Real-time Control of Soft-Robots using Asynchronous Finite Element Modeling

Frederick Largilliere, Valerian Verona, Eulalie Coevoet, Mario Sanz-Lopez,
Jeremie Dequidt, Christian Duriez

HAL Id: hal-01163760

https://hal.inria.fr/hal-01163760

Submitted on 15 Jun 2015

Real-time Control of Soft-Robots
using Asynchronous Finite Element Modeling

Frederick Largilliere[3,2], Valerian Verona[1,2], Eulalie Coevoet[2],
Mario Sanz-Lopez[2], Jeremie Dequidt[3,2] and Christian Duriez[2,3]

Abstract— **Finite Element analysis can provide accurate deformable models for soft-robots. However, using such models is very difficult in a real-time system of control. In this paper, we introduce a generic solution that enables a high-rate control and that is compatible with strong real-time constraints. From a Finite Element analysis, computed at low rate, an inverse model of the robot outputs the setpoint values for the actuator in order to obtain a desired trajectory. This inverse problem uses a QP (quadratic-programming) algorithm based on the equations set by the Finite Element Method. To improve the update rate performances, we propose an asynchronous simulation framework that provides a better trade-off between the deformation accuracy and the computational burden. Complex computations such as accurate FEM deformations are done at low frequency while the control is performed at high frequency with strong real-time constraints. The two simulation loops (high frequency and low frequency loops) are mechanically coupled in order to guarantee mechanical accuracy of the system over time. Finally, the validity of the multi-rate simulation is discussed based on measurements of the evolution in the QP matrix and an experimental validation is conducted to validate the correctness of the high-rate inverse model on a real robot.**

I. INTRODUCTION

As pointed out by Trivedi *et. al.* [1] and by recent works in the field such as [2], [3], [4], soft (biomimetic) robots have gained much interest in the community due to their natural compliance, the variety of possible actuation and sensing, as well as their potential applications. One of the main challenge remains the control of these deformable robots since they have an infinite number of degrees of freedom and their design rely on coupled and *redundant*[1] actuation. Therefore standard control methods which make assumptions of rigid structures and limited degrees of freedom are not relevant in that context. Dedicated approaches for particular soft robots have been considered (for instance curvilinear robots in [5]) or generic approaches using FEM (finite element method) and solving interactively an inverse problem through an iterative Gauss-Seidel algorithm [6]. The latter approach, while generic, is limited to simple robots where the computational cost of FEM simulation could be bounded to match real-time constraints, needed by the control loop. Indeed despite recent advances in GPGPU computation for

[1]Telecom Physique Strasbourg
[2]INRIA, Shacra Team `1stname.lastname@inria.fr`
[3]University of Lille `1stname.lastname@univ-lille1.fr`
—

[1]Redundant actuation is not theoretically possible on soft robot with a infinite number of degrees of freedom. As many papers that concerns deformable robotics, redundant means that there is more actuators than the number of DOFs of the effector.

FEM simulation [7], the computation of the compliance matrix (*i.e.* the inverse of the stiffness matrix), which is a requirement for the inverse problem solving, remains the main bottleneck. Even with fast FEM simulation and thorough optimizations (matrix factorization and matrix pre-conditioning), the computation time of the control algorithm is too long to allow fast response times for the soft robot.

Research works related to medical imaging have also addressed the resolution of an inverse problem based on mechanical modeling. For instance, in the context of deformable registration images, Coevoet *et. al.* [8] use FEM simulation of anatomical structures to register intra-patient images. The FEM simulation is used by an inverse problem formulated as a QP (quadratic-programming) problem allowing to provide plausible mechanical deformations of the structures according to medical images.

The haptic rendering community has already faced a similar problem when dealing with haptic rendering of deformable objects. An high-quality haptic rendering requires fast high frequency updates (up to 1KHz) whereas the simulation of deformable objects has a much lower frequency (typically around 10-50 Hz). Several works have addressed this issue: for instance Saupin *et. al.* [9], Garre and Otaduy [10] use an approximation of the compliance matrix while other works consider reduced deformation models (Barbic and James [11]) or a fast linearized deformable model (Jacobs *et. al.* [12]). Dervaux *et. al.* [13] introduce a multi-rate simulation that can handle and mechanically couple *simple* high-frequency simulations and more complex but low-frequency simulations.

Given the similarities between the problem addressed in haptic rendering and the one related to soft-robot control, we propose in this paper to extend the generic approach of Dervaux *et. al.* [13] in the context of robot control. Compared to the work of Duriez [6], the contributions in this paper are the formulation of the inverse problem as a QP (Quadratic Programming) problem, in which the QP matrix provides the contributions of each actuator in order to move the soft robot to the desired location (section II) and the use of a multi-rate simulation in order to provide fast response time to control the robot (section III). Finally, in section IV, we present a numeric assessment for the validity of the multi-rate simulation based on the evolution of the QP matrix and we quantitatively validate, on an experiment, that the inverse model computed at high rates is correct.

The results presented in this paper illustrate that a high-frequency response of 600 Hz can be achieved while having

accurate mechanical behavior of the soft deformable robot. To our best knowledge this paper is the first that allows the computation of an inverse model of a soft robot with such a high frequency. This contribution opens, for the first time, the possibility of using a inverse FEM model of soft-robot in an embedded robot controller with strong real-time constraints.

II. QP ALGORITHM FOR SOFT-ROBOT CONTROL

This work builds on the control method of Duriez [6] which proposes to model the non-linear deformation of volumetric soft robots using the Finite Element Method, at low refresh rates. A non-standard optimization solver is used to compute the inverse model of the robot. This paper extends the approach to high refresh rates, compatible with real-time constraint of a robotic control system. Moreover, the inverse model is obtained by solving a Quadratic Programming (QP) optimization, but the rationale of our method is similar.

The method begins by obtaining a volumetric mesh of the robot (in practice, we use the open source software CGAL[2]) The finite element analysis allows to integrate the constitutive law of the deformable material over the whole structure. It leads to a quasi-static formulation of the equilibrium between internal forces inside the deformable structure $\mathbf{f}(\mathbf{x})$ (where $\mathbf{x}$ is the position vector of the FEM nodes and $\mathbf{f}(\mathbf{x})$ provides the volumetric internal stiffness forces at a given position $\mathbf{x}$ of the nodes), external loads $\mathbf{f}_{ext}$ and contributions of the actuators $\mathbf{J}^T \lambda_a$ (where $\mathbf{J}^T$ provides the direction of the effort on the nodes concerned by each actuators and λ_a is the contribution of each actuator) according to the equation:

$$\mathbf{f}(\mathbf{x}) + \mathbf{f}_{ext} + \mathbf{J}_a^T \lambda_a = 0 \qquad (1)$$

Classical resolutions of a FEM problem, such as solving the static equilibrium of the structure described at equation 1, imply a direct (or forward) problem: it allows to compute the displacements of the structure, given the values of the efforts put on the actuators λ_a. In the context of deformable robots control, the displacements of the structure are known and the main challenge is to find how much each actuator will be triggered so that the whole structure will deform to follow a desired motion or to reach a desired location. This leads to the resolution of an inverse problem for which the vector λ_a is unknown. To solve the inverse problem in an efficient way, a three-step strategy is proposed.

Step 1: the starting point is a *linearization* of the internal forces, that is performed at each step i of the computation:

$$\mathbf{f}(\mathbf{x_i}) \approx \mathbf{f}(\mathbf{x_{i-1}}) + \mathbf{K}(\mathbf{x_{i-1}})d\mathbf{x} \qquad (2)$$

where $\mathbf{K}(\mathbf{x})$ is the tangent stiffness matrix that depends on the actual position of the nodes and $d\mathbf{x}$ is the difference between positions $d\mathbf{x} = \mathbf{x_i} - \mathbf{x_{i-1}}$. The lines and columns that correspond to fixed nodes are removed from the system to improve the condition number of the matrix $\mathbf{K}$. In $\mathbf{f}$ and $\mathbf{K}$, the rows (and columns for $\mathbf{K}$) contain the component of the internal forces (x, y, z) for the nodes, and the nodes are

[2]www.cgal.org

stored in the order corresponding to their numbering in the mesh.

Step 2: the second step is to define a constraint space of a reduced size using Lagrange multiplier formulation. As described in equation 1, we can define a direction of actuation given the position of the deformable model of the robot in the matrix $\mathbf{J}_a^T$. Three types of actuators have been yet considered in our formulation, as shown in figure 1:

- In the case of a *linear* actuation, the actuator applies a force on a point of the deformable structure of the robot, or between two points of the deformable structure, both directions are considered and the unknown is the force of actuation. The construction of matrix $\mathbf{J}_a$ is based on one row for each actuator. This row contains 0 value everywhere except at the column that corresponds to the point. There, we store the direction of the actuator force.
- The *cable* actuator is similar to the linear actuation, except the actuation is unilateral (the cable can only pull) meaning that the intensity is unknown but it satisfies the inequality: $\lambda_a \geq 0$ The construction of the matrix $\mathbf{J}_a$ is similar to the *linear* actuation.
- For *pressure* actuation, the direction is not applied to a unique node of the FEM mesh: the direction is spread over all the nodes inside the cavity and depends on the area of the cavity surface. So each rows in matrix $\mathbf{J}_a$ corresponds to a cavity, but the number of non-zero value on the column is higher than the two previous cases. In that case, the unknown λ_a is homogeneous to a pressure.

In all cases, we can limit the range of effort, given the characteristic of the actuators: $a_{min} \leq \lambda_a \leq a_{max}$, and we can also define a displacement limit to model the course of the actuator $l_{min} \leq \delta_a \leq l_{max}$ where δ_a is the relative displacement of the actuator in the constraint space.

Finally, some Lagrange multipliers are also defined for the terminal effector using matrix $\mathbf{J}_e$. A point or a set of points of the deformable robot needs to be considered as the end effector. For each point, the constraint objective is to reduce the shift (noted δ_e) between the end effector position and a desired position p_{des}. For now, we suppose that there is no load applied on the effector, so that $\lambda_e = 0$.

Step 3: the size of matrix $\mathbf{K}$ is often very large (related to the number of nodes) so an optimization in the motion space would be computationally very expensive. Instead, this step is based on the computation of the Schur complement (also called the Delassus operator) of the constraint problem:

$$\delta_e = \underbrace{\left[\mathbf{J}_e \mathbf{K}^{-1} \mathbf{J}_a^T\right]}_{\mathbf{W}_{ea}} \lambda_a + \delta_e^{\text{free}} \qquad (3)$$

$$\delta_a = \underbrace{\left[\mathbf{J}_a \mathbf{K}^{-1} \mathbf{J}_a^T\right]}_{\mathbf{W}_{aa}} \lambda_a + \delta_a^{\text{free}} \qquad (4)$$

Where δ_e^{free} and δ_a^{free} are, respectively, the gap between the effector position and the desired position p_{des}, and the displacement of the actuators when the effort of the actuator
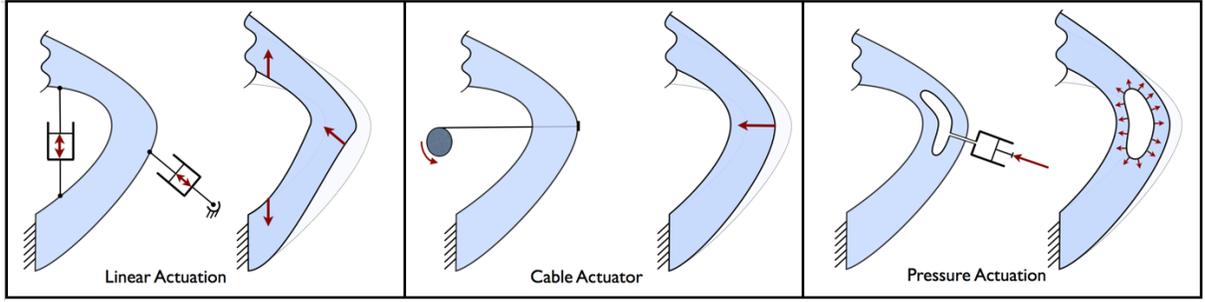
Fig. 1. Three types of actuation that are considered in this study: (a) Linear actuation, (b) cable transmission (tendon), (c) pneumatic actuator.

vanishes $\lambda_a = 0$, Matrices $\mathbf{W}_{ea}$ and $\mathbf{W}_{aa}$ are homogeneous to a compliance. Using $\mathbf{W}_{ea}$, we can get a measure of the mechanical coupling between effector and actuator, and with $\mathbf{W}_{aa}$, the coupling between actuators.

Convex optimization: the main novelty compared to [6] is the use of a convex optimization. The goal of the optimization is to find how to actuate the structure so that it the end effector of the robot reaches a desired position. It consists in reducing the norm of $\boldsymbol{\delta}_e$ which measures the shift between the end effector and its desired position. Thus, the computation of $min(\frac{1}{2}\boldsymbol{\delta}_e^T\boldsymbol{\delta}_e)$ can be done by setting a Quadratic Programming (QP) problem:

$$min\left(\frac{1}{2}\boldsymbol{\lambda_a}^T\mathbf{W}_{ea}^T\mathbf{W}_{ea}\boldsymbol{\lambda_a} + \boldsymbol{\lambda_a}^T\mathbf{W}_{ea}^T\boldsymbol{\delta}_e^{\text{free}}\right) \quad (5)$$

$$subject\ to\ \text{(course of actuators)}:$$
$$l_{min} \le \boldsymbol{\delta}_a = \mathbf{W}_{aa}\boldsymbol{\lambda_a} + \boldsymbol{\delta}_a^{\text{free}} \le l_{max}$$
$$and\ \text{(case of unilateral effort actuation)}: \quad (6)$$
$$a_{min} \le \boldsymbol{\lambda_a} \le a_{max}$$

The use of a minimization formulation allows to find an acceptable solution even when the desired position is out of the workspace of the robot. In such case, the algorithm will find the point that minimizes the distance with the desired position while matching the limits introduced for the course of the actuators.

This type of problem can be solved by a variety of methods. In practice, we use the QP solver available in the CGAL library [14]. The matrix of the QP, $\mathbf{W}_{ea}^T\mathbf{W}_{ea}$ is symmetric. If the number of actuators is equal or less than the size of the effector space, the matrix is also definite. In that case, the solution of the minimization is unique.

In the opposite case, i.e, when the number of actuators is greater than the degrees of freedom of the effector points (redundant actuation), the matrix of the QP is only semi-positive. Consequently, the solution can sometimes be non-unique. In such case, some QP algorithms are able to find one solution among all possible solutions [15].

Thus, the main benefit of this QP formulation over the previous formulation in [6] is the possibility of having an objective analysis of the existence and uniqueness of the solution. Moreover, as QP is an usual formulation, efficient implementation of solvers are available, making the computation of an inverse model very straight-forward.

III. MULTIRATE SIMULATION

The main limitation of the method is due to the computation costs needed by the Finite Element Model and by the compliance matrices $\mathbf{W}$. Using an optimized linear solver, we could achieve a refresh rate of $15Hz$ for a model of the robot composed of 1628 Nodes and 4147 elements (see figure 3), which provides a very good precision on static positions (1.4mm of mean error between the desired position and the actual position of the robot). However, such rates are insufficient in a standard real-time controller of a robot. It could lead to issues when the velocity of the desired motion increases, because of the low frequency update of the inverse model. Moreover, it reduces the spectrum of actuators that can be directly controlled by the method. Finally, in the perspective of using such method in a closed loop, we need to have some guarantee on the computation time taken by the model.

To overcome this limitation, we propose to add a multirate strategy that takes advantage of the sequence used in the three steps strategy described above. From a computational standpoint, these three steps induce a loop that repeats the following sequence:

1) acquisition of the new desired position of the effector $\mathbf{x}$ and computation of $\boldsymbol{\delta}_e^{\text{free}}$
2) estimation of the tangent stiffness matrix $\mathbf{K}(\mathbf{x})$ (equation 7)
3) computation of the compliance matrices $\mathbf{W}_{ea}$ and $\mathbf{W}_{aa}$ (equations 3 and 4)
4) setting-up the QP problem (equations 5 and 6)
5) resolution of the QP problem to find λ_a
6) send the λ_a to the actuators of the soft robot
7) use the λ_a to update the simulated position of the effector according to equation 1

From a simulation standpoint, the softness of the material used for the robot does not require a small simulation time-step in order to be accurate. This is a consequence of the fact that soft materials have slow dynamics and are not subject to high-frequency deformation that needs to be captured.

This is illustrated by the fact that a refresh rate of $15Hz$ provides an acceptable precision on static positions and therefore we can make the assumption that the tangent stiffness matrix $\mathbf{K}(\mathbf{x})$ does not significantly evolve in this $66.6msec$ interval between two simulation updates. The
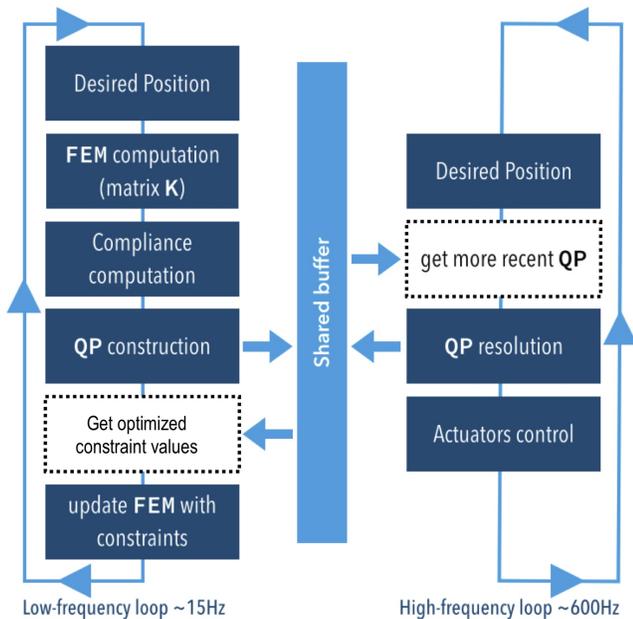
Fig. 2. Overview of our multi-rate algorithm that combines two loops: one at a low frequency (around $15Hz$) and another at a higher frequency (around $600Hz$). Data are exchanged through a shared buffer. Plain blocks indicate computations that are performed by the loop while hollow blocks indicate data retrieved from the other loop (through the shared buffer).

constraint space definition relies on the robot and therefore is fixed by design meaning that $\mathbf{J}_e$ and $\mathbf{J}_a$ are constant and therefore that the matrices $\mathbf{W}_{ea}$ and $\mathbf{W}_{aa}$ are valid between two simulation updates and will not evolve much in the next step.

Since only $\boldsymbol{\delta}_e^{\text{free}}$ can possibly be modified at high frequency rates, updating and solving the QP problem should be conducted at high frequency. As a consequence, this sets the rationale of our multi-rate simulation where there are two loops: a low-frequency loop that will be in charge of computationally-intensive algorithms (steps 2, 3, 4 and 7 of the sequence) with a refresh rate of $15Hz$ and a high-frequency rate that will only deal with solving the QP problem and sending efforts to the robot at the highest frequency possible. We extended the approach of Dervaux *et. al.* [13] by having these two loops in separate threads that communicate through shared buffers. The figure 2 illustrates the whole process with the two loops and the data that are sent to / received from the shared buffer.

One feature of this multi-rate algorithm is the fact that we can update the QP with the latest desired position very quickly and obtain a new optimized value of the inverse problem in a few ms. The main benefit, in the perspective of using this method in a real-time robotic controller, is that only a reduced part (the one at high rates) of the algorithm would be really subject to hard real-time constraints. This is an major practical step because, otherwise, it would be very complicated to have some strong warranty on the execution time of a hole FEM model.

Fig. 3. The deformable robot made of silicone is registered with its FEM model [6]. The robot is actuated with cables that are placed on the four legs of the robot, and that are pulled by servo-motors. We did some additional measurement on this robot to validate our multirate approach.

IV. FIRST VALIDATION AND DISCUSSION

To validate the hypothesis made on the small variation in the tangent stiffness matrix, we have conducted experimental measurements on the similarity of the QP matrix $\mathbf{W}_{ea}^T \mathbf{W}_{ea}$ from one step to the following one. As $\mathbf{W}_{ea}$ directly depends on the inverse of the FEM matrix $\mathbf{K}$, we can then quantify the validity of our hypothesis about this matrix not evolving significantly between two steps of the generic simulation. In order to get an indicator of the current state and composition of the QP matrix, we compute the infinity-norm of this matrix defined as :

$$\|\mathbf{M}\|_\infty = \max_i \sum_j |m_{ij}| \qquad (7)$$

We then compare this norm from one step to the following one by computing its rate of variation as a percentage value.

We evaluate the maximum value of this rate on a virtual model of the robot presented on figure 3 and we reproduce this measurement with the same simple line trajectory but different velocities, in order to see a possible influence of the displacement speed. The results obtained are presented in the following table :

Speed of displacement (mm/s)	Max. rate of variation in $\|\mathbf{M}\|_\infty$ (%)
6.25	1.2
12.5	2.97
25	5.28
37.5	5.66
50	5
62.5	9.09
75	7.34
87.5	8.94
100	10.67
112.5	12.52
125	38
150	38.01

We can see in this table that while this indicator of the variation in the QP matrix doesn't contain full knowledge on the modification of the matrix, it already presents an interesting information on how much the speed range can

influence the validity of our asynchronous FEM method. In order to stay in an acceptable level of validity, one can consider that a maximum rate of variation of 10% should be a limit, which would translate in a maximum speed of 100mm/s.
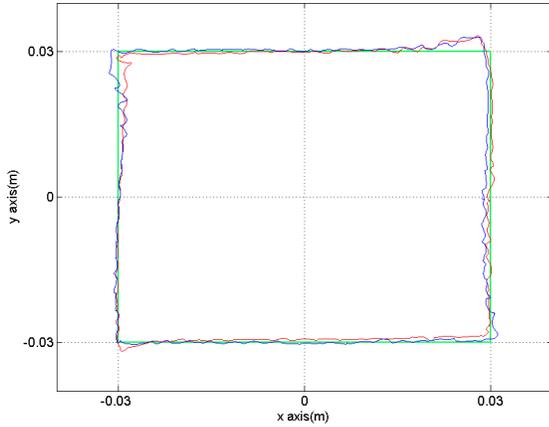

Fig. 4. Trajectories of the effector point under the asynchronous simulation (red) and the standard simulation (blue) compared to the desired trajectory (green).

We have conducted an experimental validation of the asynchronous algorithm. Namely, we compare an asynchronous control strategy with a control based on the QP output from the full simulation, for the soft robot (actuated through cables pulled by servo-motors) shown in figure 3. The robot itself was casted with soft silicone rubber in a 3D printed mold which ensures the geometrical accuracy between the simulated and the actuated robot. Following the method described in [6], the Young Modulus of the FEM model was set to *150kPa*. Both control strategies are in open loop: we directly set the values (computed by the QP solvers) to the actuators so that we can verify the validity of employed inverse models.
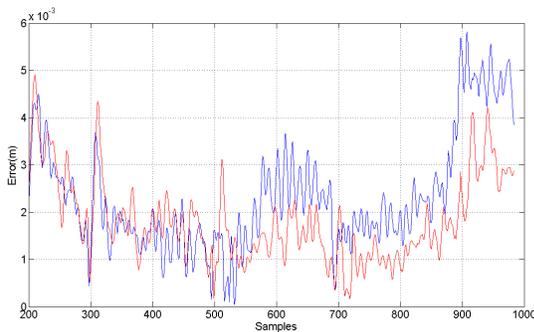

Fig. 5. Euclidean error on the trajectory of the effector point for the asynchronous simulation (red) and the generic simulation (blue) compared to the desired trajectory.

The figure 4 presents the displacement of the end effector point under a predefined square trajectory. We can observe that both trajectories are very close to the desired one. It

means that all the intermediate values that are computed at high rates using the asynchronous QP are correct. This is a first step of the asynchronous method validation that does not introduce any perturbation. The figure 5 presents the error on trajectory of the two simulations. While the average error for the generic simulation is 2.33 mm, the asynchronous simulation reduces it to 1.82 mm. This is a consequence of the more frequent update of position in the second case. The generic method also suffers of a simulation time step that doesn't stay constant and that induces an accumulative error over time. The trajectory is then slightly smoother for the asynchronous simulation than for the non asynchronous one but some oscillations remain: These oscillations are due to the structure and the particularly low rigidity of this soft robot compared to its mass. Furthermore, neither the FEM model or the QP algorithm consider yet the dynamic behavior of the material in order to compensate such vibration.

We can therefore conclude that the asynchronous decoupling method provides adequate result to the control, that allows to follow the trajectory. But the benefit for this particular robot is not very clear due to natural vibration of the device.
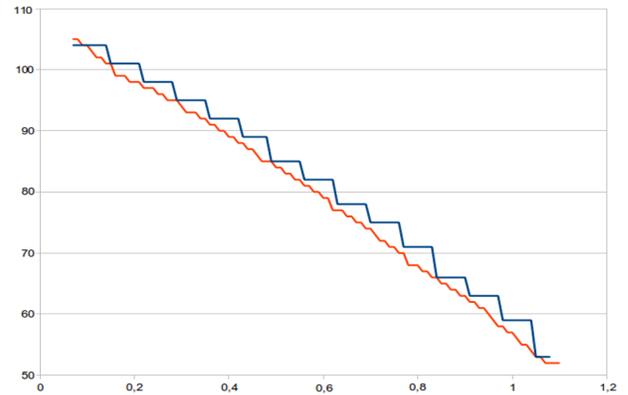

Fig. 6. Input signal sent to one actuator under the asynchronous simulation (red) and the generic simulation (blue) with respect to time (in s). Input signal sent by the asynchronous simulation is smoother than in the case of the generic simulation.

The figure 6 shows the input signal provided by the asynchronous simulation and the generic simulation to one of the servo-motors that pull the cables. We can clearly observe that the asynchronous simulation provides smoother signal than the standard simulation. If the structure of the robot were stiffer, it could clearly have more influence on the trajectory. Here, the robot naturally filters the tremor of the actuators. For a more convincing evaluation, we will investigate the needs of high update rates in the control according to the type of device. We could compare the results obtained on several devices with the same shape but with different mass / stiffness ratios and also with different type of actuators.

V. CONCLUSION AND FUTURE WORK

This paper presents several contributions for the modeling and control of soft-robots using FEM. We show that the formulation of an inverse model (as a QP optimization) can be extracted from an interactive simulation computed at low rates. This allows for a multirate solving of the inverse problem and opens the perspective of a truly real-time implementation of such inverse model, which is necessary for further studies on feedback control.

In the short term, we will investigate the use of stiffer materials to test the control in position and we will complete the development of a stiffness-controlled haptic rendering device that makes use of the methods presented in this paper.

From broader perspective, the paper opens a wide range of possibilities in haptics, with new types of devices based on soft-robotic technology, piloted by FEM simulation.

REFERENCES

[1] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics and Biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.

[2] S.-W. Yeom and I.-K. Oh, "A biomimetic jellyfish robot based on ionic polymer metal composite actuators," *Smart materials and structures*, vol. 18, no. 8, 2009.

[3] Y. Bar-Cohen, "Focus issue on biomimetics using electroactive polymers as artificial muscles," *Bioinspiration & Biomimetics*, vol. 2, no. 2, 2007.

[4] C. Menon and M. Sitti, "Biologically inspired adhesion based surface climbing robots," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*.   IEEE, 2005, pp. 2715–2720.

[5] L. G. Torres and R. Alterovitz, "Motion planning for concentric tube robots using mechanics-based models," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*.   IEEE, 2011, pp. 5153–5159.

[6] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*.   IEEE, 2013, pp. 3982–3987.

[7] J. Allard, H. Courtecuisse, F. Faure, *et al.*, "Implicit fem solver on gpu for interactive deformation simulation," *GPU Computing Gems Jade Edition*, pp. 281–294, 2011.

[8] E. Coevoet, N. Reynaert, E. Lartigau, L. Schiappacasse, J. Dequidt, and C. Duriez, "Introducing interactive inverse FEM simulation and its application for adaptive radiotherapy," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2014 - 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part II*, 2014, pp. 81–88.

[9] G. Saupin, C. Duriez, and S. Cotin, "Contact model for haptic medical simulations," in *Biomedical Simulation*.   Springer, 2008, pp. 157–165.

[10] C. Garre and M. A. Otaduy, "Haptic rendering of complex deformations through handle-space force linearization," in *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*.   IEEE, 2009, pp. 422–427.

[11] J. Barbic and D. L. James, "Six-dof haptic rendering of contact between geometrically complex reduced deformable models," *Haptics, IEEE Transactions on*, vol. 1, no. 1, pp. 39–52, 2008.

[12] P. Jacobs, M. J. Fu, and M. C. Çavuşoğlu, "High fidelity haptic rendering of frictional contact with deformable objects in virtual environments using multi-rate simulation," *The International Journal of Robotics Research*, 2010.

[13] F. Dervaux, I. Peterlik, J. Dequidt, S. Cotin, and C. Duriez, "Haptic rendering of interacting dynamic deformable objects simulated in real-time at different frequencies," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*.   IEEE, 2013, pp. 2010–2016.

[14] A. Fabri and S. Pion, "Cgal: the computational geometry algorithms library," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*.   ACM, 2009, pp. 538–539.

[15] F. Sha, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming in support vector machines," in *NIPS*, 2002, pp. 1041–1048.