



Inverse real-time Finite Element simulation for robotic control of flexible needle insertion in deformable tissues

Yinoussa Adagolodjo, Laurent Goffin, Michel de Mathelin, Hadrien Courtecuisse

► To cite this version:

Yinoussa Adagolodjo, Laurent Goffin, Michel de Mathelin, Hadrien Courtecuisse. Inverse real-time Finite Element simulation for robotic control of flexible needle insertion in deformable tissues. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016) , Oct 2016, Daejeon, South Korea. hal-01353925

HAL Id: hal-01353925

<https://hal.archives-ouvertes.fr/hal-01353925>

Submitted on 1 Sep 2016

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.

Inverse real-time Finite Element simulation for robotic control of flexible needle insertion in deformable tissues

Yinoussa Adagolodjo, Laurent Goffin, Michel de Mathelin, Hadrien Courtecuisse
AVR Team-Project, CNRS Strasbourg and Strasbourg University

Abstract—This paper introduces a new method for automatic robotic needle steering in deformable tissues. The main contribution relies on the use of an inverse Finite Element (FE) simulation to control an articulated robot interacting with deformable structures. In this work we consider a flexible needle, embedded in the end effector of a 6 arm Mitsubishi RV1A robot, and its insertion into a silicone phantom. Given a trajectory on the rest configuration of the silicone phantom, our method provides in real-time the displacements of the articulated robot which guarantee the permanence of the needle within the predefined path, taking into account any undergoing deformation on both the needle and the trajectory itself. A forward simulation combines i) a kinematic model of the robot, ii) FE models of the needle and phantom gel iii) an interaction model allowing the simulation of friction and puncture force. A Newton-type method is then used to provide the displacement of the robot to minimize the distance between the needle's tip and the desired trajectory. We validate our approach with a simulation in which a virtual robot can successfully perform the insertion while both the needle and the trajectory undergo significant deformations.

I. INTRODUCTION

Needle insertion in soft-tissues is widely performed during minimally invasive procedures, both for diagnosis and treatment of particular diseases. Contrary to traditional open surgery, needle-based approaches only affect a localized area around the needle, therefore reducing significantly the occurrence of traumas and risks of complications[4]. Nevertheless, the effectiveness of the treatment is highly dependent on the accuracy of the needle positioning on the target lesion. This can be particularly challenging due to the fact that needles are manipulated from outside the patient and that often these procedures rely on radiological image guidance, which is known to provide a limited visibility.

Robotic manipulators have the potential to overcome limitations coming from human factor, for instance by filtering any operator's tremors. Moreover the use of robotized assistance can contribute to reduce exposure to ionizing radiation during X-ray guided procedures. Needle insertion devices have already been developed for numerous applications such as: liver cancer, breast, and biopsies.

The deformable properties of living tissues make complicated the needle insertion procedures. Indeed, such deformations can cause a displacement of the target and produce significant deflection of flexible needle. For the aforementioned reasons, a remaining limitation for robotic needle insertion is the lack of control models when the needle is inserted in deformable structures.

Problem Statement: The contribution of this paper is an

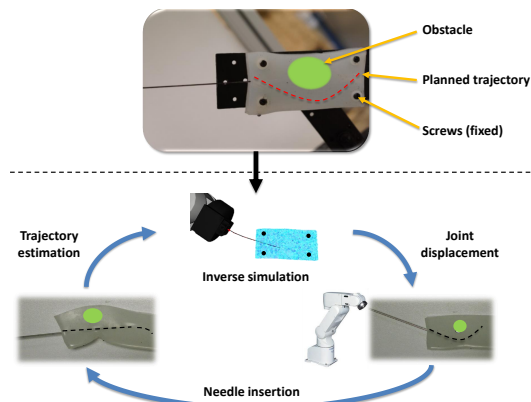


Fig. 1. Robotic needle insertion controlled by an inverse FE simulation. A predefined trajectory is given as input of the system. An inverse simulation locally find displacement of the robot to process the needle insertion along the desired path. Global deformations are applied on the gel in order to align needle's shaft with the desired trajectory.

inverse real-time FE simulation to control an articulated robot in order to take into account deformations of the structures during needle insertion. Our goal is to insert a flexible needle in a deformable volume, following a pre-defined trajectory (see fig 1). The flexible needle is embedded in the end-effector of the robot, which is considered as being infinitely rigid. The gel is fixed onto a support, within the working space of the robot, and it is considered to be deformable. It is important to notice that, although the trajectory is completely defined at the initial step, the path followed by the needle during insertion will not remain constant. As soon as the gel becomes deformed the trajectory has to be updated.

We emphasize that the goal of this paper is not to provide an achievable or optimal trajectory. Instead we assume that the trajectory is known (for example previously defined by an operator, or by a trajectory planning system that can rely on biomechanical models to enforce the attainability of the trajectory). Our goal is rather to develop a real-time inverse FE simulation, that allows the control of a robot which performs a needle insertion in a deformable mean, taking into account any deformation that occurs.

II. RELATED WORKS

Needle steering in soft tissues is a highly challenging topic of research. Although robotic needle insertion has been a subject of considerable interest over the last few years, recent surveys [1], [4], [13], [8] still underline the need for control models when dealing with deformable tissues. In this

section we report on important related work focusing on the interactions and modeling of deformable structures.

Telemanipulated systems can filter the operator's tremors, scale clinician's movements to enhance their accuracy, and reduce X-ray exposure for the medical staff. In [20] a 6 DoF haptic controller provides haptic cues on the ideal position and orientation of the needle, while a slave linear stage-coupled with a 3D ultrasound tracking device processes needle insertion in a silicone gel. In [22], an X-Y robotic system was developed for prostate brachytherapy. The robot allows the shifting and insertion of the needle along the Z direction. Due to the proximity with the skin and to the small displacements of anatomical structures during the insertion, prostate brachytherapy is another interesting application for roboticists. Piccin et al. [21] developed a robot for percutaneous interventions in liver surgery, compatible with CT-scanner constraints. In order to compensate for the patient's movements, the robot is directly mounted onto the patient himself. An important aspect of this work is the ability of the robot to release the needle in the case of extreme physiological displacements to avoid skin rupture or tissue cutting. In [11], a robotized needle insertion system using stereoscopic cameras is proposed for biopsy in small animals. Many issues could appear due to the registration of the robot with the cameras and the localization of the target in the preoperative CT-scans. However, deformations can be avoided by inserting the needle with a sufficiently high speed; this facilitates the penetration and decreases the risk of organs motions.

Image-guided techniques extract information from vision sensors to control the motion of robots and compensate for any deformations. [19] uses X-ray fluoroscopy to 3D align a needle, held by a medical robot, inside a porcine kidney. In [15], Kobayashi et al. develop an ultrasound-guided needle insertion manipulator combined with a physics-based model of the liver. After registration, the biomechanical model provides information of the stress inside the tissue during the insertion. Other approaches exploit the possibility of changing the curvature of beveled-tip flexible needles, rotating the base of the needle itself. Krupa [16] proposes a duty-cycling robotized system for the 3D steering of beveled needles, which allows an additional 3 DoF control of the tip. Using visual servoing, such a method does not require a trajectory pre-planning but it allows the creation of complex non-straight trajectories to reach a target and avoid obstacles. Instead of simply manipulating the needle, [18] proposes the deform the tissue in order to displace the target. The method combines real-time fluoroscopic images with a robotic end-effector, to manipulate breast tissue and displace a tumor towards the needle path. An important limitation of image-guided techniques is that both the needle and in the target must be constantly visible the images, which is not always possible due to imaging intrinsic constraints.

Mechanical simulations can predict the behavior of needles and tissues for a given load and deformation [2]. Finite element methods (FEM) provide high biomechanical realism, mainly because the soft-tissues behavior is directly

explained through constitutive relations. Khadem et al. [14] introduce a mechanics-based model for the simulation of needle insertion in soft tissues. The proposed model is based on beams' theory, allowing the prediction of the deflection of the needle. Tests are conducted to identify the parameters of the model, where the speed of the needle base can be used as input of a control command to steer the needle itself. A FE model for the simulation of needle insertion in soft tissue was proposed in [7]. This model allows for the simulation of tissue deformations, needle-tissue friction, and puncture force. Real-time computations are enabled avoiding expensive re-meshing operations when the needle is inserted. [12] extended the method for optimal trajectory planning for liver surgeries. The method takes into account deformations (breathing, needle deformation, friction) to avoid the obstacles. Nevertheless, some significant changes may be introduced between pre-operative and intra-operative configurations, invalidating the chosen trajectory.

Recent studies aim at deriving a physics-based model for path planning and path correction of flexible needles in soft tissues. [5] proposed to numerically derive the manipulation Jacobian matrix for tissue deformation and needle deflection. The method is combined with some trajectory planning strategies: attractive fields drive the needle towards the desired target whereas repulsive fields avoid obstacles. The method provides a velocity value for the base of the needle, that is then used in an open robotic control loop. Since real time computations are necessary for the dynamic control of a robot, [10] proposed to model soft-tissues with springs whose stiffness coefficients vary along the length of the needle. Although these methods allow fast path planning and needle insertion into viscoelastic tissues, they are limited to the cases of planar insertion and linear models. More advanced FE simulations were used in [17] for the position control of soft robots. An inverse problem based on a QP (quadratic-programming) algorithm is used to solve equations of motion and control a soft-robot in real-time. However, interactions between deformable structures (needle and gel) and complex behaviors, such as non linear friction, are still not possible.

The central contribution of this work is the use of an inverse real-time FE simulation to compute the joint displacements of an articulated robot, locally minimizing the distance between the needle tip and a desired trajectory. The overall system (robot, needle and gel) can be seen as a deformable robot with an infinite number of degrees of freedom. Our method provides inputs command to control in real-time a robotic system, taking into account complex behaviors such as: non-linear behavior models, friction along the shaft of the needle and puncture force.


III. FORWARD SIMULATION

In this section we describe FE models and their numerical integration for the forward simulation.

A. Direct model of the Robot

The 6DOF Mitsubishi RV1A robot is a 6 arm anthropomorphic robot. The position of the end effector \mathbf{E} is given by

values \mathbf{q} of each joint, used in the kinematic model $\mathcal{R}(\mathbf{q})=\mathbf{E}$ given by Denavit-Hartenberg formulation in fig. 2.



	J1	J2	J3	J4	J5	J6
θ	0	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	0	0	0
a	300	0	0	160	0	72
d	0	250	-90	0	0	0
α	$-\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	0

Fig. 2. Denavit-Hartenberg model of the Mitsubishi RV1A robot.

For this application, we chose a joint space derivation to avoid large reconfiguration of the robot, but the method could also be derived for Cartesian control.

B. Finite Element Models

FE formulation of both the needle and the gel is based on the corotational FE formulation [9] which allows large displacements but is restricted to small stresses. The volume of the gel is discretized with linear tetrahedral elements, whereas the needle is composed of a set of linked beams, where each node has 6 dof. Both models are parametrized with E being the young modulus and ν the Poisson ratio. In both cases, the local stiffness matrix \mathbf{K}_e can be written with the synthetic formulation (see [9] and [7] for details):

$$\mathbf{K}_e = \mathbf{R}_e \int_{V_e} (\mathbf{C}_e \mathbf{D}_e \mathbf{C}_e \partial V_e) \mathbf{R}_e^T \quad (1)$$

where \mathbf{C}_e is the strain-displacement matrix, \mathbf{D}_e is the stress-strain matrix and \mathbf{R}_e is a rotation matrix derived in the corotational formulation. \mathbf{C}_e and \mathbf{D}_e are constant during the simulation, whereas \mathbf{R}_e needs to be updated at each step.

C. Needle-Tissue-Robot interactions

Lagrange multipliers λ are used to impose constraint forces on FE models in order to cancel the violation of constraints δ . Three type of constraints are considered:

Bilateral constraint χ is used to fix the position of the needle's base to the terminal part of the robot. χ is a holonomic 6DOF constraint (position and orientation) whose violation is defined as the relative displacement between the needle's base and \mathbf{B} an arbitrary position located on the terminal part of the robot. The transformation between the end effector \mathbf{E} and \mathbf{B} is assumed to be known. Since the robot is infinitely rigid, only the needle is affected by χ .

Penetration constraint ϕ is applied before penetrating the tissue, between the needle's tip and its closest surface on the gel's model. ϕ is a 3 dof constraint: a unilateral contact force is applied along the normal of the triangular surface to avoid the penetration, whereas coulomb friction is added in the tangential direction (see [6] for details). This constraint satisfies the Signorini conditions $\lambda \perp \delta$, i.e. if objects are distant ($\delta > 0$) any contact force vanishes ($\lambda = 0$), otherwise a positive contact force ($\lambda > 0$) is applied to cancel the penetration ($\delta = 0$). ϕ allows two parameters: p_f the puncture force threshold and μ_s the friction coefficient of the surface. Objects are considered to be in contact (leading

to a deformation) since $\lambda < p_f$, else ϕ is changed into a sliding constraint (see fig 3).

Sliding constraints ψ enforces the shaft of the needle to follow the path created by advancing needle tip. Each instance of ψ is a 3 dof constraint: a first constraint force applies a resistance to penetration along the needle's shaft, whereas two additional components prevent displacements in the tangential plane. ψ are dynamically added during the simulation as the needle is being inserted. Each constraint is defined by its barycentric coordinates on the tetrahedral mesh, which allows the definition of ψ at an arbitrary location in the volume (without any expensive remeshing). To solve FE problems, the constraint forces are then transferred to an equivalent nodal formulation using the principle of virtual work (see [7]). ψ are parametrized with μ_n being the penetration resistance coefficient along the shaft and d_n the minimum distance between constraints.

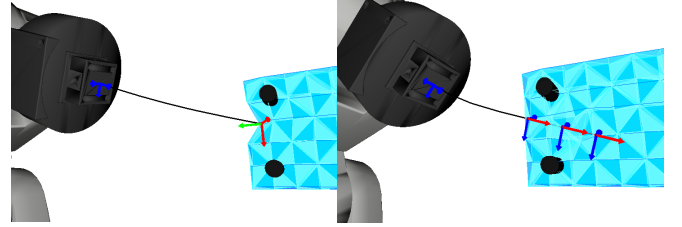


Fig. 3. Constraints applied during needle insertion simulation. Blue arrows are bilateral constraints. Red arrows are friction constraints. Green arrow is an unilateral constraint. Fixed parts of the gel (screws) are shown in black.

In the following section, the set of constraints χ , ϕ and ψ is expressed with the non linear function:

$$\mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathbf{q}) = \delta \quad (2)$$

where \mathbf{p}_n and \mathbf{p}_v are respectively the positions of the needle and gel. Also, Dirichlet conditions are added to impose null displacement of the gel at the screws location.

D. Time Integration and Constraint-based simulation

The governing equation is given by the static formulation:

$$\mathcal{F}_n(\mathbf{p}_n) + \mathcal{F}_v(\mathbf{p}_v) + \mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathbf{q}) \lambda = 0 \quad (3)$$

where \mathcal{F}_n and \mathcal{F}_v are non linear functions providing internal forces of needle and gel. Instead of directly solving this non-linear problem, each simulation step i consists in solving a linearized problem¹ :

$$\begin{aligned} \mathbf{K}_n^i \Delta \mathbf{p}_n^i + \mathbf{K}_v^i \Delta \mathbf{p}_v^i + (\mathbf{H}_n^i + \mathbf{H}_v^i)^T \lambda^i &= -\mathcal{F}_n(\mathbf{p}_n^i) - \mathcal{F}_v(\mathbf{p}_v^i) \\ \mathbf{K}^i &= \left. \frac{\partial \mathcal{F}}{\partial \mathbf{p}} \right|_{\mathbf{p}^i} = \sum (\mathbf{G}_e \mathbf{K}_e^i \mathbf{G}_e^T) \\ \mathbf{H}^i &= \left. \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \right|_{\mathbf{p}^i} \end{aligned} \quad (4)$$

where $\Delta \mathbf{p}^i$ and λ^i are unknown, \mathbf{H}^i are the *Jacobians of Contacts* and \mathbf{G}_e are *globalization matrices* transferring local

¹Note that solution of equation (4) is equivalent to perform the first iteration of the Newton-Raphson algorithm. Therefore, after several simulation steps, the positions at equilibrium (i.e. when $\Delta \mathbf{p}_n = \Delta \mathbf{p}_v = 0$) are the actual solution of the non linear problem formulated in the equation (3).

stiffness \mathbf{K}_e^i to global stiffness matrices \mathbf{K}^i . Since \mathbf{R}_e^i is not constant, \mathbf{K}_n^i and \mathbf{K}_v^i must be recomputed at each simulation step. For the sake of simplicity superscript i is now omitted.

Rewriting equation (4) and deriving (2), provides the following Karush-Kuhn-Tucker (KKT) system:

$$\begin{cases} \mathbf{K}_n \mathbf{x}_n + \mathbf{H}_n^T \lambda = \mathbf{b}_n & (5) \\ \mathbf{K}_v \mathbf{x}_v + \mathbf{H}_v^T \lambda = \mathbf{b}_v & (6) \\ \mathbf{H}_n \mathbf{x}_n + \mathbf{H}_v \mathbf{x}_v = \delta & (7) \end{cases}$$

where $\mathbf{x}_n = \Delta \mathbf{p}_n$ and $\mathbf{x}_v = \Delta \mathbf{p}_v$ and $\mathbf{b}_n = -\mathcal{F}_n(\mathbf{p}_n)$ and $\mathbf{b}_v = -\mathcal{F}_v(\mathbf{p}_v)$. The KKT problem is solved in five steps with the Schür complement method:

1- Free motion: We first compute the terms that do not depends on λ (i.e. $\mathbf{x}_n^{\text{free}} = \mathbf{K}_n^{-1} \mathbf{b}_n$) using a direct solver and $\mathbf{x}_v^{\text{free}} = \mathbf{K}_v^{-1} \mathbf{b}_v$ using a preconditioned Conjugate Gradient (see [3]). Dirichlet conditions enforce \mathbf{K}_n and \mathbf{K}_v to be invertible. Replacing \mathbf{x}^{free} in (5) and (6) gives:

$$\begin{cases} \mathbf{x}_n = \mathbf{x}_n^{\text{free}} - \mathbf{K}_n^{-1} \mathbf{H}_n^T \lambda & (8) \\ \mathbf{x}_v = \mathbf{x}_v^{\text{free}} - \mathbf{K}_v^{-1} \mathbf{H}_v^T \lambda & (9) \end{cases}$$

2- Collision detection: A proximity-based detection is performed allowing the definition of \mathbf{H}_n and \mathbf{H}_v (see [6] for details). These matrices represent the directions of constraints χ , ϕ and ψ as described in section III-C. In order to simplify the solution process, \mathbf{H}_n and \mathbf{H}_v are assumed to be constant during each simulation step. On the other hand, since $\mathbf{x}_n^{\text{free}}$ and $\mathbf{x}_v^{\text{free}}$ would be the positions of the needle and the gel if any constraint force λ was applied, the violation of constraint δ is defined based on \mathbf{H}_n , \mathbf{H}_v , $\mathbf{x}_n^{\text{free}}$ and $\mathbf{x}_v^{\text{free}}$.

3- Compliance computation: Replacing (8) and (9) in (7) requires the computation of the *compliance matrix*:

$$\mathbf{W} = \mathbf{H}_n \mathbf{K}_n^{-1} \mathbf{H}_n^T + \mathbf{H}_v \mathbf{K}_v^{-1} \mathbf{H}_v^T \quad (10)$$

Although \mathbf{W} is a small matrix (whose dimension is the square of the number of the constraints), its computation is the most time consuming step in the simulation since it involves the inversion of the large matrix \mathbf{K}_v . In order to reach real-time computations, \mathbf{W} is computed based on an asynchronous approximation and GPU parallelization as described in [3].

4- Constraints solving: Replacing (8), (9) and (10) in (7) gives the NLCP (Non-linear complementarity problem):

$$\mathbf{W} \lambda = \delta - \mathbf{H}_n \mathbf{x}_n^{\text{free}} - \mathbf{H}_v \mathbf{x}_v^{\text{free}} \quad (11)$$

where both λ and δ are unknown. The problem is solved using a modified Gauss-Seidel algorithm (see [6] for details).

5- Corrective motion: Once λ is known it is replaced in equations (5) and (6). The final position \mathbf{p}_n and \mathbf{p}_v that fulfill the contact and friction's laws can then be obtained inverting again \mathbf{K}_n and \mathbf{K}_v .

IV. INVERSE SIMULATION

Motions of needle and gel are entirely driven by the displacements of the robot. The problem is then reduced to find \mathbf{q} minimizing the objective function \mathcal{T} :

$$\min_{\mathbf{q}} (\mathcal{T}(\mathbf{q}, \mathbf{p}_n, \mathbf{p}_v)) \quad (12)$$

A. Objective function definition

The first component of the objective function \mathcal{T} aims at minimizing the distance between the needle's tip \mathbf{n} and a desired point on the trajectory \mathbf{t} . The trajectory is defined as a set of connected points whose positions are given by their barycentric coordinates of the respective tetrahedral mesh passing by. The desired position \mathbf{t} is given by the user controlling a parameter $c \in [0..1]$ which allows moving from the first to the last point of the trajectory.

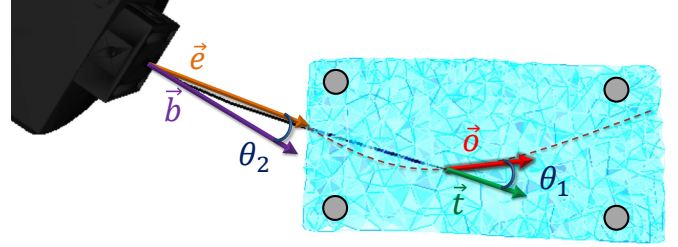


Fig. 4. Angular objective function: $\theta_1 = \arccos(\text{dot}(\vec{o}, \vec{t}))$ is the angle between needle's tip direction \vec{t} and the tangent of the target trajectory \vec{o} . $\theta_2 = \arccos(\text{dot}(\vec{e}, \vec{b}))$ is the angle between needle's base orientation \vec{b} and direction to the entry point \vec{e} .

In order to anticipate the displacement of the tip inside the volume, a minimization criteria is added to cancel θ_1 (i.e. the angle between the needle's tip and the tangent to the target trajectory). In addition, in order to avoid large deformations of the needle outside the gel, a second criteria aims at minimizing θ_2 , which is the angle between the needle base and the entry point. Orientation criteria are mutually exclusive. Outside the gel, the needle does not undergo any deformation ($\theta_2 = 0$) and only the minimization of θ_1 is considered. On the contrary, when the needle is sufficiently inserted, the mechanical coupling between the two structures prevents any control on the orientation of the needle tip. In this case: perpendicular motions of the base are applied to globally deform the gel in order to align the target trajectory with the needle shaft. Indeed, due to mechanical coupling with the gel, the inserted part of the needle becomes stiffer than the rest outside of the tissue. Any manipulation of the needle base in the lateral direction, may therefore cause large deformations of the needle's shaft. The minimization of θ_2 imposes that a roughly straight line is maintained between the needle base and the entry point.

The minimization criteria are summed in the same expression and \mathcal{T} is a vector composed of four scalars defined by:

$$\mathcal{T}(\mathbf{q}, \mathbf{p}_n, \mathbf{p}_v) = \begin{pmatrix} e_x \\ e_y \\ e_z \\ k \cdot \theta_1 + (1 - k) \cdot \theta_2 \end{pmatrix} \quad (13)$$

where $\mathbf{e} = \mathbf{t} - \mathbf{n}$ is the error between the needle tip and the desired point on the trajectory. Given a distance parameter d : the factor $k = 0$ when the needle is outside the gel, $k = \frac{l}{d}$ if the distance between the tip and the entry point is smaller than d , otherwise $k = 1$.

B. Inverse kinematics based on FE simulation

For each simulation step, a single iteration of the Newton-Raphson method is used to compute $\Delta \mathbf{q}$ which is the variation of the robot joints \mathbf{q} minimizing equation (12). The newton method requires the computation of the Jacobian of the needle's tip motion \mathbf{J} . Despite the analytical derivation of $\mathbf{J} = \frac{\partial \mathcal{T}}{\partial \mathbf{q}}$ is too complex because it also requires the derivation of the forward simulation of section III, a numerical approximation can be computed defining $\delta \mathbf{q}_i$ a small perturbation of articulation i :

$$\mathbf{J}_i = \frac{\mathcal{T}(\mathbf{q}, \mathbf{p}_n, \mathbf{p}_v) - \mathcal{T}(\mathbf{q} + \delta \mathbf{q}_i, \mathbf{p}_n, \mathbf{p}_v)}{\|\delta \mathbf{q}_i\|} \quad (14)$$

where \mathbf{J}_i is the column i of the Jacobian and $\mathcal{T}(\mathbf{q}, \mathbf{p}_n, \mathbf{p}_v)$ is the value of the objective function at the beginning of the simulation step. Solving 6 independent simulations the entire Jacobian (whose dimension is 4×6) can be entirely computed. $\Delta \mathbf{q} = \mathbf{J}^{-1} \times \mathcal{T}(\mathbf{q}, \mathbf{p}_n, \mathbf{p}_v)$ can finally be obtained by computing the pseudo inverse of \mathbf{J} .

Algorithm 1: Inverse Simulation Loop

```

1 Free Motion:  $\mathbf{x}^{\text{free}} = \mathbf{K}^{-1} \mathbf{b}$ 
2 Store:  $\bar{\mathbf{p}} = \mathbf{p}$ 
3 Collision Detection:  $\mathbf{H} = \text{proximity}(\bar{\mathbf{q}}, \bar{\mathbf{p}})$ 
4 Compute Compliance:  $\mathbf{W} = \sum \mathbf{H} \mathbf{K}^{-1} \mathbf{H}^T$ 
5 Compute error:  $\mathbf{e} = \mathcal{T}(\bar{\mathbf{q}}, \bar{\mathbf{p}})$ 
6 if  $\text{dot}(\mathbf{e}, \mathbf{e}) > \varepsilon$  then
7   for  $i = 0$  to  $\delta$  do
8     Move Robot:  $\mathbf{E}_i = \mathcal{R}(\mathbf{q} + \delta \mathbf{q}_i)$ 
9     Compute Violation:  $\delta = \mathcal{H}(\mathbf{p}, \mathbf{q} + \delta \mathbf{q}_i)$ 
10    Solve Constraints:  $\mathbf{W} \lambda = \delta$ 
11    Corrective Motion:  $\mathbf{p} = \mathbf{x}^{\text{free}} - \mathbf{K}^{-1} \mathbf{H}^T \lambda$ 
12    Compute Jacobian:  $\mathbf{J}_i = \frac{\mathcal{T}(\bar{\mathbf{q}}, \bar{\mathbf{p}}) - \mathcal{T}(\mathbf{q} + \delta \mathbf{q}_i, \mathbf{p})}{\|\delta \mathbf{q}_i\|}$ 
13    Reload:  $\mathbf{p} = \bar{\mathbf{p}}$ 
14     $\Delta \mathbf{q} = \mathbf{J}^{-1} \cdot \mathbf{e}$ 
15    Move Robot:  $\mathbf{E} = \mathcal{R}(\mathbf{q} + \Delta \mathbf{q})$ 
16 Compute Violation:  $\delta = \mathcal{H}(\mathbf{p}, \mathbf{q} + \Delta \mathbf{q})$ 
17 Solve Constraints:  $\mathbf{W} \lambda = \delta$ 
18 Corrective Motion:  $\mathbf{p} = \mathbf{x}^{\text{free}} - \mathbf{K}^{-1} \mathbf{H}^T \lambda$ 

```

The modified simulation loop is shown in algorithm 1. Despite the fact that 6 additional simulations must be solved to compute the numerical Jacobian, real-time computations are still possible since \mathbf{W} (being the most expensive task) is computed only once per simulation step. This is possible because \mathbf{H} is assumed to be constant during the simulation step, which is a reasonable approximation while the displacements of the robot remains small. Finally, only the violation of constraints δ must be recomputed requiring the solution of a new constraint problem to obtain λ . However, λ have mechanical coherency and GS solver can be significantly improved using an initial guess λ_{i-1} of the previous step.

V. RESULTS

In this section we evaluate our method in terms of accuracy, computational time and parameters' sensitivity. The use of our method with a real robotic system would require the biomechanical characterization of deformable structures. However, the identification of such parameters is not in the scope of this paper. Numerical validations are then proposed

to verify that the displacements of the robot fulfill the minimization objective of section IV.

A. Evaluation of the accuracy

To evaluate the accuracy of our method, we compare two different strategies where the robot is controlled using: i) our inverse simulation ii) a rigid model of the needle and the gel. Both strategies are used uniquely to control the robot. The joint displacements obtained are applied in forward simulations where we consider a deformable needle and gel.

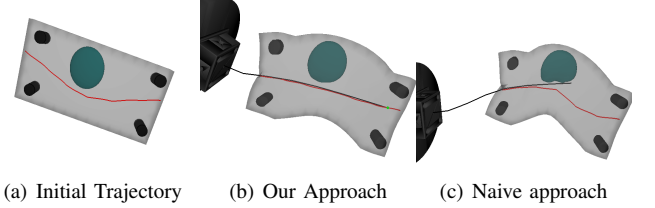


Fig. 5. Simple trajectory (red) with 1 obstacle (green).

A simple trajectory is shown in fig. 5. Although the naive approach could lead to a successful insertion over the first centimeters (when the trajectory stays straight), significant errors are progressively introduced due to a miss-evaluation of the deflection of the needle in the control loop (fig. 7 left).

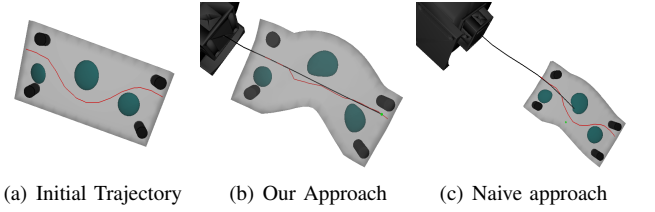


Fig. 6. Complex trajectory (red) with 3 obstacles (green).

A more complex scenario is shown in fig. 6, where it is required to overcome several obstacles throughout the insertion. In this case, the naive approach leads straight inside an obstacle, whereas our method can successfully perform the whole insertion by avoiding every obstacle. As shown in the images, the global shape of the gel results in being deformed. Such deformation is induced by the needle itself, which being more rigid tends to stay straight, resulting in being aligned with the trajectory when inserted.

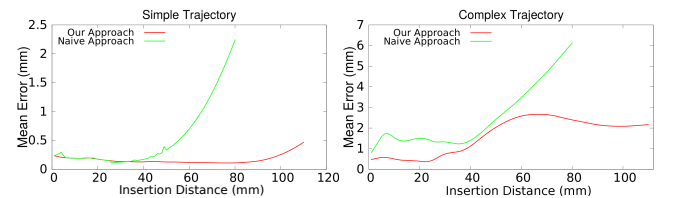


Fig. 7. The mean error based on the insertion distance for needle insertion scenarios of figure 5 and 6. Parameters: $\delta \mathbf{q}_i = 0.01$ radian, mesh of 1500 nodes, $d_n = 10$ mm and $d = 20$ mm.

Error $e_{\text{mean}} = 0.5$ mm and $e_{\text{max}} = 1.8$ mm are reported when the needle is entirely inserted with the simple scenario,

and $e_{mean} = 2.6 \text{ mm}$ and $e_{max} = 6.0 \text{ mm}$ for the complex one (see fig. 7). The whole insertion couldn't be processed with the naive approach since significant deformations of the FE models raised numerical instabilities.

B. Sensitivity analysis of parameters

Independently from biomechanical parameters, the method relies on 4 numerical parameters for whom we performed a sensitivity analysis. δq_i does not significantly affect the accuracy of the insertion since variations of less than 0.1 mm have been reported for a range between 0.001 and 0.1 radians. Higher values of d avoid large bending of the part of the needle outside the gel, but it does not affect the accuracy of the insertion either. Indeed varying d in a range going from 10% to 50% of the length of the gel, causes variations of less than 1 mm. However, d remains an interesting parameter to prevent potential needle rupture or the tissue's tearing. Discretization of the FE mesh and distance between constraints have been evaluated. Although the error remains lower than 1 mm for the simple trajectory, the fine mesh allows an accurate needle insertion in complex scenarios. The mean error of 3.2 mm is reported for a mesh of 600 nodes, but it decreases to 1 mm for 1800 nodes. The distance between constraints d_n has a higher influence on the the accuracy, as the gel and the needle can freely move between constraints. Decreasing the distance d_n improves the accuracy, but over-constrained problems may occur if d_n is lower than the element size of the FE mesh.

C. Computation Time

The computational time and the percentage of main simulation steps are reported in table 8. The free motion (FM) and the compliance computation (CC) are processed only once per simulation step. The constraint solver (SC) is the most expensive task, but being processed 7 times per step each articulation requires around 7%. Finally a real-time control of the robot is possible since for meshes of 1800 node we can achieve 40 FPS.

		Simulation Steps (%)				Time (ms)
		FM	CD	CC	SC	
# Nodes	600	19.19	2.01	27.52	39,21	8.85
	1100	25.12	0.97	16.20	49,71	13.24
	1500	28.36	1.23	19.96	43,45	12.92
	1800	25.93	0.58	17.63	52,03	24.43

Fig. 8. Computation time with different mesh size. FM: Free Motion, CD: Collision Detection, CC: Compute Compliance, SC: Solve constraints.

VI. CONCLUSION

An inverse real-time simulation has been proposed allowing the control of an articulated robot for needle steering in deformable tissues. The method provide in real-time displacements of the robot allowing the generation of complex, non straight trajectories to avoid obstacles. Future work aims at using the method with a real robotic system. Since identification of the biomechanical parameters would remain an open problem, we consider using real-time registration of the FE models with visual tracking of deformable structures.

REFERENCES

- [1] Niki Abolhassani, Rajni Patel, and Mehrdad Moallem. Needle insertion into soft tissue: A survey. *Medical Engineering and Physics*, 29(4):413–431, 2007.
- [2] Nuttapon Chentanez, Ron Alterovitz, Daniel Ritchie, Lita Cho, Kris K Hauser, Ken Goldberg, Jonathan Richard Shewchuk, and James F O'Brien. Interactive simulation of surgical needle insertion and steering. In *ACM Transactions on Graphics*, volume 28, 2009.
- [3] Hadrien Courtecuisse, Jérémie Allard, Christian Duriez, and Stéphane Cotin. Preconditioner-based contact response and application to cataract surgery. *MICCAI*, 6891 LNCS(PART 1):315–322, 2011.
- [4] Noah J Cowan, Ken Goldberg, Gregory S Chirikjian, Gabor Fichtinger, Ron Alterovitz, Kyle B Reed, Vinutha Kallem, Wooram Park, Sarthak Misra, and Allison M Okamura. Robotic needle steering: Design, modeling, planning, and image guidance. *Surgical Robotics*, 2011.
- [5] Simon P DiMaio and Septimiu E Salcudean. Needle Steering and Motion Planning in Soft Tissues. *IEEE Transactions on Biomedical Engineering*, 52(6):965–974, 2005.
- [6] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic Haptic Rendering of Interactive Deformable Objects in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):36–47, 2006.
- [7] Christian Duriez, Christophe Guébert, Maud Marchal, Stéphane Cotin, and Laurent Grisoni. Interactive simulation of flexible needle insertions based on constraint models. In *MICCAI*, 2009.
- [8] Inko Elgezua, Yo Kobayashi, and Masakatsu G. Fujie. Survey on current state-of-the-art in needle insertion robots: Open challenges for application in real surgery. *Procedia CIRP*, 5:94–99, 2013.
- [9] C A Felippa and B Haugen. Unified Formulation of Small-Strain Corotational Finite Elements : I . Theory Unified Formulation of Small-Strain Corotational Finite Elements : I . Theory, 2005.
- [10] Daniel Glzman and Moshe Shoham. Image-guided robotic flexible needle steering. *IEEE Transactions on Robotics*, 23(3):459–467, 2007.
- [11] Laurent Goffin, Gaetan Bour, Fernand Martel, Stéphane Nicolau, Jacques Gangloff, Jean Marc Egly, and Bernard Bayle. Design and in vivo evaluation of a robotized needle insertion system for small animals. *IEEE Transactions on Biomedical Engineering*, 60(8), 2013.
- [12] Noura Hamzé, Igor Peterlík, Stéphane Cotin, and Caroline Essert. Preoperative trajectory planning for percutaneous procedures in deformable environments. *Computerized Medical Imaging and Graphics*.
- [13] Deborah R Kaye, Dan Stoianovici, and Misop Han. Robotic ultrasound and needle guidance for prostate cancer management: review of the contemporary literature. *Current opinion in urology*, 24(1), 2014.
- [14] Mohsen Khadem, Bitia Fallahi, Rossa Carlos, Ron Sloboda, Nawaid Usmani, and Mahdi Tavakoli. A Mechanics-based Model for Simulation and Control of Flexible Needle Insertion in Soft Tissue. *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [15] Y. Kobayashi, A. Onishi, H. Watanabe, T. Hoshi, K. Kawamura, M. Hashizume, and M. G Fujie. Development of an integrated needle insertion system with image guidance and deformation simulation. *Computerized Medical Imaging and Graphics*, 34(1):9–18, 2010.
- [16] Alexandre Krupa. A new duty-cycling approach for 3D needle steering allowing the use of the classical visual servoing framework for targeting tasks. *Biomedical Robotics and Biomechanics (2014)*.
- [17] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequid, and C. Duriez. Real-time control of soft-robots using asynchronous finite element modeling. *2015 IEEE ICRA*, 2015.
- [18] Vishnu G Mallapragada, Nilanjan Sarkar, and Tarun K Podder. Robot-assisted real-time tumor manipulation for breast biopsy. *IEEE Transactions on Robotics*, 25(2):316–324, 2009.
- [19] N. Navab, B. Bascle, M. Loser, B. Geiger, and R. Taylor. Visual servoing for automatic and uncalibrated needle placement for percutaneous procedures. *CVPR 2000*, 2:327–334, 2000.
- [20] Claudio Pacchierotti, Momen Abayazid, Sarthak Misra, and Domenico Prattichizzo. Steering of flexible needles combining kinesthetic and vibratory force feedback . *IROS*, 2014.
- [21] Olivier Piccin, Laurent Barbe, Bernard Bayle, Michel De Mathelin, and A. Gangi. A Force Feedback Teleoperated Needle Insertion Device for Percutaneous Procedures. *The International Journal of Robotics Research*, 28(9):1154–1168, 2009.
- [22] Septimiu E. Salcudean, Thomas D. Prananta, William J. Morris, and Ingrid Spadinger. A robotic needle guide for prostate brachytherapy. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2975–2981, 2008.