

The following condition is also a sufficient condition for V(x) to be positive definite in open-loop for system (9):

$$0 < \epsilon < \frac{\alpha}{1-\alpha} \tag{41}$$

- Proof of 2) is trivial.
- Proof of 3)

$$\dot{V}(x) < 0 \Leftrightarrow -2\begin{pmatrix} (1+\epsilon)D - \epsilon M & 0 \\ 0 & \epsilon K \end{pmatrix} < 0$$

With $\epsilon > 0$, it directly follows $\epsilon K > 0$. One just need $(1+\epsilon)D - \epsilon M > 0$, for which condition (41) is also a sufficient condition.

REFERENCES

[1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, 2015.

[2] C. Majidi, "Soft robotics: A perspective—current trends and prospects for the future," *Soft Robotics Journal, Volume 1*, 2013.

[3] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: A bioinspired evolution in robotics," vol. 31, 04 2013.

[4] F. Renda *et al.*, "Discrete Cosserat approach for multi-section soft robots dynamics," *arXiv preprint arXiv:1702.03660*, 2017.

[5] V. Falkenhahn *et al.*, "Dynamic modeling of bellows-actuated continuum robots using the Euler–Lagrange formalism," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1483–1496, 2015.

[6] P. Benner and J. Saak, "Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey," *GAMM-Mitteilungen*, vol. 36, no 1, p. 32-52, 2013.

[7] G. B. Koo, J. B. Park, and Y. H. Joo, "Decentralized fuzzy observer-based output-feedback control for nonlinear large-scale systems: an LMI approach," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 2, pp. 406–419, 2014.

[8] W. Chang and W.-J. Wang, "H_∞ fuzzy control synthesis for a large-scale system with a reduced number of LMIs," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 1197–1210, 2015.

[9] P. Benner *et al.*, *Model Reduction and Approximation : Theory and Algorithms*. SIAM, 2017.

[10] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, 2015.

[11] A. Lismonde, V. Sonnevile, and O. Brüls, "Trajectory planning of soft link robots with improved intrinsic safety," *International Federation of Automatic Control (IFAC) 20^{th} World Congress*, 2017.

[12] Z. Zhang *et al.*, "Kinematic modeling and observer based control of soft robot using real-time finite element method," *International Conference on Intelligent Robots and Systems, IROS2016*, 2016.

[13] S. Sadati *et al.*, "Control space reduction and real-time accurate modeling of continuum manipulators using Ritz and Ritz–Galerkin methods," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 328–335, 2018.

[14] E. Coevoet *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.

[15] P. Vuillemin, C. Poussot-Vassal, and D. Alazard, "Poles residues descent algorithm for optimal frequency-limited H_2 model approximation," *European Control Conference*, 2014.

[16] K. Gallivan, A. Vandendorpe, and P. Van Dooren, "Sylvester equations and projection-based model reduction," *Journal of Computational and Applied Mathematics*, vol. 162, no. 1, pp. 213–229, 2004.

[17] P. Benner, P. Kürschner, and J. Saak, "An improved numerical method for balanced truncation for symmetric second-order systems," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 19, no. 6, pp. 593–615, 2013.

[18] S. Gugercin, "An iterative SVD-krylov based method for model reduction of large-scale dynamical systems," *Linear Algebra and its Applications*, 2008.