



**HAL**  
open science

# Robotic Insertion of Flexible Needle in Deformable Structures Using Inverse Finite-Element Simulation

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

► **To cite this version:**

Yinoussa Adagolodjo, Laurent Goffin, Michel de Mathelin, Hadrien Courtecuisse, Yinoussa Adagolodjo. Robotic Insertion of Flexible Needle in Deformable Structures Using Inverse Finite-Element Simulation. IEEE Transactions on Robotics, 2019, 35 (3), pp.697-708. 10.1109/TRO.2019.2897858 . hal-02144487

**HAL Id: hal-02144487**

**<https://inria.hal.science/hal-02144487v1>**

Submitted on 30 May 2019

**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.



Distributed under a Creative Commons Attribution 4.0 International License

# Robotic insertion of flexible needle in deformable structures using inverse Finite Element simulation

Yinoussa Adagolodjo, Laurent Goffin, Michel De Mathelin and Hadrien Courtecuisse

**Abstract**—This paper introduces a new approach for the control of a robotic system interacting with deformable structures. The method is applied to needle insertion procedures, which are among the least invasive surgical approaches to access deep internal structures with sometimes poor access conditions. Yet, during the insertion both tissues and needles deform resulting in a displacement of targets identified at the planning step and significantly raise the technical difficulty of these approaches. Robotic assistance may offer new possibilities to enforce the accuracy of the needle’s positioning, but the deformation of tissues remains an open problem. In this paper we propose a numerical approach where Finite Element (FE) models are used in a closed-loop robotic control. We introduce a complete forward simulation of deformable structures (needle, environment), and constraint-based interaction models allowing for the simulation of needle insertion and complex non-linear phenomena (friction, puncture, insertion) at high frequency. For the control, we numerically derive the so-called *Jacobian of the Simulation* using inverse method. The most original aspect of this work lies in the fact that inverse steps are performed in constraints space, allowing this way for fast estimation of the Jacobian (i.e. between 40-100 Hz). The method is validated both numerically and experimentally using a flexible needle inserted inside a deformable foam. We show that the robot is able to follow a given trajectory, defined during the planning step, taking into account any occurring deformation of both the needle and the foam during the insertion; without any need for tracking the needle nor the target nor the trajectory.

**Index Terms**—Robotic Needle insertion; Flexible needle; Needle steering; Modeling and FE Simulations.

## I. INTRODUCTION

NEEDLE-based interventions are among the least invasive surgical approaches to access deep internal structures into organs’ volumes without damaging surrounding tissues. Unlike traditional open surgery, needle-based approaches only affect a localized area around the needle, reducing this way the occurrence of traumas and risks of complications [1]. Many surgical procedures rely on needles in nowadays clinical routines (biopsies, local anesthesia, blood sampling, prostate brachytherapy, vertebroplasty ...). Radiofrequency ablation (RFA) is another percutaneous example that uses heat at the tip of a needle to destroy cancer cells. Such alternative treatments may open new solutions for unrespectable tumors or metastasis (concerns about the age of the patient, the extent or localization of the disease). However, contrary to what one may think, needle-based approaches can be an exceedingly complex intervention. Indeed, the effectiveness of the treatment is highly dependent on the accuracy of the needle positioning (generally required around few millimeters) which can be particularly challenging when needles are manipulated from outside the

patient with intra-operative images (X-ray, fluoroscopy or ultrasound ...) offering poor visibility of internal structures. Human factors, organs’ deformations, needle deflection and image limitations can be causes of needle misplacement and rise significantly the technical level necessary to master these surgical acts.

The use of surgical robots has revolutionized the way surgeons approach minimally invasive surgery. Robots have the potential to overcome several limitations coming from the human factor: for instance by filtering operator tremors, scaling the motion of the user or adding new degrees of freedom at the tip of instruments. A rapidly growing number of surgical robots has been developed and applied to a large panel of surgical applications [2]. Yet, an important difficulty for needle-based procedures lies in the fact that both soft tissues and needles tend to deform as the insertion proceeds in a way that cannot be described with geometrical approaches. Standard solutions address the problem of the deformation extracting a set of features from per-operative images (also called *visual servoing*) and locally adjust the pose/motion of the robot to compensate for deformations [3]. Nevertheless, *visual servoing* raises several limitations, in particular for the needle insertion:

- 1) Per-operative images usually offer poor visibility of internal structures (such as a tumor or vessels), and it is very challenging to extract essential data at high frequency. This is especially true for disappearing liver metastases: due to chemotherapy effects, the shape of tumors may change or they may become invisible in intraoperative images, even if the lesions still contain active tumors[4].
- 2) When large deformations occur the control law of the robot can be significantly modified which is extremely difficult to relate with image-based displacements. For instance, when the needle is deeply inserted inside the tissue, the needle shaft becomes completely constrained, preventing for any lateral motions of the needle.
- 3) Traditional controllers do not have access to any biomechanical models capable of predicting the deformation of organs at high frequency. Yet, the trajectory taken by the needle at the beginning of the insertion has a significant impact on the ability to reach or not the target later.

To overcome these limitations, we introduce a numerical method allowing performing inverse Finite Element simulations at high frequency. We show that it can be used to control an articulated robot while considering deformations of structures during needle insertion. Our approach relies

on a forward FE simulation of a needle insertion (involving complex non-linear phenomena such as friction, puncture and needle constraints). Control commands are then derived from two important steps:

**Corrective Step:** As for *visual servoing*, we extract a set of features from live images in order to enforce the consistency of the models with real-data. However, instead of directly steering the needle toward these features, we first register FE models with the observations. The advantage of relying on FE models lies in the fact that it provides a regularization technique to extrapolate the displacement field extracted from images. Moreover, it allows to interpolate the whole volume displacement of the organs (including internal structures such as tumors, vessels or non-visible tumors), even if only few landmarks are visible in the images.

**Predictive Step:** Input commands of the robot are obtained from an optimization process based on inverse simulation steps of FE models. This allows anticipating the behavior of mechanical structures, in order to adapt input commands much faster than waiting for a correction from the images. Inverse steps are performed to numerically derive the so-called *Jacobian of the Simulation*, which relates Cartesian displacements of the base of the needle with displacements of the tip inside the volume, allowing to compensate, or even induce, necessary deformations to reach a target.

In this paper, we introduce a closed-control robotic loop (see Fig. 1). The non-rigid registration process allows maintaining low deformation errors between FE models and real structures. An important difficulty concerns the computation time of inverse steps, especially because the overall system is highly non-linear and the validity domain of the Jacobian is limited to small deformations and remains valid a small amount of time. In order to meet computation time constraints necessary for the robotic control, a generic constraint-based formulation is proposed. In a quasi-static scenario, we show that a sufficient frame-rate can be achieved for a stable needle insertion in a deformable environment, allowing for a total insertion time compatible with clinical constraints.

## II. RELATED WORKS

Although robotic needle insertion has been a subject of considerable interest over the last years, recent surveys [5], [1], [6], [7] still highlights the need for control models when dealing with deformable tissues. In this section, we review the main methods that have been proposed for accurate robotic needle steering.

### A. Needle and target Tracking

Image-guided techniques extract information (for instance needle and target positions) from vision sensors. This information is used in a closed-control loop to guide the needle tip towards the target. In [8], Kobayashi et al. track the needle shape using ultrasound (US) images. In [9] authors

estimate flexible needle's tip position from live US images. The method is combined with tissue stiffness estimation (from localized tissue displacements), and applied to the robotic needle insertion inside soft tissues. Although US are fast, portable, widely available, and easy to combine with robotic systems, the needle visibility in ultrasound images remains an open problem. In [10], Okazawa et al. presented two methods to detect the needle in 2D ultrasound that specifically address needle curvature.

Image quality of US being limited, other image modalities has been investigated for needle tracking (see [5] for a survey). For instance, in [11] Seifabadi presents an MRI-compatible robot, for teleoperated bevel-tip needle steering under real-time MRI guidance. Navab et al. [12] use X-ray fluoroscopy to align a needle (held by a medical robot) inside a porcine kidney. Image quality of MRI is often much better, but the acquisition time is usually slower and it raises several difficulties to align the images' plane with the structures.

Alternatively to imaging systems that are often limited by image quality or acquisition time, other solutions have been proposed to reconstruct the needle trajectory using optical fibers. [13] used an optical fiber embedded into needle's shaft for a direct measurement of the deflection and even for a three-dimensional reconstruction of the needle shape. In [14], [15] Kim et al. use a set of Fiber Bragg Gratings (FBG) sensors and general elastic rod theory to reconstruct the shape of the needle in 3D even for large deflection of the needle.

Instead of steering the needle to the tumor, Mallapragada et al. [16] proposed a method to move the tumor toward the needle trajectory. The method is used for breast biopsy and takes as input real-time fluoroscopic images, in which the tumor is located.

A common limitation of the above methods lies in the fact that both the target and the needle must be visible in online images. Yet, images qualities being most of the time antagonistic with the acquisition frequency, it limits applications either to off-line insertions or raises significant difficulties on image-processing algorithms and images localization.

### B. Duty-cycling approach

Some approaches explore the possibility of changing the curvature of beveled-tip flexible needles. The method named *duty-cycling*, consists of spinning the needle along its insertion axis to add 3 additional Degrees of Freedom (DoFs) at the tip of the needle, hence allowing the steering of the needle inside the volume during the insertion. Bernardes et al. [17] developed a robot-assisted approach for the automatic steering of flexible beveled needles in percutaneous procedures. The method uses duty-cycled rotation of the needle to perform insertion with arcs of adjustable curvature. The method is used in a closed-loop imaging feedback with an intra-operative motion replanning strategies to compensate for system uncertainties and disturbances. In [18] Krupa et al. presented a duty-cycling robotized system for steering beveled needles allowing this way for the creation of complex non-straight trajectories to reach a target and avoid obstacles. [19] proposed a bio-inspired multi-part needle and validated the control strategy by fitting experimental models. Reed et al. describes

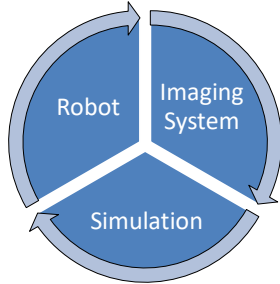


Fig. 1. Control loop: The robot is controlled by an inverse FE simulation; itself registered with intra-operative images.

in [20] a robot-assisted needle steering system using three integrated controllers. They obtain similar results by applying a stochastic motion planner with paths replanning for duty-cycling techniques.

All the aforementioned methods share two main limitations: (1) They all rely directly on images to track the needle and define the target point; either using tomographic reconstruction (CT, MRI), or live images (echography, fluoroscopy), entailing many challenges for image features extraction and noise correction. (2) Even if these approaches add new DoFs at the tip of the needle, the correction of the needle trajectory is always performed a posteriori (i.e. only when errors are detected in the images). None of these methods are capable to predict the behavior of the tissues in order to generate a priori deformations to reach the target.

### C. Simulation, modeling and trajectory planning

Modeling the interactions between needle and tissue is essential to predict the behavior of the needle inside tissues. Misra et al. [21] studied interaction force at the tip of asymmetry bevel-tip needles to estimate the rupture toughness of the tissues. In [22], Rucker et al. proposed a closed-loop control system for asymmetric-tipped needle. [13] proposed an image-guided control system to steer flexible needles. Both kinematics and mechanical models are used to predict the deflection of the needle model.

Large scale simulations of needle insertion (including advanced FE models of needle, tissue and interactions) have also received a considerable interest, mainly for training purpose. Chentanez et al. [23] presented a FE approach based on the beam's theory to predict needle deflection. However, the method is not compatible with real-time computations mainly due to expensive re-meshing steps along the needle shaft. Interactive models were proposed in [24]. The method does not require any re-meshing process when the needle goes through the tissue. In addition, it allows for the simulation of complex phenomena such as tissue deformations, needle-tissue friction, and puncture force.

Based on similar models, several researchers have developed motion planners for symmetric-tip flexible needles in 3D tissues [25], [26]. Duindam et al. derived an inverse kinematics solution to reach a desired position and orientation in 3D [27]. However, all these methods assumed that the tissue is rigid. Alterovitz et al. [28] presented trajectory planning algorithms including probabilistic methods considering uncertainty. The method was improved in [29] to explicitly consider motion and uncertainties while guiding the needle to a target in 3D anatomy. [30] relied on the method proposed in [24] for optimal trajectory planning for liver surgery. The method takes into account deformations (breathing, needle deflection, friction) to avoid obstacles identified at the planning step.

The main limitation of off-line trajectory planning strategies lies in the fact that some significant changes might occur between pre-operative and intra-operative configurations, and invalidate the chosen trajectory in a real application. Li et al. [31] proposed a path planning (and online replanning) approach for steerable needles based on discrete potential field in 3D anatomical structures but the method is not real-time.

### D. Kinematic and mechanical model-based control

DiMaio and Salcudean [32] were among the pioneers to investigate robotic needles' steering through soft tissue. They compute numerically the *Jacobian matrix* from mechanical models, from which the needle base velocity is derived and used as input in an open robotic control loop. The method is combined with trajectory planning strategies: attractive fields drive the needle towards the desired target whereas repulsive fields avoid obstacles, but the method is not real-time. Glozman and Salcudean [33] proposed a real-time steering system that integrates planning and control in a closed-control loop for dynamic systems. A mechanical model (springs with different stiffness coefficients along the needle shaft) was used to simulate the interaction between the needle and soft tissues.

In order to predict the behavior of structures, more advanced mechanical models have been used to predict the behavior of the tissue. In [34] barbe et al. uses a linear Kelvin-Voigt (KV) model to estimate online the forces involved in percutaneous interventions, but the method is limited to the forces applied in the direction of the needle. Khadem et al. [35] presented a mechanics-based model for the simulation of a needle insertion in soft tissues. Robotic tests are conducted to identify the parameters of the model, then used to steer the real needle. In [36] Kobayashi uses a nonlinear viscoelastic model calculated intra-operatively to manipulate a needle while considering organ deformations. The force upon the needle is measured using a force sensor and applied to a mechanical model allowing for the estimation of deformations of the tissue. Nevertheless, no interaction models between the needle and the tissue is proposed restricting the method to superficial insertions. The method is then extended in [8] using an ultrasound-guided 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, but this method is limited to 2D insertions.

To the best of our knowledge, no robotic solution allows for generating large deformation and predicting the behavior of structures in order to guide a robot for automatic needle insertion. One of the main reasons lies in the fact that once the needle is inserted, the overall system (robot, needle and tissue) can be seen as a deformable robot with an infinite number of degrees of freedom, which cannot be controlled with standard robotic approaches. The control of soft robot is a recent research topic. Largilliere [37] proposed a more advanced control strategy, based on FE models to deform a soft robot. An inverse problem based on a QP (quadratic-programming) algorithm is used to solve the equations of motion and control a soft-robot in real-time. However, interactions between deformable structures with complex behaviors, such as nonlinear friction, are not taken into account.

### E. Non-rigid registration of FE models

The use of bio-mechanical models in the operating room is a dynamic research topic. Recent works aimed at introducing biomechanical models for augmented reality (AR). As opposed to image-to-image registration, biomechanical models provide a physics-based extrapolation, not just geometric, in areas where few or no intraoperative data are available (see [38] for



a detailed discussion). Marami et al. [39] presents a method for non-rigid 3D to 3D and 3D to 2D registration of medical images. The registration technique employs a dynamic linear elastic model of the tissue to dynamically track a target inside breast tissue during a biopsy. In [40], a biomechanical model of the heart is controlled by active surface constraints created from features extracted from a stereo-vision camera. The method allows for accurate estimation of the position of internal structures, but is limited to cyclic movements.

Suwelack et al. [41] proposed an electrostatic-elastic problem formulation for the registration of a physically based model derived from pre-operative mesh and intra-operative surface computed from a stereo laparoscopic camera. However, the method requires that at least 50% of the organ surface is captured by the camera, which is generally not possible. [42] exploits salient anatomical features, identifiable in both pre-operative and intra-operative images of the liver. The method was used to display with AR internal structures of the pre-operative scan on top of the intra-operative view obtained from a laparoscopic camera. 2D dynamic MRI has been used in [43] for the registration of a pig's liver during breathing motion. The method provides the complete 3D motion of the organ from a single 2D dynamic MRI slice and a pre-operative scan. Finally, [44] developed an approach to compensate for craniotomy-induced brain-shift. A biomechanical simulation, relying on a non-linear constitutive law, is constrained to register vessels extracted from preoperative MR and intraoperative Doppler Ultrasound.

#### F. Contribution and positioning

This paper is an extension of the method introduced in [45] which only provides validation in a simulated environment. We show that our control strategy, based on inverse FE simulations, allows the control of a real robotic system in order to automatically steer a flexible needle in a deformable structure. The main motivation of this work comes from the fact that FE simulators are now considered as a clinically relevant tool for both training and assistance during surgery with AR. We show that a sufficient knowledge can be extracted from FE models in order to guide the needle and reach a target in a deformable environment.

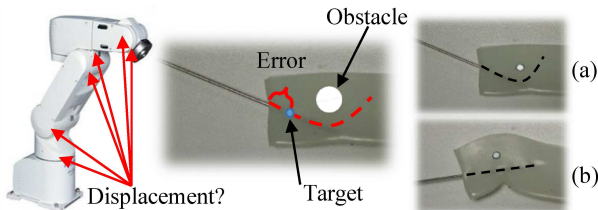


Fig. 2. Needle insertion inside a deformable environment. (a) A curved trajectory is defined (dashed-line) to avoid an obstacle (gray circle). (b) Since the needle is stiffer than the foam, it is necessary to deform the foam (with tangential motion of the needles base) in order to deform the trajectory to a straight line aligned with the needles shaft.

**Problem Statement:** Our goal is to insert a flexible needle in a deformable environment (see fig 2). A polyurethane foam model was chosen for its low-friction property during the needle insertion (comparable to an organ), but the method is independent of the inserted media. A trajectory is manually

defined based on a tomographic reconstruction (CT) of the foam, in an undeformed configuration (see fig 2). A flexible needle is attached to the end-effector of an articulated robot considered as being infinitely rigid. The foam is attached to a support, positioned within the working space of the robot, and it is considered to be deformable. The goal of this work is to provide Cartesian displacements of the base of the robot such that the tip of the needle remains on the predefined trajectory for any occurring deformation. It is important to note that an inverse problem must be solved to perform this insertion. Indeed, although the trajectory is completely defined at the initial step, it will deform as the insertion proceeds. Once the needle is inserted inside the volume, any displacement of the base of the robot will modify the desired trajectory, and the input displacement command must be adapted accordingly.

We underline that our goal is neither to provide an optimal trajectory nor to enforce the feasibility of the trajectory. Instead we assume the trajectory being generated by a planning system taking into account mechanical effects such as [30], or given by an expert (surgeon). If the input trajectory is not reachable, we expect the robotic system to stop falling in a local minimum. In addition, the non-rigid registration of FE models with live images remains an open research problem. In order to focus on robotic aspects, in this paper we voluntarily simplified this step relying on optical markers placed on the surface of the foam. In the rest of the paper we assume that a sparse set of 3D observation points  $\mathbf{m}$  located of foam's surface are visible from an external tracking system. This solution has recently been used for AR of open liver surgery [46]. The accuracy of the registration step is not detailed in this paper (see [46], [44] and [38] for more information).

The rest of this paper is organized as follows. In section III we describe the forward biomechanical models of the needle, the foam and the interactions as well as the integration and solving process. Section IV is dedicated to the inverse steps, and the computation of input commands of the robot. In section V we evaluate the method both numerically and experimentally.

### III. FINITE ELEMENT MODELS

In this section we introduce FE formulations and interaction models and we will describe the governing equations of the simulation.

#### A. Finite Element (FE) Models

The model of the needle is based on the Timoshenko formulation [47], which relies on beam's theory. The needle is described as a set of linked beams, each beam being composed of two nodes and each node having 6 DoFs (position and rotation). The local stiffness matrix  $\mathbf{K}_{ne}$  is assembled as follows:

$$\mathbf{K}_{ne} = \int_{V_{ne}} (\mathbf{C}_{ne}^T \mathbf{D}_{ne} \mathbf{C}_{ne} dV_{ne}) \quad (1)$$

where  $\mathbf{C}_{ne}$  is the strain-displacement matrix and  $\mathbf{D}_{ne}$  is the stress-strain matrix.  $\mathbf{K}_{ne}$  is a  $12 \times 12$  matrix.

The geometry of the mesh is obtained from a segmentation of the foam performed before the insertion. The volume is meshed with linear tetrahedral elements, where each element

has four nodes with 3 DoFs. The local stiffness matrix of each element (tetrahedral) can be written :

$$\mathbf{K}_{ve} = \int_{V_{ve}} (\mathbf{C}_{ve}^T \mathbf{D}_{ve} \mathbf{C}_{ve} dV_{ve}) \quad (2)$$

where  $\mathbf{C}_{ve}$  and  $\mathbf{D}_{ve}$  are respectively the strain-displacement matrix and the stress-strain matrix.  $\mathbf{K}_{ve}$  is  $12 \times 12$  matrix, see [48] for further details about the computation of each component.

The computation of the global stiffness matrix of both tissue and needle are based on the co-rotational formulation [49] which allows for large displacements (but is restricted to small stresses). Let's note:

$$\mathbf{R}_{ve} = \begin{bmatrix} \mathbf{R}_{ve*} & & & \\ & \mathbf{R}_{ve*} & & \\ & & \mathbf{R}_{ve*} & \\ & & & \mathbf{R}_{ve*} \end{bmatrix}; \mathbf{R}_{ne} = \begin{bmatrix} \mathbf{R}_{ne1} & & & \\ & \mathbf{R}_{ne1} & & \\ & & \mathbf{R}_{ne2} & \\ & & & \mathbf{R}_{ne2} \end{bmatrix} \quad (3)$$

$\mathbf{R}_{ve}$  is a block-diagonal rotation matrix of a tetrahedral element  $e$ . The  $3 \times 3$  rotation matrix  $\mathbf{R}_{ve*}$  is obtained from a polar decomposition of the nodal positions of the tetrahedron as proposed in [48].  $\mathbf{R}_{ne}$  is a rotation matrix of a beam element  $e$  and  $\mathbf{R}_{ne1}$  and  $\mathbf{R}_{ne2}$  are the nodal rotations of beam's nodes.

Based on the above definitions the co-rotational elastic forces of an element (tetrahedral or beam) can be written with the synthetic formulation :

$$\mathbf{f}_e = \mathbf{R}_e \mathbf{K}_e (\mathbf{R}_e^T \bar{\mathbf{p}}_e - \mathbf{p}_e) \quad (4)$$

where indices  $n$  and  $v$  (denoting needle and volume) have been omitted for the sake of simplicity.  $\mathbf{K}_e$  is the local stiffness matrix of an element  $e$ ,  $\bar{\mathbf{p}}$  and  $\mathbf{p}$  are respectively the deformed and the initial vector of positions of element (tetrahedral or beam).  $\mathbf{R}_e$  are the rotation matrices described above.

### B. Time Integration and Constraint-based simulation

The governing equation is given by the following static formulation, where Lagrangian multipliers are used to impose constraints:

$$\mathcal{F}(\mathbf{u}) + \mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathcal{X}, \mathbf{m})\boldsymbol{\lambda} = \mathbf{0} \quad , \quad \text{with} \quad \mathbf{u} = \begin{pmatrix} \mathbf{p}_n \\ \mathbf{p}_v \end{pmatrix} \quad (5)$$

where  $\mathcal{F}$  is a non-linear function providing the internal forces of the needle and the tissue.  $\mathbf{p}_n$  and  $\mathbf{p}_v$  are respectively the positions of needle's and foam's models.  $\boldsymbol{\lambda}$  is the vector of Lagrange multipliers, representing the unknown response forces.  $\mathcal{X}$  is the position of the tool of the robot, and  $\mathbf{m}$  are the positions of the observation used for the registrations step. In addition, equation (5) must satisfy a set of constraints  $\mathcal{H}$  which mathematically can be represented as a non-linear function :

$$\mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathcal{X}, \mathbf{m}) = \boldsymbol{\delta} \quad (6)$$

where  $\boldsymbol{\delta}$  are violations of constraints that must be satisfied for each simulation step.

The non-linear problem (5), is solved using a single iteration Newton-Raphson solver. This choice is motivated by the fact that only small deformations are simulated between two consecutive simulation steps. Therefore, after several simulation steps an equilibrium state (i.e. when  $\Delta \mathbf{p}_n = \mathbf{p}_n^{i+1} - \mathbf{p}_n^i = \mathbf{0}$

and  $\Delta \mathbf{p}_v = \mathbf{p}_v^{i+1} - \mathbf{p}_v^i = \mathbf{0}$ ) provides  $\mathbf{p}_n$ ,  $\mathbf{p}_v$  and  $\boldsymbol{\lambda}$  being the actual solutions of the non-linear problem formulated in the equation (5).

The linearized equation is given by :

$$-\mathcal{F}(\mathbf{u}^i) - \left( \frac{\partial \mathcal{F}}{\partial \mathbf{u}} \Big|_{\mathbf{u}^i} + \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \Big|_{\mathbf{u}^i} \boldsymbol{\lambda}^i \right) \Delta \mathbf{u} = \mathcal{H}(\mathbf{p}_n^i, \mathbf{p}_v^i, \mathcal{X}^i, \mathbf{m}^i) \boldsymbol{\lambda}^i + \left( \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} \Big|_{\boldsymbol{\lambda}^i} \boldsymbol{\lambda}^i + \mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathcal{X}, \mathbf{m}) \right) \Delta \boldsymbol{\lambda} \quad (7)$$

At the beginning of each simulation step  $i$  we assume that all constraints are solved, i.e. no constraint forces are applied to the FE models ( $\boldsymbol{\lambda}^i = \mathbf{0}$ ). Then one can rewrite (7) as:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{u}} \Big|_{\mathbf{u}^i} \Delta \mathbf{u} + \mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathcal{X}, \mathbf{m}) \Delta \boldsymbol{\lambda} = -\mathcal{F}(\mathbf{u}^i) \quad (8)$$

The linearization of the constraint laws (6) gives :

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}} \Big|_{\mathbf{u}^i} \Delta \mathbf{u} = \boldsymbol{\delta}^{i+1} - \boldsymbol{\delta}^i \quad (9)$$

where  $\boldsymbol{\delta}^i = \mathbf{0}$ . In order to simplify the constraint problem, the directions of constraints equations are assumed to be constant during the entire integration step which can be written:

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}} \Big|_{\mathbf{u}^i} \simeq \mathbf{H} \quad \text{and} \quad \mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathcal{X}, \mathbf{m}) \simeq \mathbf{H}^T \quad (10)$$

where  $\mathbf{H}$  is the so-called *Jacobian of the constraints* (see [50] for details). For the sake of simplicity superscript  $i$  will now be omitted.

In a static scenario, stiffness matrices are not invertible because no boundary conditions are applied to the models. An artificial stiffness is added in order to regularize the problem, which corresponds to a damped Newton-Raphson method:

$$\begin{aligned} \mathbf{A}_v &= \frac{\partial \mathcal{F}}{\partial \mathbf{p}_v} \Big|_{\mathbf{p}_v^i} = \mathbf{M}_v + \mathbf{K}_v \\ \mathbf{A}_n &= \frac{\partial \mathcal{F}}{\partial \mathbf{p}_n} \Big|_{\mathbf{p}_n^i} = \mathbf{M}_n + \mathbf{K}_n \\ \mathbf{K}_v^i &= \sum \left( \mathbf{G}_{ve} \mathbf{R}_{ve}^i \mathbf{K}_{ve}^i \mathbf{R}_{ve}^{i T} \mathbf{G}_{ve}^T \right) \\ \mathbf{K}_n^i &= \sum \left( \mathbf{G}_{ne} \mathbf{R}_{ne}^i \mathbf{K}_{ne}^i \mathbf{R}_{ne}^{i T} \mathbf{G}_{ne}^T \right) \end{aligned} \quad (11)$$

where  $\mathbf{K}_v^i$  and  $\mathbf{K}_n^i$  are global stiffness matrices of respectively the volume and the needle.  $\mathbf{G}_{ve}$  and  $\mathbf{G}_{ne}$  are the *globalization matrices* transferring local stiffness  $\mathbf{K}_{ve}^i$  and  $\mathbf{K}_{ne}^i$  to global stiffness matrices.  $\mathbf{M}_n$  and  $\mathbf{M}_v$  are diagonal regularization matrices with non-null values (equal to  $k_n$  and  $k_v$ ) for each line/column corresponding to a constrained DoF (i.e. indices of the base of the needle for  $\mathbf{M}_n$  and indices of the nodes of tetrahedrons embedding a marker for  $\mathbf{M}_v$ ).  $k_n$  and  $k_v$  must be chosen sufficiently high to regularize the problem, we choose a value corresponding to an estimated mass of each model. However, these regularization terms does not modify the converged solution because displacements are then imposed on each DoF impacted by these matrices (see below). Finally, due to the variation of rotation matrices  $\mathbf{R}_{ne}$  and  $\mathbf{R}_{ve}$ , matrices  $\mathbf{A}_v$  and  $\mathbf{A}_n$  are not constant and must be recomputed for each simulation step.

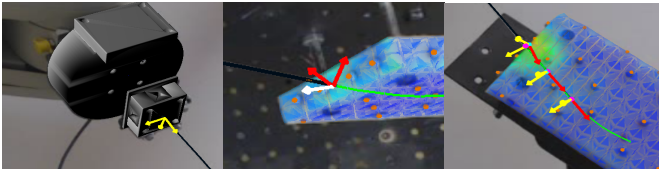
### C. Constraint Definition

Four types of constraints are applied to the models for each forward simulation step:

**1) Bilateral constraints**  $\mathcal{H}_\chi(\mathbf{p}_n, \mathcal{X})$  are used to fix the position of the needle's base to the terminal part of the robot see Fig.3(a).  $\mathcal{H}_\chi$  is a holonomic 6DoFs constraints (position and orientation) whose violation is defined as the relative displacement between the needle's base and  $\mathcal{X}$  the position of the end effector located on the terminal part of the robot. Since the robot is infinitely rigid, only the needle is affected by  $\mathcal{H}_\chi$ .

**2) Penetration constraint**  $\mathcal{H}_\phi(\mathbf{p}_n, \mathbf{p}_v)$  is applied before penetrating the tissue, between the needle's tip and its closest surface on the foam's model see Fig.3(b).  $\mathcal{H}_\phi$  is a 3Dofs 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 [51] for details). This constraint satisfies the Signorini conditions  $\boldsymbol{\lambda} \perp \boldsymbol{\delta}$ , i.e. if objects are distant ( $\boldsymbol{\delta} > 0$ ) any contact force vanishes ( $\boldsymbol{\lambda} = \mathbf{0}$ ), otherwise a positive contact force ( $\boldsymbol{\lambda} > 0$ ) is applied to cancel the penetration ( $\boldsymbol{\delta} = 0$ ).  $\mathcal{H}_\phi$  is parametrized by:  $p_f$ , the puncture force threshold and  $\mu_s$ , the friction coefficient of the surface. Objects are considered to be in contact (leading to a deformation) since  $\boldsymbol{\lambda} < p_f$ , else  $\mathcal{H}_\phi$  is changed for a sliding constraint (see fig 3).

**3) Sliding constraints**  $\mathcal{H}_\psi(\mathbf{p}_n, \mathbf{p}_v)$  enforces the shaft of the needle to follow the path created by advancing the needle tip (see Fig.3(c)).  $\mathcal{H}_\psi$  is a set of 3DoFs constraints: the first component of each constraint applies a resistance to penetration along the needle's shaft, whereas the two other components prevent displacements in the tangential plane. Constraints are dynamically added during the simulation while the needle is inserted. Each constraint is defined by its barycentric coordinates with respect to the tetrahedral mesh, allowing this way their definition at arbitrary location within the volume, without any need for expensive re-meshing.  $\mathcal{H}_\psi$  has two additional parameters,  $0 \leq \mu_n \leq 1$  (0 no friction, 1 sticking), being the penetration resistance coefficient along the shaft and  $d_n$ , the minimum distance between constraints.



(a)  $\mathcal{H}_\chi(\mathbf{p}_n, \mathcal{X})$  (b)  $\mathcal{H}_\phi(\mathbf{p}_n, \mathbf{p}_v)$  (c)  $\mathcal{H}_\psi(\mathbf{p}_n, \mathbf{p}_v)$

Fig. 3. Constraints applied during the needle insertion simulation. Yellow arrows are bilateral constraints. Red arrows are friction constraints. White arrow is a unilateral constraint.

**4) Observation constraint**  $\mathcal{H}_\Omega(\mathbf{p}_v, \mathbf{m})$   $\mathcal{H}_\Omega$  is a set of 3DoFs bilateral constraints, used to register the model of the foam with respect to the observations (see Fig.4). These constraints are applied between observation points  $\mathbf{m}$  given by an external tracking system, and a set of 3D points on model's surface  $\bar{\mathbf{m}}$  segmented during the meshing step. An Iterative Closest Point (ICP) algorithm is used to bind each point of  $\bar{\mathbf{m}}$  with its respective closest point in  $\mathbf{m}$ .

We gather  $\mathcal{H}_\chi(\mathbf{p}_n, \mathcal{X})$ ,  $\mathcal{H}_\phi(\mathbf{p}_n, \mathbf{p}_v)$ ,  $\mathcal{H}_\psi(\mathbf{p}_n, \mathbf{p}_v)$  and  $\mathcal{H}_\Omega(\mathbf{p}_v, \mathbf{m})$ , functions defined above, in the same non-linear function  $\mathcal{H}(\mathbf{p}_n, \mathbf{p}_v, \mathcal{X}, \mathbf{m})$  introduce in (6). After the linearization the *Jacobian of the constraints* is defined as follows:

$$\begin{aligned} \mathbf{H}_n &= \begin{bmatrix} \frac{\partial \mathcal{H}_\chi}{\partial \mathbf{p}_n}; \frac{\partial \mathcal{H}_\phi}{\partial \mathbf{p}_n}; \frac{\partial \mathcal{H}_\psi}{\partial \mathbf{p}_n}; \frac{\partial \mathcal{H}_\Omega}{\partial \mathbf{p}_n} \end{bmatrix} \\ \mathbf{H}_v &= \begin{bmatrix} \frac{\partial \mathcal{H}_\chi}{\partial \mathbf{p}_v}; \frac{\partial \mathcal{H}_\phi}{\partial \mathbf{p}_v}; \frac{\partial \mathcal{H}_\psi}{\partial \mathbf{p}_v}; \frac{\partial \mathcal{H}_\Omega}{\partial \mathbf{p}_v} \end{bmatrix} \end{aligned} \quad (12)$$

It is important to note that thanks to the simplification introduced in equation (10),  $\mathbf{H}$  can be computed only once based on the position of the models  $\mathbf{p}_n^i, \mathbf{p}_v^i, \mathcal{X}^i, \mathbf{m}^i$  at the beginning of each simulation step  $i$ .

Combining equations (8) and (9) with (12) it provides the following Karush-Kuhn-Tucker (KKT) system:

$$\begin{cases} \mathbf{A}_n \mathbf{x}_n + \mathbf{H}_n^T \boldsymbol{\lambda} = \mathbf{b}_n & (13) \\ \mathbf{A}_v \mathbf{x}_v + \mathbf{H}_v^T \boldsymbol{\lambda} = \mathbf{b}_v & (14) \\ \mathbf{H}_n \mathbf{x}_n + \mathbf{H}_v \mathbf{x}_v = \boldsymbol{\delta} & (15) \end{cases}$$

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  $\mathbf{b}_v = -\mathcal{F}_v(\mathbf{p}_v)$ .

### D. Solving Step

At each iteration step, the KKT problem is solved with the following steps:

**1- Free motion:** First, we solve (13) and (14) independently while setting  $\boldsymbol{\lambda} = \mathbf{0}$ , free deformation, i.e.  $\mathbf{x}_n^{\text{free}} = \mathbf{A}_n^{-1} \mathbf{b}_n$  and  $\mathbf{x}_v^{\text{free}} = \mathbf{A}_v^{-1} \mathbf{b}_v$ . We computed  $\mathbf{x}_n^{\text{free}}$  and  $\mathbf{x}_v^{\text{free}}$  using respectively a direct solver and a preconditioned Conjugate Gradient (see [52]). Replacing  $\mathbf{x}^{\text{free}}$  in (13) and (14) gives:

$$\begin{cases} \mathbf{x}_n = \mathbf{x}_n^{\text{free}} - \mathbf{A}_n^{-1} \mathbf{H}_n^T \boldsymbol{\lambda} & (16) \\ \mathbf{x}_v = \mathbf{x}_v^{\text{free}} - \mathbf{A}_v^{-1} \mathbf{H}_v^T \boldsymbol{\lambda} & (17) \end{cases}$$

**2- Constraint definition:** At this step we build both  $\mathbf{H}_n$  and  $\mathbf{H}_v$  (see [51] for details), matrices of constraint directions, 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. Since  $\mathbf{x}_n^{\text{free}}$  and  $\mathbf{x}_v^{\text{free}}$  would be the positions of the needle and the deformable object if any constraint force  $\boldsymbol{\lambda}$  was applied, the violation of constraint  $\boldsymbol{\delta}$  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 (16) and (17) in (15) we obtained the NLCP (Non-linear complementarity problem):

$$(\mathbf{H}_n \mathbf{A}_n^{-1} \mathbf{H}_n^T + \mathbf{H}_v \mathbf{A}_v^{-1} \mathbf{H}_v^T) \boldsymbol{\lambda} = \mathbf{H}_n \mathbf{x}_n^{\text{free}} + \mathbf{H}_v \mathbf{x}_v^{\text{free}} - \boldsymbol{\delta} \quad (18)$$

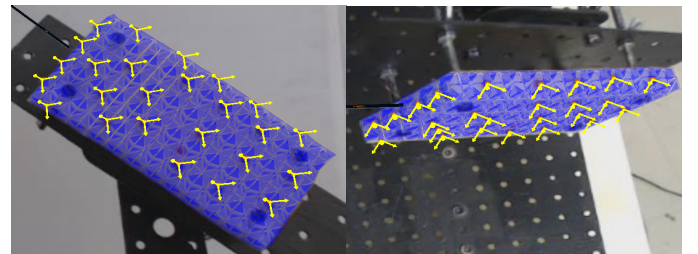


Fig. 4. Constraints  $\mathcal{H}_\Omega$  applied to register the deformable model according to observation points in two different views. Arrows are bilateral constraints.

Solving this problem requires the computation of the *compliance matrix*:

$$\mathbf{W} = \mathbf{H}_n \mathbf{A}_n^{-1} \mathbf{H}_n^T + \mathbf{H}_v \mathbf{A}_v^{-1} \mathbf{H}_v^T \quad (19)$$

**4- Constraints solving:** To solve the NLCP created by the non-linear interactions (frictions and contacts):

$$\mathbf{W} \boldsymbol{\lambda} = \mathbf{H}_n \mathbf{x}_n^{\text{free}} + \mathbf{H}_v \mathbf{x}_v^{\text{free}} - \boldsymbol{\delta} \quad (20)$$

Where both  $\boldsymbol{\lambda}$  and  $\boldsymbol{\delta}$  are unknown, we used a modified Gauss-Seidel algorithm (see [51] for details).

**5- Corrective motion:** Once  $\boldsymbol{\lambda}$  is known it is replaced in equations (13) and (14). Thus we computed  $\mathbf{x}_n$  and  $\mathbf{x}_v$  by inverting again  $\mathbf{A}_n$  and  $\mathbf{A}_v$ . The final position  $\mathbf{p}_n$  and  $\mathbf{p}_v$  that fulfill the contact and friction's laws can then be obtained by integrating  $\mathbf{x}_n$  and  $\mathbf{x}_v$ .

The most critical step to solve the above step at high frequency is the computation of  $\mathbf{W}$ . Although  $\mathbf{W}$  is a small matrix (whose dimension is the square of the number of constraints), its computation is the most time-consuming step since it requires the computation of  $\mathbf{A}_v$ . In order to reach interactive computations (i.e. between 50 to 100 Hz),  $\mathbf{W}$  is computed based on an asynchronous approximation and GPU parallelization as described in [52].

#### IV. CONTROL MODELS

The integration process described in the previous section can be written as a forward non-linear problem  $\mathcal{S}(\mathcal{X}, \mathbf{p}_n, \mathbf{p}_v, \mathbf{m})$ . For any time  $t$ , solving  $\mathcal{S}$  provides positions  $\mathbf{p}_n^{(t)}$  and  $\mathbf{p}_v^{(t)}$  that fulfills constraints  $\mathcal{H}_\chi$ ,  $\mathcal{H}_\phi$ ,  $\mathcal{H}_\psi$  and  $\mathcal{H}_\Omega$  for any position of the robot  $\mathcal{X}^{(t)}$  and observation  $\mathbf{m}^{(t)}$ . Constraints embedded in  $\mathcal{H}_\Omega$  enforce a small displacement error of  $\mathbf{p}_v^{(t)}$  in the neighborhood of the observations  $\mathbf{m}^{(t)}$ ; which significantly decrease the sensibility of the method with respect to mechanical parameters of the volume.

The trajectory is defined as a set of connected points  $\mathbf{s}$  whose positions are given by a linear relation  $\mathbf{s}^{(t)} = \mathbf{N}^T \mathbf{p}_v^{(t)}$ , where  $\mathbf{N}$  is derived from barycentric coordinates of the trajectory with respect to the tetrahedral mesh at the initial step. For any occurring deformation, it is therefore possible to estimate the position of the trajectory in a deformed state, even if the trajectory is virtual and cannot be tracked by any imaging system. On the other hand, the needle's positions  $\mathbf{p}_n^{(t)}$  are entirely driven by both bilateral constraints  $\mathcal{H}_\chi(\mathbf{p}_n, \mathcal{X})$  at the terminal part of the robot  $\mathcal{X}^{(t)}$ , and the constraints of interactions  $\mathcal{H}_\phi(\mathbf{p}_n, \mathbf{p}_v)$  and  $\mathcal{H}_\psi(\mathbf{p}_n, \mathbf{p}_v)$  with foam's model. Therefore, our method does not explicitly require to track the needle in live images, which is often a limitation of existing methods. However, if such observations are available, they may easily be added in the constraint equations.

##### A. Objective function definition

Let  $\mathbf{n}$  be the position of the needle's tip after integration, and  $\mathbf{t}$  be the desired point on the trajectory.  $\mathbf{t}$  is given by a parameter  $c \in [0..1]$  which allows to move  $\mathbf{t}$  from the first to the last point of the trajectory. An inverse problem must be solved in order to compute the next position of the robot  $\mathcal{X}^{(t+1)}$ , minimizing the distance between  $\mathbf{n}^{(t+1)}$  and  $\mathbf{t}^{(t+1)}$ .

In addition, in order to anticipate the displacement of the needle inside the volume, a function  $g(\mathcal{X}, \mathbf{p}_n, \mathbf{p}_v, \mathbf{m})$  is added

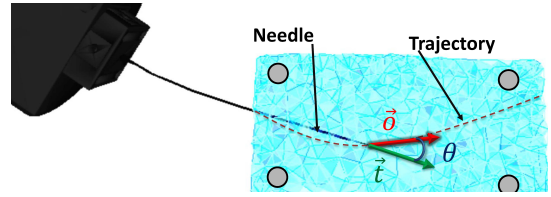


Fig. 5. Angular objective function:  $\theta = \arccos(\text{dot}(\vec{o}, \vec{t}))$  is the angle between needle's tip direction  $\vec{t}$  and the tangent of the target trajectory  $\vec{o}$ .

to cancel the angle between needle's tip direction and the tangent to the target trajectory  $\theta$  ( $\theta = \frac{\partial g}{\partial \mathcal{X}}$  see Fig.5). The objective function  $E(\mathcal{X}, \mathbf{p}_n, \mathbf{p}_v, \mathbf{m})$  of the inverse problem is given by the following equation:

$$E(\mathcal{X}, \mathbf{p}_n, \mathbf{p}_v, \mathbf{m}) = \begin{pmatrix} \mathbf{n} - \mathbf{t} \\ \eta \theta \end{pmatrix} = \mathbf{0} \quad (21)$$

where  $\eta$  is a scalar weighting the constraint function  $g$ , ranging from 0 to 1.

##### B. Inverse kinematics based on FE simulation

Instead of solving the non-linear problem (21), each simulation step  $t$  consist of solving a linearized version of  $E(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)})$  around the current position of the robot  $\mathcal{X}^{(t)}$ :

$$E(\mathcal{X}^{(t)} + d\mathcal{X}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)}) = E(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)}) + \frac{\partial E}{\partial \mathcal{X}} d\mathcal{X} \quad (22)$$

The use of a first order Taylor expansion is motivated by the fact that solution  $\mathcal{X}^{(s)}$  satisfying the non-linear problem may be far from the current position  $\mathcal{X}^{(t)}$  leading to potentially large and complex deformations of structures during the displacement of the robot. Therefore, instead of solving the non-linear equation (21) we rather iteratively solve a linearized version of the problem for each simulation step asynchronously (see below), and update the solution during the displacement of the robot.

The solution of the equation (22) requires the computation of the derivative of  $E$ :

$$E(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)}) + \frac{\partial E}{\partial \mathcal{X}} d\mathcal{X} = \mathbf{0} \quad (23)$$

$$\Rightarrow d\mathcal{X} = \mathbf{J}^{-1} E(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)})$$

where  $\mathbf{J} = -\frac{\partial E}{\partial \mathcal{X}}$  is a  $4 \times 6$  Jacobian matrix which relates the displacement of the tip of the needle and the trajectory with respect to displacements of the base of the robot in Cartesian space.

A numerical derivative is performed providing each component of the Jacobian matrix:

$$\mathbf{J}[:, i] = \frac{E(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)}) - E(\mathcal{X}^{(t)} + \delta \mathcal{X}_i, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)})}{\|\delta \mathcal{X}_i\|} \quad (24)$$

where  $\delta \mathcal{X}_i$  is a small perturbation of the Cartesian coordinates  $i$  of the End Effector.  $\mathbf{J}[:, i]$  is the column  $i$  of the Jacobian and  $E(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)})$  is the value of the objective function at the beginning of the simulation step. A pseudo-inverse (using Singular Value Decomposition) is used to compute  $\mathbf{J}^{-1}$  and solve equation (23).



Due to the high non-linear nature of the problem  $\mathbf{J}$  remains valid for a small amount of time after time  $t$ . Indeed, the constraints equations defined in  $\mathcal{S}$  significantly modify the Jacobian (for instance adding a new penetration constraint  $\phi$  between the volume and the needle) which must be re-computed as soon as possible<sup>1</sup>. Yet, since six independent simulations must be performed to build the entire  $\mathbf{J}$  matrix, each requiring to solve a direct forward problem, it raises significant restrictions on the computation time.

### C. The Control Loop:

In order to maintain reasonable insertion time, we propose an asynchronous control system (see Fig.6).

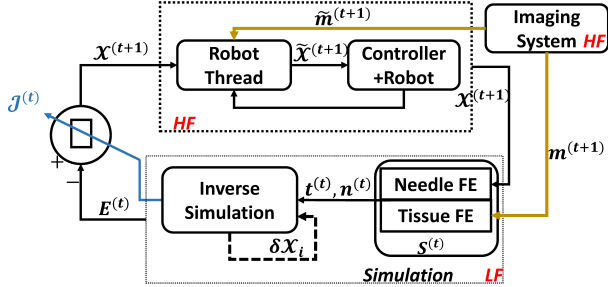


Fig. 6. Control loop diagram; HF:high frequency; LF:low frequency;  $\mathcal{S}^{(t)} = \mathcal{S}(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)})$ ;  $E^{(t)} = E(\mathcal{X}^{(t)}, \mathbf{p}_n^{(t)}, \mathbf{p}_v^{(t)}, \mathbf{m}^{(t)})$ .

The external imaging system provides observations  $\mathbf{m}$  located on the surface of the foam at high frequency. A dedicated thread computes at high frequency a joint-based interpolated motion between the current position of the robot and the desired position of the end effector  $\mathcal{X}^{(t+1)}$  provided by the simulation. Since  $\mathcal{X}^{(t+1)}$  is defined in images' frame coordinates a *Hand and Eye Problem* is solved (using additional observation points  $\tilde{\mathbf{m}}$  attached to the tool of the robot and Umeyama's method [53]) at high frequency in order to estimate the transformation between robot's frame coordinates and observations  $\tilde{\mathbf{m}}$ .

While the robot is moving, the forward problem  $\mathcal{S}$  is solved asynchronously at a lower frequency. Each simulation step takes as input  $\mathcal{X}^{(t)}$  being the position of the end effector of the robot, and  $\mathbf{m}^{(t)}$  begin observations of the model at the time  $t$ . The forward problem  $\mathcal{S}$  provides  $\mathbf{t}^{(t)}$  and  $\mathbf{n}^{(t)}$ , used to compute the residual of the objective function  $E$ . The inverse simulation step is then executed to compute the Jacobian  $\mathbf{J}^{(t)}$  where six independent forward simulations are performed. During these steps,  $\mathcal{X}^{(t)}$  and  $\mathbf{m}^{(t)}$  are considered constant. Based on the Jacobian  $\mathbf{J}^{(t)}$  and the error  $E$  a new desired position of the end effector is computed and sent to the robot's thread asynchronously (i.e. before  $\mathcal{X}^{(t+1)}$  is reached).

### D. Inverse problem in the constraint space

The desired position  $\mathcal{X}^{(t+1)}$  is updated at the frequency of the simulation. The computation of the Jacobian  $\mathbf{J}$  is therefore the main critical step in terms of computation time and must

<sup>1</sup>One may note that the computation of  $\mathbf{J}$  is performed while registration constraints  $\mathcal{H}_\Omega^{(t)}$  are active. This choice allows computing  $\mathbf{J}$  without any need for a priori knowledge of boundary conditions. On the other hand, these constraints artificially stiffen the model in the neighborhood of observation  $\mathbf{m}^{(t)}$ . An expected limitation would be an over-constrained estimation of  $\mathbf{J}$  if the tip of the needle is close to an observation  $\mathbf{m}$ . However, since the trajectory is embedded inside the volume, this is not considered in this paper.

be carefully optimized to maintain acceptable insertion time. In addition, although the computations of the columns of the numerical Jacobian are independent, parallelizing this task would rise technical issues since most of the algorithms used in the forward problem already rely on GPU to reach high frequency computations. Instead we propose to modify the simulation loop (see algorithm 1) in order to perform the computation of inverse steps in constraint's space.

### Algorithm 1: Inverse Simulation Loop

```

1 Free Motion:  $\mathbf{x}^{\text{free}} = \mathbf{A}^{-1}\mathbf{b}$ 
2 Store data( $p_n, p_v, \mathcal{X}$ ):  $\bar{\mathbf{p}} = \mathbf{p}^{(t)}, \bar{\mathcal{X}} = \mathcal{X}^{(t)}$ 
3 Constraint Definition:  $\mathbf{H}$ 
4 Compute Compliance:  $\mathbf{W} = \sum \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^T$ 
5 Compute error:  $\mathbf{e} = E(\bar{\mathcal{X}}, \bar{\mathbf{p}})$ 
6 if  $\text{dot}(\mathbf{e}, \mathbf{e}) < \varepsilon$  then
7   Increment Target:  $c+ = \delta c$ 
8 for  $i = 0$  to 6 do
9   Compute Violation:  $\delta^i = \mathcal{H}(\bar{\mathbf{p}}, \bar{\mathcal{X}} + \delta\mathcal{X}^i, \mathbf{m}^{(t)})$ 
10  Solve Constraints:  $\mathbf{W}\lambda^i = \delta^i$ 
11  Corrective Motion:  $\mathbf{p}^i = \mathbf{x}^{\text{free}} - \mathbf{A}^{-1}\mathbf{H}^T\lambda^i$ 
12  Compute Jacobian:  $\mathbf{J}^i = \frac{E(\bar{\mathcal{X}}, \bar{\mathbf{p}}) - E(\bar{\mathcal{X}} + \delta\mathcal{X}^i, \mathbf{p}^i)}{\|\delta\mathcal{X}^i\|}$ 
13  Reload data:  $\mathbf{p} = \bar{\mathbf{p}}$ 
14  $d\mathcal{X} = \mathbf{J}^{-1} \cdot \mathbf{e}$ 
15 Move End Effector:  $\mathcal{X}^{(t+1)} = \bar{\mathcal{X}} + d\mathcal{X}$ 
16 Compute Violation:  $\delta = \mathcal{H}(\bar{\mathbf{p}}, \mathcal{X}, \mathbf{m}^{(t)})$ 
17 Solve Constraints:  $\mathbf{W}\lambda = \delta$ 
18 Corrective Motion:  $\mathbf{p} = \mathbf{x}^{\text{free}} - \mathbf{A}^{-1}\mathbf{H}^T\lambda$ 

```

The Delassus operator  $\mathbf{W} = \sum \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^T$  does not depend on any parameter that needs to be modified during the computation of  $\mathbf{J}$ . Indeed, the stiffness matrices  $\mathbf{A}$  and the Jacobian of the constraints  $\mathbf{H}$  are both defined using positions of the models at the beginning of the simulation step. Thus, lines 1, 3 and 4 (representing the most time-consuming tasks) can be performed only once per simulation step. Therefore, inverse steps used to compute  $\mathbf{J}$  (line 8 to 14) only requires to:

- 1) Compute a new violation of the constraint  $\delta^i$  for perturbations  $\delta\mathcal{X}^i$  (line 9).
- 2) Solve the constraint problem providing Lagrangian's multipliers  $\lambda^i$  (line 10).
- 3) Project back  $\lambda^i$  in the motion space (line 11).

All these steps can be efficiently performed at interactive frequency since  $\mathbf{W}$  is a small matrix with much less number of degrees of freedom than in motion's space. Additional operation must be introduced in order to store and reload states of models during the computation of the Jacobian. Finally, line 15 to 18 solve the forward problem in order minimize the distance between the needle's tip and the target.

## V. EXPERIMENTAL RESULTS

We validated our approach using the experimental setup shown in Fig.7; it includes i) optical tracking system; ii) monocular camera iii) robotic arm with its needle holder; iv) foam and its support.

**1-Tracking system:** We used an OptiTrack motion-capture system<sup>2</sup> which includes both hardware and software components for the calibration and localization of cameras. The system is composed of 6 Flex13 cameras, arranged around the working space of the robot (see Fig.7(i)). The tracking

<sup>2</sup><http://optitrack.com/>

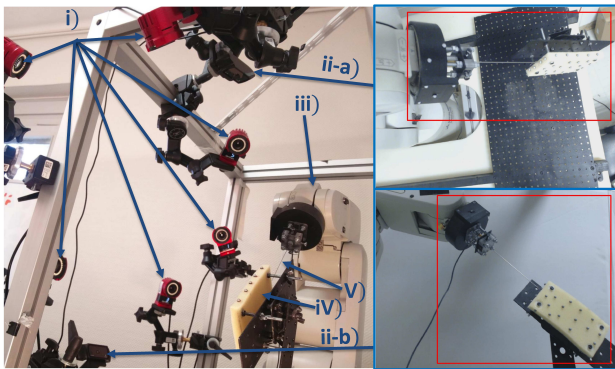


Fig. 7. (left) Experimental setup. (Right) view from the top and front cameras.

system provides the 3D positions (at high-speed sampling, i.e. at least 120 frames per second) of a set of markers placed at the surface of the foam. After the calibration, the system reports back-projection errors of 0.025 mm on average.

**2-Monocular camera:** We are using two monocular cameras (Logitech webcam C920). The positions of the optical cameras and intrinsic parameters are estimated solving the Perspective-n-Point (PnP) problem based on the location of 3D positions provided by optitrack and the 2D coordinates manually segmented in the images. In Fig.7, red rectangles indicate the region of interest of each camera where the calibration was performed. After the registration step, the 2D back-projection error of optical markers is under 1 pixel in the whole zone of interest.

**3-Foam:** The foam is attached and fixed at its four corners within the working space of the robot. The central part is free and can be deformed both tangentially and laterally in order to create complex 3D deformations. The dimensions of the foam are voluntarily long and thin [ $12 \times 6 \times 1$ ] cm in order to create potentially large deformations but also to minimize errors during the non-rigid registration step (which, we recall, is not the contribution of this paper). Yet, it also raises significant difficulty in order to maintain needle tip within the volume of 1cm thick including during large in-depth deformations.

**4-Robot:** A Mitsubishi RV1A anthropomorphic robot arm with 6DoFs is used. The robot includes a 3D printed needle support (see Fig.8), mounted on its end effector. The CAD model was designed to include the location of 6 markers allowing the definition of a rigid body attached to the support. These markers allow solving the Hand and Eye Problem, providing this way the transformation between the robot's frame coordinates and the tracking system coordinates. The location of the base of the needle  $\mathcal{X}$  is known from the CAD model.

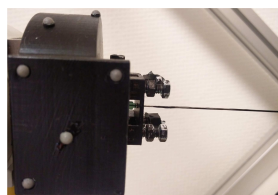


Fig. 8. Needle support.

The accuracy of the system is evaluated solving a kinematic problem in order to position the tip of the needle (assumed rigid) at several 3D positions given by markers. We reported a back-projection error in the monocular cameras views of less than 1.0 pixels, between the needle's tip and markers, showing consistent registration of the overall system.

### A. Needle mechanical characterization

An important advantage of our method lies in the fact that needle positions are derived from a mechanical model and interaction constraints described above, without any need for tracking the needle in live images. Yet, it relies on mechanical parameters that must be identified for an accurate registration of the model. We evaluate these parameters that can be characterized offline.

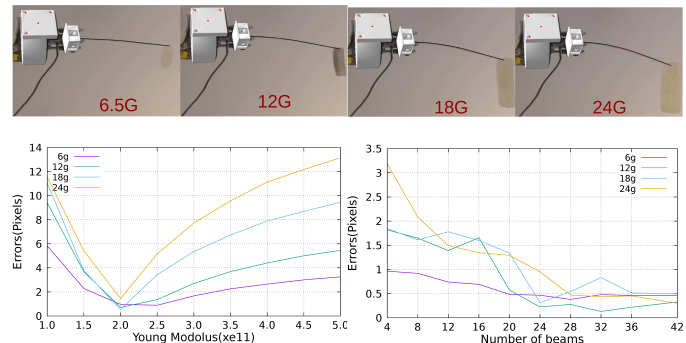


Fig. 9. (Top) The real needle and the model under different loads. (Bottom) The back-projection errors (pixels) in images between the real and the virtual needle deformation for various parameters.

We used a 21 Gauge 12 cm Sterican<sup>®</sup> needle (B. BRAN Melsungen AG). The Poisson ratio  $\nu = 0.3$  is chosen equally to steel [54]. The needle is positioned horizontally (thanks to the robot) with various loads attached to its tip. Several simulations were performed varying the value of  $E$  and the number of beam elements discretizing the model. We measured the back-projection (in pixels) between the simulated model and the real bent needle. We found experimentally  $E = 200GPa$  which is consistent with the literature [35], and a converged solution after 28 beams elements discretizing needle's shaft.

### B. Non-rigid registration

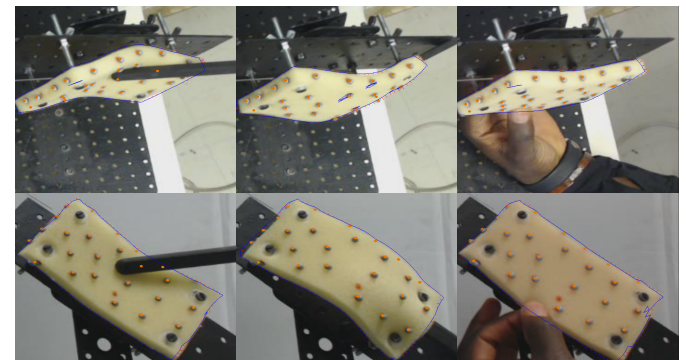


Fig. 10. (Top) top camera view (Down) the front camera view. Three large deformations of the foam were manually created. After registration, the outline of the model is projected on the image and shown in blue.

Contrary to the needle, volume model parameters may not be accurately known during the insertion (boundary conditions, attach points, mechanical parameters...). However, since we impose displacements on the FE model (based on observations  $\mathbf{m}$ ), it significantly decrease the sensitivity of the method with respect to these unknowns.

The Young Modulus of the foam was evaluated at  $E = 1.3kPa$  [55]. Figure 10 shows the accuracy of our approach using 25 markers uniformly distributed at the surface of

the foam. We measured the Hausdorff distance between the projected contour of the model and the outline of the foam manually segmented in images. A mean error varying between **1** and **3 pixels** is reported with maximum values of **6 pixels**. A sensitivity study was performed varying  $E \pm 20\%$  (which corresponds to the standard uncertainty of liver's parameters in healthy subjects [56]), without a significant impact on the registration accuracy.

## VI. CONTROL STRATEGY AND VALIDATION

An important difficulty to evaluate our method lies in the fact that the trajectory is virtual and only known in the undeformed state. Therefore, there is no ground truth about the actual path taken by the needle. In order to evaluate our method we propose both a synthetic and a real validation study (see Fig 11). In both scenarios, the inverse loop pro-

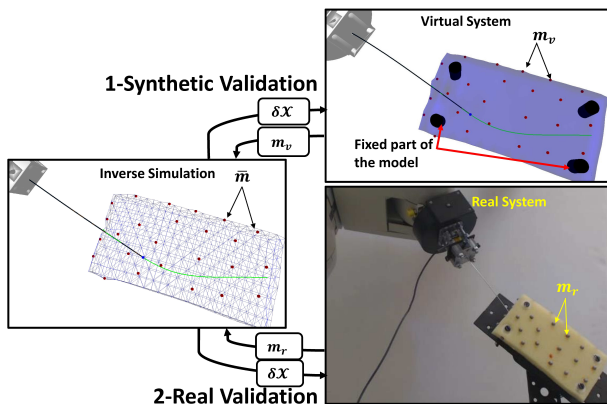


Fig. 11. Real and synthetic validation. The Inverse Simulation computes displacements of the robot while observation points  $\mathbf{m}$  are used to register the models.

vides displacements  $d\mathcal{X}$  of the base of the robot using the method introduced in this paper. For the synthetic scenario, the displacement  $d\mathcal{X}$  is sent to a numerical simulation of a needle insertion (corresponding to forward steps described in section III). This numerical simulation is independent from the inverse loop (and may be parametrized differently); while a set of observations  $\mathbf{m}_v$  is sent back to the inverse loop to perform the corrective step. For the real scenario, the displacement  $d\mathcal{X}$  is sent to the robot, inserting a real needle into the foam. The 3D position of markers  $\mathbf{m}_r$  is sent back to the inverse loop for the corrective step.

### A. Synthetic validation and sensitivity analysis

Independently from mechanical parameters (Young's Modulus, Poisson's ratio) our method relies on additional parameters being  $\mu_n \in [0 : 1]$  the friction coefficient between the foam and the needle,  $d_n$  the distance between sliding constraints. Experimentally we choose  $\mu_n = 0.1$ ,  $d_n = 0.007 \text{ mm}$  and a tetrahedral mesh of **2592** elements. Using the same parameters in both the virtual system and the inverse loop, we measured an average distance of **0.75 mm** (max **1.2 mm**) between needle's and the desired trajectory for an insertion of **8 cm**, which remains compatible with most of clinical applications. At the end of the insertion a maximal Von Mises Stress of **41.509 kPa** was reported on the tetrahedral mesh, showing large deformations during the insertion.

We modified successively each parameter in the inverse loop

(but not in the virtual simulation), in order to evaluate their impact on the control method. The young modulus of the foam was modified  $E \pm 20\%$  providing an average error ranging from **0.75 mm** to **1.1 mm** at the end of the insertion. This range of error remains acceptable for clinical applications, and is mainly due to the corrective step. The same conclusion was obtained for the friction coefficient since variations of **0.4 mm** have been reported on the error for values ranging from  $\mu_n \in [0: 0.6]$ . For larger values, the Jacobian obtained from the inverse loop didn't provide any displacement allowing for the insertion (due to sticky conditions). The system falls into local minimum without being able to move forward the needle. However, one may note that our method didn't diverge and the inserted part of the needle remained close (below **1.1 mm**) to the trajectory. The parameter  $d_n$  has a higher influence on the accuracy since the needle can move independently of the volume between the constraints. Varying this parameter from **5 mm** to **40 mm** provides an average error between **0.79 mm** and **3.4 mm**. Decreasing the distance  $d_n$  improves the accuracy of the method, but over-constrained problems may occur if  $d_n$  is smaller than the element size of the mesh.

### B. Evaluation of the accuracy during needle insertion

We now evaluate the accuracy of our method in a real scenario. In order to define the trajectory and enforce its feasibility, we manually inserted the needle where a metallic thread was beforehand slipped within the shaft. During the manual insertion, deformations were applied on both the needle and the foam which created a curved path. After the insertion, the needle was removed letting the metallic wire within the volume. A CT scan of the foam including the metallic thread and markers were performed and segmented to create FE meshes.

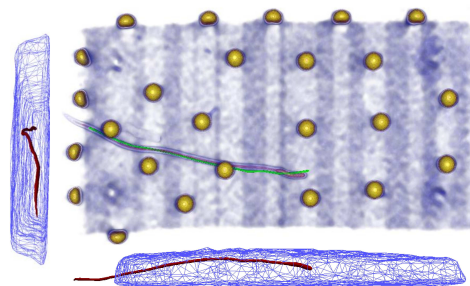


Fig. 12. CT scan after the robotic insertion. The desired trajectory is shown in green and the path taken by the needle is shown in red.

A robotic insertion was then performed to automatically follow the desired path. The foam was attached at its four corners allowing this way for vertical and lateral deformations of the internal part. Our control method was used to drive the robot and follow the trajectory (estimated through the registration of the models). During the insertion vertical and lateral deformations were applied (see fig 13) leading to significant modification of the undeformed trajectory, important bending of the needle (we measured a bending of more than  $30^\circ$  at the base of the needle at the end of the insertion) and even an off-plane shift of approximately 2cm between the base of the needle and the insertion point. Despite these important modifications, the method maintained the tip of the needle



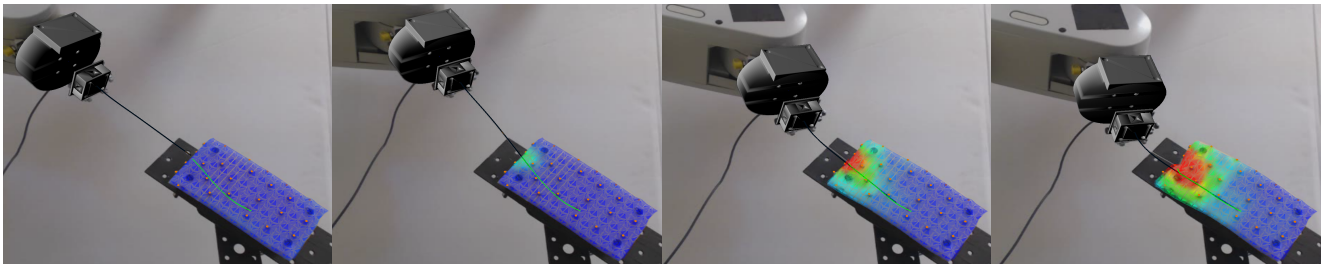


Fig. 13. Augmented reality view at different steps of the insertion. The desired trajectory is shown in green. The model of the needle is shown in black and overlay the real needle (gray) in the projective view. The color map on the foam shown Von Mises stress indicating the deformation of the model with respect to the initial configuration.

within the thickness of  $1\text{ cm}$  of the volume, and followed the desired path without any human intervention. Before the insertion, another metallic thread was placed within the shaft of the needle, allowing for error measurement between manual and robotic paths (see Fig. 12). We reported an average error along the trajectory of  $1.62\text{ mm}$  with a maximum error of  $3.73\text{ mm}$  which is acceptable for a medical application.

### C. Computation Time

Finally, the computational time and the percentage of main simulation steps are reported in the following table. The Constraints Solving step remains the most expensive task since it must be performed 7 times for the inverse loop (6 for the Jacobian estimation, and 1 for the forward step). Performing inverse steps in constraint space (see section IV-D) provides a speedup of 4.2 compared to a version where inverse steps are performed in motion space (where all the operations are executed 7 times). The overall method provides on average 94 input commands per second.

Process	min(ms)	mean(ms)	max(ms)	time%
FM	2.26	2.95	4.76	25.62
CD	0.19	0.22	0.34	1.91
CC	2.46	2.61	3.18	22.67
SC( $\times 7$ )	4.29	4.84	6.72	42.07

Fig. 14. Computation time and percentage of main simulation steps. FM: Free Motion; CD: Constraint Definition; CC: Compliance Computation; SC: Solve Constraints

The maximum velocity of the robot is set to  $25\text{ mm}$  per second, resulting in new Jacobian computation for a maximal displacement of  $0.26\text{ mm}$  of the base of the robot, which is sufficient for a quasi-static scenario. Finally, the total insertion time has been performed in 6 minutes, which is similar to what is obtained by other methods [57], allowing this way for clinical applications.

## VII. DISCUSSION

We proposed a new approach for an automatic control of a robotic needle insertion in deformable environment. Input commands are derived from an inverse FE simulations allowing for the prediction of the behavior of deformable structures. Errors of FE models with respect to real material are controlled thanks to a non-rigid registration performed at high frequency. We proposed a constraint-based formulation allowing for the computation of inverse steps in constraint space providing this way total insertion time compatible with clinical applications.

In order to bring the system to the operating room the main limitation lies in the image modality and image processing issues providing reliable per-operative observation points, nec-

essary for the corrective step. We are currently considering the possibility to combine this work with our preliminary study [46] where similar markers were used for Augmented Reality during an open liver surgery.

## REFERENCES

- [1] N. J. Cowan, K. Goldberg, G. S. Chirikjian, G. Fichtinger, R. Alterovitz, K. B. Reed, V. Kallem, W. Park, S. Misra, and A. M. Okamura, "Robotic needle steering: Design, modeling, planning, and image guidance," *Surgical Robotics*, pp. 557–582, 2011.
- [2] J. Troccaz, *Medical robotics*. ISTE Ltd., 2012.
- [3] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [4] J. W. Owen, K. J. Fowler, M. B. Doyle, N. E. Saad, D. C. Linehan, and W. C. Chapman, "Colorectal liver metastases: disappearing lesions in the era of Eovist hepatobiliary magnetic resonance imaging," *HPB*, vol. 18, no. 3, pp. 296–303, mar 2016.
- [5] N. Abolhassani, R. Patel, and M. Moallem, "Needle insertion into soft tissue: A survey," *Medical Engineering and Physics*, no. 4, 2007.
- [6] D. R. Kaye, D. Stoianovici, and M. Han, "Robotic ultrasound and needle guidance for prostate cancer management: review of the contemporary literature," *Current opinion in urology*, vol. 24, no. 1, pp. 75–80, 2014.
- [7] I. Elgezua, Y. Kobayashi, and M. G. Fujie, "Survey on current state-of-the-art in needle insertion robots: Open challenges for application in real surgery," *Procedia CIRP*, vol. 5, pp. 94–99, 2013.
- [8] 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*, vol. 34, 2010.
- [9] Z. Neubach and M. Shoham, "Ultrasound-guided robot for flexible needle steering," *IEEE Transactions on Biomedical Engineering*, 2010.
- [10] S. H. Okazawa, R. Ebrahimi, J. Chuang, R. N. Rohling, and S. E. Salcudean, "Methods for segmenting curved needles in ultrasound images," *Medical Image Analysis*, 2006.
- [11] R. Seifabadi and C. Laboratory for Percutaneous surgery (Perk Lab), Queen's University, Kingston, "Real-time tracking of a bevel-tip needle with varying insertion depth: Toward teleoperated MRI-guided needle steering," *IROS*, pp. 469–476, 2013.
- [12] N. Navab, B. Bascle, M. Loser, B. Geiger, and R. Taylor, "Visual servoing for automatic and uncalibrated needle placement for percutaneous procedures," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 327–334.
- [13] M. Abayazid, R. J. Roesthuis, R. Reilink, and S. Misra, "Integrating deflection models and image feedback for real-time flexible needle steering," *IEEE Transactions on Robotics*, vol. 29, no. 2, 2013.
- [14] B. Kim, J. Ha, F. C. Park, and P. E. Dupont, "Optimizing curvature sensor placement for fast, accurate shape sensing of continuum robots," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5374–5379, 2014.
- [15] J. S. Kim, J. Guo, M. Chatrasingh, S. Kim, and I. Iordachita, "Shape determination during needle insertion with curvature measurements," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-September, pp. 201–208, 2017.
- [16] V. G. Mallapragada, N. Sarkar, and T. K. Podder, "Robot-assisted real-time tumor manipulation for breast biopsy," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 316–324, 2009.
- [17] M. C. Bernardes, B. V. Adorno, P. Poignet, and G. A. Borges, "Robot-assisted automatic insertion of steerable needles with closed-loop imaging feedback and intraoperative trajectory replanning," *Mechatronics*, vol. 23, no. 6, pp. 630–645, 2013.

- [18] A. Krupa, "A new duty-cycling approach for 3D needle steering allowing the use of the classical visual servoing framework for targeting tasks," *International Conference on Biomedical Robotics and Biomechanics (EMBS)*, pp. 301–307, 2014.
- [19] R. Secoli and F. Rodriguez y Baena, "Adaptive path-following control for bio-inspired steerable needles," in *IEEE BioRob*. IEEE, jun 2016.
- [20] K. B. Reed, A. Majewicz, V. Kallem, R. Alterovitz, K. Goldberg, N. J. Cowan, and A. M. Okamura, "Robot-Assisted Needle Steering," *IEEE Robotics and Automation Magazine*, no. 4, 2011.
- [21] S. Misra, K. T. Ramesh, and A. M. Okamura, "Modeling of Tool-Tissue Interactions for Computer-Based Surgical Simulation: A Literature Review," *Presence: Teleoperators and Virtual Environments*, vol. 17, no. 5, pp. 463–491, 2008.
- [22] D. C. Rucker, J. Das, H. B. Gilbert, P. J. Swaney, M. I. Miga, N. Sarkar, and R. J. Webster, "Sliding mode control of steerable needles," *IEEE Transactions on Robotics*, 2013.
- [23] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O'Brien, "Interactive simulation of surgical needle insertion and steering," in *ACM Transactions on Graphics*, vol. 28, no. 3, 2009, p. 1.
- [24] C. Duriez, C. Guébert, M. Marchal, S. Cotin, and L. Grisoni, "Interactive simulation of flexible needle insertions based on constraint models," *Lecture Notes in Computer Science*, pp. 291–299, 2009.
- [25] W. Park, Y. Liu, Y. Zhou, M. Moses, and G. S. Chirikjian, "Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map," *Robotica*, no. 4, jul 2008.
- [26] X. Jijie, V. Duindam, R. Alterovitz, and K. Goldberg, "Motion planning for steerable needles in 3D environments with obstacles using Rapidly-exploring random trees and backchaining," *4th IEEE Conference on Automation Science and Engineering, CASE 2008*, pp. 41–46, 2008.
- [27] V. Duindam, J. Xu, R. Alterovitz, S. Sastry, and K. Goldberg, "3D motion planning algorithms for steerable needles using inverse kinematics," in *Springer Tracts in Advanced Robotics*, vol. 57. NIH Public Access, 2010, pp. 535–549.
- [28] R. Alterovitz, M. Branicky, and K. Goldberg, "Motion planning under uncertainty for image-guided medical needle steering," in *International Journal of Robotics Research*, vol. 27, no. 11-12, 2008, pp. 1361–1374.
- [29] S. Wen and R. Alterovitz, "Motion planning under uncertainty for medical needle steering using optimization in belief space," in *IEEE International Conference on Intelligent Robots and Systems*, 2014.
- [30] N. Hamzé, I. Peterlík, S. Cotin, and C. ESSERT, "Preoperative trajectory planning for percutaneous procedures in deformable environments," *Computerized Medical Imaging and Graphics*, vol. 47, pp. 16–28, 2016.
- [31] P. Li, S. Jiang, D. Liang, Z. Yang, Y. Yu, and W. Wang, "Modeling of path planning and needle steering with path tracking in anatomical soft tissues for minimally invasive surgery," *Medical Engineering and Physics*, vol. 41, pp. 35–45, 2017.
- [32] S. P. S. DiMaio and S. S. E. Salcudean, "Needle steering and motion planning in soft tissues," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 6, pp. 965–974, jun 2005.
- [33] D. Glozman and M. Shoham, "Image-guided robotic flexible needle steering," *IEEE Transactions on Robotics*, vol. 23, no. 3, 2007.
- [34] L. Barbé, B. Bayle, M. de Mathelin, and A. Gangi, "Needle Insertions Modeling: Identifiability and Limitations," *Biomedical Signal Processing and Control*, vol. 2, no. 3, pp. 191–198, 2007.
- [35] M. Khadem, B. Fallahi, R. Carlos, R. Sloboda, N. Usmani, and M. Tavakoli, "A Mechanics-based Model for Simulation and Control of Flexible Needle Insertion in Soft Tissue," *IEEE ICRA*, 2015.
- [36] Y. Kobayashi, A. Onishi, T. Hoshi, K. Kawamura, and M. G. Fujie, "Viscoelastic and nonlinear organ model for control of needle insertion manipulator," *IEEE Engineering in Medicine and Biology Society*, 2007.
- [37] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," *IEEE ICRA*, pp. 2550–2555, 2015.
- [38] I. Peterlík, H. Courtecuisse, R. Rohling, P. Abolmaesumi, C. Nguan, S. Cotin, and S. Salcudean, "Fast elastic registration of soft tissues under large deformations," *Medical Image Analysis*, vol. 45, pp. 24–40, 2018.
- [39] B. Marami, S. Sirouspour, and D. W. Capson, "Non-rigid registration of medical images based on estimation of deformation states," *Physics in Medicine & Biology*, vol. 59, no. 22, p. 6891, 2014.
- [40] P. Pratt, D. Stoyanov, M. Visentini-Scarzanella, and G.-Z. Yang, "Dynamic guidance for robotic surgery using image-constrained biomechanical models," *MICCAI*, vol. 13, no. Pt 1, 2010.
- [41] S. Suwelack, S. Röhl, S. Bodenstedt, D. Reichard, R. Dillmann, T. Dos Santos, L. Maier-Hein, M. Wagner, J. Wünscher, H. Kennigott, B. P. Müller, and S. Speidel, "Physics-based shape matching for intraoperative image guidance," *Med. Phys.*, vol. 41, no. 11, p. 111901, oct 2014.
- [42] R. Plantefève, I. Peterlík, N. Haouchine, and S. Cotin, "Patient-specific Biomechanical Modeling for Guidance during Minimally-invasive Hepatic Surgery," *Annals of Biomedical Engineering*, 2015.
- [43] R. Trivisonne, I. Peterlík, S. Cotin, and H. Courtecuisse, "3D Physics-Based Registration of 2D Dynamic MRI Data," in *MMVR*, 2016.
- [44] F. Morin, H. Courtecuisse, I. Reinertsen, F. L. Lann, O. Palombi, Y. Payan, and M. Chabanas, "Brain-shift compensation using intraoperative ultrasound and constraint-based biomechanical simulation," *Medical Image Analysis*, 2017.
- [45] Y. Adagolodjo, L. Goffin, M. De Mathelin, and H. Courtecuisse, "Inverse real-time Finite Element simulation for robotic control of flexible needle insertion in deformable tissues," in *IROS*, 2016, pp. 1–6.
- [46] Y. Adagolodjo, N. Golse, V. Eric, C. Stephane, and H. Courtecuisse, "Marker-based Registration for Large Deformations - Application to Open Liver Surgery -," *IEEE ICRA*, vol. 6, 2018.
- [47] U. Goerguelue, "BEAM THEORIES The difference between Euler-Bernoulli and Timoshenko," *Bernoulli*.
- [48] M. Nesme, Y. Payan, and F. Faure, "Efficient, physically plausible finite elements," *EUROGRAPHICS*, 2005.
- [49] C. A. Felippa and B. Haugen, "A unified formulation of small-strain corotational finite elements: I. Theory," pp. 2285–2335, 2005.
- [50] C. Duriez, "Real-time haptic simulation of medical procedures involving deformations and device-tissue interactions," *HDR*, 2013.
- [51] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot, "Realistic Haptic Rendering of Interacting Deformable Objects in Virtual Environments," *IEEE Transactions on Visualization and Computer Graphics*, 2006.
- [52] H. Courtecuisse, S. Cotin, J. Allard, and L. Soler, "GPU-based interactive simulation of liver resection," in *ACM SIGGRAPH 2011*. New York, USA: ACM Press, 2011.
- [53] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, apr 1991.
- [54] E. Dehghan, O. Goksel, and S. E. Salcudean, "A comparison of needle bending models," *MICCAI*, vol. 9, no. Pt 1, 2006.
- [55] W. Witkiewicz and A. Zielinski, "Properties of the Polyurethane (PU) Light Foams," *Advances in Materials Science*, vol. 6, no. 2 (10), pp. 35–51, 2006.
- [56] D. Roulot, S. Czernichow, H. Le Clésiau, J. L. Costes, A. C. Vergnaud, and M. Beaugrand, "Liver stiffness values in apparently healthy subjects: Influence of gender and metabolic syndrome," *Journal of Hepatology*, vol. 48, no. 4, pp. 606–613, apr 2008.
- [57] B. Schulz, K. Eichler, P. Siebenhandl, T. Gruber-Rouh, C. Czerny, T. J. Vogl, and S. Zangos, "Accuracy and speed of robotic assisted needle interventions using a modern cone beam computed tomography intervention suite: A phantom study," *European Radiology*, vol. 23, no. 1, pp. 198–204, 2013.



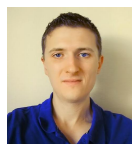
**Yinoussa Adagolodjo** obtained a M.S. degree in mathematics in 2011 in Lomé University (Togo). He arrived in France in September 2012 and studied mathematical engineering and received a M.S. degree in computer science in June 2014. Then he started a PhD in the AVR/ICube team in Strasbourg (France) on medical simulation in February 2015, in collaboration with the Mimesis team. He received his Ph.D degree in september 2018.



**Laurent Goffin** received in 2000, the M.S. degree in medical computer science from the University Pierre et Marie Curie, Paris VI, France. Then, he worked as biomedical software engineer at INRIA Sophia Antipolis and at IRCAD Strasbourg. Since 2010, he joined the Control, Vision and Robotics Team at ICube Laboratory, University of Strasbourg, France. His research interests include augmented reality, computer vision and robotics.



**Michel de Mathelin** received the Electr. Eng. degree from Louvain University, Belgium, in 1987 and the M.S. and Ph.D. degrees in electrical and computer engineering from the Carnegie Mellon University, Pittsburgh, PA, in 1988 and 1993. Since 1999, he has been Prof. at Télécom Physique Strasbourg. He was head of the AVR research group (60 people) from 2000 to 2012. Since 2013, he is the Dir. of the ICube Laboratory (UMR CNRS-UNISTRA 7357) with more than 650 people in Engineering and Computer Science with a strong focus on biomedical and environmental Eng. In 2017, he became vice-president for innovation and technology transfer of the University of Strasbourg.



**Hadrien Courtecuisse** received his Ph.D degree in the SHAMAN team at Inria in 2011. Then he worked during one year as a research associate at Cardiff University in the Institute of Mechanics and Advanced Materials. He became Research Scientist in 2014 in the team AVR/ICube at CNRS in Strasbourg and joined the Mimesis project at Inria. His reseach interests include real-time FE models, linear algebra, collision detection and response, augmented reality and robotic. He is an active contributor of the SOFA framework.