Real-time Robotic Flexible Needle Insertion In Deformable Living Organs Using Isolated Objective Constraint

Thuc Long Ha, Julien Bert, Hadrien Courtecuisse

Real-time Robotic Flexible Needle Insertion In Deformable Living Organs Using Isolated Objective Constraint

Thuc Long HA[1] and Julien Bert[2] and Hadrien Courtecuisse [1]

***Abstract*— This paper presents an innovative approach for executing robotic needle insertion within deformable living organs. The objective is to maintain the insertion pivot point on the skin, which remains stationary. At the same time, the organs undergo displacement and deformation due to respiration. Therefore, real-time control and precise needle steering are crucial. The proposed method relies on isolated objective constraints to ensure the objectives while steering the needle along a predefined trajectory. The needle insertion process benefits from Finite Element (FE) models to simulate the environment and address the inverse problem to drive the robot's end effector (EE) by re-evaluating the objective functions in the constraint space for each time step. So, the desired motion of the robot's EE could be calculated at a small cost for non-linear functions in real-time, resulting in better precision and reducing stress caused to the organs.**

***Index Terms*— Finite Element Simulation, Flexible Needle Steering, Medical Robotics**

I. INTRODUCTION

Liver cancer can be treated with various percutaneous approaches like chemotherapy or radiofrequency ablation (RFA) In cancer treatment, rigid needles are commonly used [1], but anatomical structures like bone or blood vessels within the liver can prevent direct tumor access. Furthermore, organ deformation during insertion or needle movement can cause bending [2]. Robotic solutions designed to assist needle placement in percutaneous procedures were developed in [3], and they have been certified for clinical acceptability and usability. These solutions provide guidance for needle insertion using a 2-DOFs automated remote center of motion. With robotic arms, needle steering becomes feasible, enabling the insertion of flexible needles through curved and intricate trajectories [4]. Despite introducing various robotic systems for percutaneous procedures [5], they encounter significant challenges in navigating dynamic and deformable organs. Accurate estimation of organ deformation is crucial to prevent undesired outcomes, necessitating robust robotic methods for real-time adaptation [6].

A. Visual servoing

Needles and soft tissue interaction require precise localization of needles in images [7]. Visual servoing guides the needle during percutaneous therapies, considering obstacles and enabling steering through complex trajectories. Real-time information from intraoperative medical images facilitates control over the needle's movement. Challenges associated with non-real-time modalities like MRI and CT have led to exploring solutions such as fluoroscopy and ultrasound. Visual servoing using X-ray or MRI faces difficulties in obtaining real-time images and constraints related to metallic materials. Hence, researchers propose designing robots compatible with imaging modalities and patient attachment to address material issues [8]. Effective visual servoing strategy and performance evaluation have led to integrating controllers like proportional-integral-derivative (PID) [9].

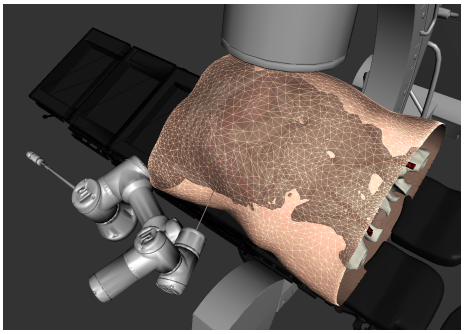


Fig. 1: Flexible Needle Insertion Simulation with robot

B. Needle-based control

The most common method to control a needle's trajectory within soft tissue involves manipulating its base while managing the six degrees of freedom (DOFs). Once the needle is inserted into the tissue, any lateral movement of the base orthogonal to the insertion axis causes both the needle and tissue to deform, consequently altering the insertion trajectory by deviating the tip. Other approaches, such as the duty-cycling technique, utilize beveled-tip flexible needles to control three DOFs (two rotations and one translation) of the tip, enabling the creation of curved paths [10].

C. Needle-tissue interaction modeling

Needle insertion into soft tissue involves complex challenges, including bending, tissue deformation, and needle-tissue interaction. Models describing these interactions can be classified into mechanical, finite element method (FEM), and kinematic models.

Mechanical models: A well-defined mechanical model captures tissue deformation, providing a foundation for research and path planning [11]. In [2], forces are categorized for needle insertion, validated through bovine liver insertions and analysis of the needle geometry's impact. Additionally, [12] proposes a friction force model based on experiments with

*This work is supported by French National Research Agency in the scope of the the Investissements d'Avenir program Labex CAMI (ANR-11-LABX-0004)

[1]ICube Laboratory, UMR 7357, CNRS - University of Strasbourg, France
[2]Laboratory of Medical Information Processing - Latim INSERM, UMR 1101, Brest, France

different velocities. Moreover, a 3D needle-tissue interaction model [13] utilizes patient-specific parameters to predict force, target movement, and deflection.

Kinematic models: Methods based on kinematic models enable the consideration of the specific geometry of needles, resulting in a more intricate relationship. Similarly, dynamic models can account for needle deformation caused by the surrounding tissue, leading to deviation [14].

Finite Element Method: Previous research has introduced methods to manipulate flexible needles in deformable tissues, improving the accuracy of needle steering [15]. This numerical solution enables the computation of complex inverse Finite Element (iFE) simulations to predict deformation. The method relies on constraint objective functions to reduce the computation cost of the inverse simulation loop, which has been introduced in [16] to adapt to the simulation environment and enhance stability, considering organ movements.

D. Contribution

This article introduces an extension of previous methods [16], [15] that derive robotic control by solving iFE problems. We introduce the concept of *isolated objective constraints* to enhance the accuracy and computation time of previous approaches. Indeed, we aim to evaluate nonlinear objective functions as introduced in [16]. However, the nonlinear objective functions were linearized in [15] to save computation time. It was shown that this reduction in computation time was necessary to maintain the stability of needle steering in a dynamic context where models were subject to respiratory motion. By relying on *isolated objective constraints*, we provide a new solution that addresses non-linearity problems while maintaining computation time, thus enforcing stability in needle steering. Main contributions include an optimized inverse simulation loop to steer the needle inside a moving and deformable object in real time and a set of objective functions to consider in the loop to achieve the criteria in complex scenarios.

II. Background

Due to space constraints, we limit the description of this section to the strictly necessary elements for understanding the method. However, we refer the interested reader to [16] for in-depth details. The FE model of the needle is based on Timoshenko's beam theory. The needle is discretized into one-dimensional beams, and each node admits 6 degrees of freedom (DOFs). A corotational formulation of Hooke's law is used for the deformable object [17]. This ensures rapid calculation of the stiffness matrix while accounting for a significant portion of the nonlinearity arising from rotational elements. This formulation partly eliminates the artifact of the increase in the volume of an elastic linear object.

Lagrange multipliers enforce displacement or simulate the interaction between two deformable objects. These constraints must be satisfied at the end of each time step. With the interactions between two objects, the mechanical problem can be formulated as a Karush-Kuhn-Tucker (KKT) system:

$$\begin{cases} \mathbf{A}_1\mathbf{x}_1 + \mathbf{H}_1^T\boldsymbol{\lambda} = b_1 \\ \mathbf{A}_2\mathbf{x}_2 + \mathbf{H}_2^T\boldsymbol{\lambda} = b_2 \\ \mathbf{H}_1\mathbf{x}_1 + \mathbf{H}_2\mathbf{x}_2 = \boldsymbol{\delta} \end{cases} \tag{1}$$

Where $\mathbf{A}_i$ is the global matrix generated from the sum of the stiffness and regularization matrices. Dirichlet conditions enforce $\mathbf{A}_i$ to be invertible. $\mathbf{A}_i$ is not constant and must be recomputed at each time step. $\mathbf{K}_i$ is the sum of the stiffness and regularization matrices with various coefficients. $\mathbf{x}_i$ is the displacement of the nodes of objects (needles, organs) $\mathbf{H}_i$ is the Jacobian of the constraints connecting the constraint space to the motion space, $\boldsymbol{\lambda}$ represents the Lagrange multipliers, and $\boldsymbol{\delta}$ is the constraint violation.

In order to solve the KKT system (1), four-step methods from [18] have been employed. Initially, the *free motion* is addressed by solving the first two equations and finding $\mathbf{x}^{free}$ without considering the constraints (with $\boldsymbol{\lambda} = \mathbf{0}$). Subsequently, after defining $\mathbf{H}_i$, the system becomes:

$$\mathbf{W}\boldsymbol{\lambda} = \boldsymbol{\delta} \tag{2}$$

where $\mathbf{W} = \sum_{n\in 1,2} \mathbf{H}_n\mathbf{A}_n^{-1}\mathbf{H}_n^T$ is the compliance matrix, and $\boldsymbol{\delta} = \sum_{n\in 1,2} \mathbf{H}_n\mathbf{x}_n^{free}$ is calculated after free motion. $\mathbf{W}$ is also known as the Delassus operator, expressing the mechanical coupling across the models in the *constraint space*, i.e., a smaller space whose dimension is equal to the number of constraints but contains all the necessary information needed to solve the coupled problem between $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$. The computation of $\mathbf{W}$ is time-consuming, but it can be optimized using GPU-based parallelization [19].

A. Automatic needle insertion control

The control method is expressed as a minimization problem of functions $E(X^t, p_n, p_t)$ of the inverse problem to calculate the errors vector $\overline{\mathbf{e}}$ of the objective functions.

$$E(X^t, p_n, p_t) = 0 \tag{3}$$

The purpose of the objective functions is to find the displacement of the target organs, which is the tumor position, to cancel. Therefore, the following equation must be solved in each time step to calculate the robotic command using first-order Taylor expansion:

$$E(X^t + dX, p_n, p_t) = E(X^t, p_n, p_t) + \frac{\delta E}{\delta X}dX = 0 \tag{4}$$

Therefore, the displacement of the end effector minimizing the objective functions is obtained by solving $dX = \mathbf{J}^{-1}E(X^t, p_n, p_t)$, where $\mathbf{J}$ is a Jacobian matrix linking the displacement of the tip of the needle to the trajectory to the displacement of the robot's end effector (EE) in Cartesian coordinates. It is derived numerically by:

$$\mathbf{J}^i = \frac{E(X^t + \Delta X_i, p_n, p_t) - E(X^t - \Delta X_i, p_n, p_t)}{2\|\Delta X^i\|} \tag{5}$$

Where ΔX_i is the small perturbation of the Cartesian coordinate $\mathbf{I}$.

The above equation requires solving a complete FE elements simulation to compute each independent column of $\mathbf{J}$,

which is highly time-consuming. To maintain performance compatible with robotic control, [16] proposed solving the inverse problem directly in the constraint space. However, projection on the degrees of freedom (DOFs) of the numerical models (i.e., solving $\mathbf{A^{-1}H^{T}\lambda}$) is used to evaluate the objective functions. Conversely, [15] proposed to extend the compliance matrix to include *virtual objective functions* to evaluate E while solving constraints. The time saved allowed for performing an additional six perturbations to create a centered Jacobian matrix $\mathbf{J}$. However, the *objective functions* need to be linearized during the collision detection to be included in the $\mathbf{H}$ matrix, preventing the evaluation of non-linear functions such as computing angles to align the shaft of the needle with the tangent of the trajectory.

III. Method

In this section, we first introduce a set of non-linear objective functions used to stabilize needle steering, particularly in dynamic movements like respiratory motion. This involves rotating the robot's end effector to maintain the needle tip on the desired trajectory while ensuring the needle shaft passes through a remote center of motion (RCM), thus avoiding tearing at the entry point. We then introduce *isolated objective constraints* to evaluate the non-linear objective functions without reprojection on the degrees of freedom. Finally, we present our modified inverse loop, where the objective constraints are assembled only once for each computation of the independent columns of $\mathbf{J}$.

A. Objective functions

Each objective function represents a task to be performed during insertion. The following objective functions are used:

1) Trajectory objective: The first objective function aims to maintain the needle's tip on a predefined trajectory path, considered moving and deformable. This objective function is equivalent to the one proposed by [16] and [15]. Its expression is calculated as $\mathbf{e}_p = \mathbf{q}_{target} - \mathbf{q}_{tip}$.

2) Needle orientation objective: We define the needle orientation objectives in two parts (see Fig. 2): The needle's orientation θ_2 from the robot's end effector (EE) to the entry point, and the needle's orientation θ_1 from the needle tip to the tangent of the trajectory:

$$\begin{aligned} \theta_1 &= arccos(dot(\vec{o}, \vec{t})) \\ \theta_2 &= arccos(dot(\vec{e}, \vec{n})) \end{aligned} \tag{6}$$

Where $\vec{o}$ is the tangent of the trajectory at the target point, $\vec{t}$ is the direction of the needle tip, $\vec{n}$ is the needle's base direction and $\vec{e}$ is the direction of the vector from needle's base to the entry point. Keeping the two angles minimum decreases the needle bending, reducing stress on the organs.

3) Entry point objective: RCM mechanisms [20] are essential in needle steering problems by penetrating the patient's body through minimal, almost stationary surgical cuts. The entry point for the insertion is the beginning of the trajectory from the skin to the surface of the liver. To keep the point immobilized, for each time step, it is projected onto the needle using a 3-DOFs objective function (see figure 3), minimizing the displacement of the insertion point by finding the projection of the initial position of the point on the needle. Combining this objective function with the initial trajectory objective results in rotations of the robot to follow respiration and reduce stress on the tissue.

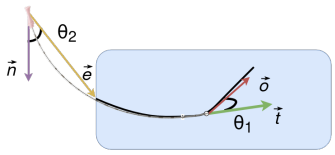


Fig. 2: Non-linear angular objectives functions.

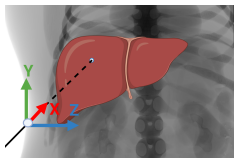


Fig. 3: RCM objective function.

Finally, our objective functions vector is defined as

$$\mathbf{T} = \begin{pmatrix} \mathbf{e}_x & \mathbf{e}_y & \mathbf{e}_z & \theta_1 & \theta_2 & \mathbf{e}_{\text{rcm}_x} & \mathbf{e}_{\text{rcm}_y} & \mathbf{e}_{\text{rcm}_z} \end{pmatrix}^T \tag{7}$$

This vector combines the displacement linear and angular non-linear values. In [16], the authors utilized the first five values within constraints, as determined by the compliance matrix size in equation (2). However, evaluating non-linear angular functions requires projecting the degrees of freedom (DOFs) space, which presents computational challenges. In contrast, [15] uses a subset of $\mathbf{T}$ based on the tip's location relative to organs, evaluating functions directly within the compliance matrix. This eliminates the need for movement corrections and reduces computation time but requires the linearization of non-linear functions. Despite its importance for stability during dynamic insertions involving respiration, linearizing functions introduce errors in needle positioning and tissue-related stresses.

B. Isolated objective-constraint space

To efficiently compute $\mathbf{J}$ while maintaining the nonlinearity of the objective functions, we introduce a new space called the *isolated constraint space*. This space contains only the degrees of freedom (DOFs) relevant to the objective functions, excluding any unnecessary DOFs to evaluate E. Thus, at each time step, we can update the values of these functions, especially the non-linear ones, by focusing solely on their positions within this isolated space.

We define the operator $\mathbf{H}_O = \hat{\mathbf{H}}_O\bar{\mathbf{I}}_O$ as the Jacobian matrix linking the space of constraints to the space of isolated objective constraints. $\bar{\mathbf{I}}_O$ is the partial identity matrix selecting the DOFs concerned by the constraint objectives. Its dimension defines the size of the *isolated constraint space*. When applying the constraint forces $\boldsymbol{\lambda}$ computed to enforce the mechanical constraints during simulation (such as those related to needle insertion or contacts between soft organs), we can efficiently integrate the positions $\mathbf{x}_{iso}$ of the relevant DOFs by solving:

$$\mathbf{x}_{iso} = \mathbf{x}^{free} + \mathbf{W}_O\boldsymbol{\lambda} * h \tag{8}$$

Where h represents the simulation's time step, and $\mathbf{W}_O = \sum \mathbf{H}_O \mathbf{A}^{-1} \mathbf{H}^T$ serves as an operator that establishes the mechanical coupling between the contact force $\boldsymbol{\lambda}$ of the simulation and the resulting displacement of the isolated constraints (refer to Fig. 4).

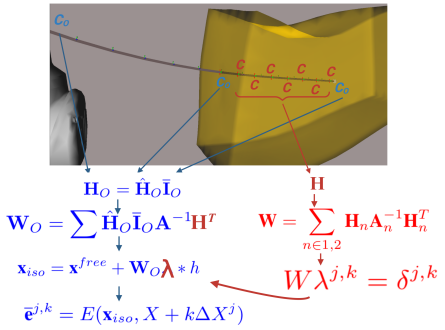


Fig. 4: Computation of the objective function in the isolated constraint space. Parts related to the mechanical computation are shown in red, and parts related to the computation of the isoconstraints are shown in blue.

By solving equation (8), the position of DOFs relevant to the isolated constraints can be directly evaluated without the need to project other DOFs. Thus, we define an isolated constraint on each DOF used to compute the objective functions described earlier. This primarily includes isolated constraints on the needle's base and tip and the projection of the RCM onto the needle shaft. It is important to note that isolated constraints must be placed on both models, i.e., the needle and the volume, as both models will undergo relative movements when applying the constraint force $\boldsymbol{\lambda}$.

C. Optimized inverse simulation loop

An essential advantage of our approach lies in the fact that the operator $\mathbf{H}_O$ and $\mathbf{W}_O$ can be computed using matrices and relations that depend only on the model's position at the beginning of the time step. Consequently, they can be computed only once to evaluate each column of the Jacobian $\mathbf{J}$ used for the robotic control. The simulation loop is then modified as follows in the **Algorithm 1**:

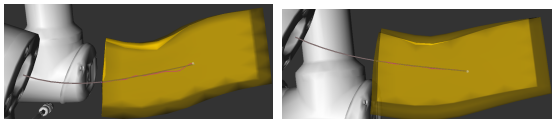
This optimized inverse simulation loop reflects the changes in lines 4, 10, and 11. The integration of positions $\mathbf{x}_{iso}$ is fast because $\hat{\mathbf{I}}_O$ consists of only a few lines. This results in the rapid re-evaluation of the objective values for each perturbation. Line 4 is the most time-consuming operation but is executed only once per simulation step. Moreover, these equations involve similar operations to those used for computing the compliance matrix $\mathbf{W}$, facilitating efficient computation on the GPU and thus saving significant processing time. As a result, the objective functions' errors converge fast, reducing the jacobian matrix's computation time $\mathbf{J}$.

Algorithm 1: Optimized Inverse simulation loop using iso-constraint

1 **Free Motion:** $\boldsymbol{x}^{free} = \mathbf{A}^{-1} b$;
2 **Constraint and Objective definition:** $\mathbf{H}$;
3 **Compute Compliance Matrix:** $\mathbf{W} = \sum_{n \in \{1,2\}} \mathbf{H}_n \mathbf{A}_n^{-1} \mathbf{H}_n^T$;
4 **Compute Isolated Objective Constraint Compliance Matrix:** $\mathbf{W}_O = \sum \mathbf{H}_O \mathbf{A}^{-1} \mathbf{H}^T$;
5 **Compute initial error:** $\bar{\mathbf{e}}_0 = E(\mathbf{q}, X)$;
6 **for** $j = 0$ *to* 6 **do**
7 **for** $k = -1, 1$ **do**
8 **Compute violation:** $\boldsymbol{\delta}^{j,k}$;
9 **Solve Constraint:** $\mathbf{W}\boldsymbol{\lambda}^{j,k} = \boldsymbol{\delta}^{j,k}$;
10 **Integrate positions in the iso-constraint space:** $\mathbf{x}_{iso} = \mathbf{x}^{free} + \mathbf{W}_O \boldsymbol{\lambda} * h$;
11 **Compute objective:** $\bar{\mathbf{e}}^{j,k} = E(\mathbf{x}_{iso}, X + k\Delta X^j)$;
12 **Compute Jacobian Line:** $\mathbf{J}^j = \frac{\bar{\mathbf{e}}^{j,1} - \bar{\mathbf{e}}^{j,-1}}{2\|\Delta X^j\|}$;
13 **Solve the inverse problem:** $\Delta X = -\mathbf{J}_\alpha^+ \cdot \bar{\mathbf{e}}$;
14 **Send the target pose:** $\mathbf{T}^{i+1} = X + \Delta X$;
15 **Compute violation:** $\boldsymbol{\delta}$;
16 **Constraints solving:** $\mathbf{W}\boldsymbol{\lambda} = \boldsymbol{\delta}$;
17 **Corrective motions:** $\mathbf{q} = \mathbf{x}^{free} - \mathbf{A}^{-1} \mathbf{H}^T \boldsymbol{\lambda}$;

IV. EXPERIMENTS

We assess the method using a curve trajectory (see Figure 5), as in [16], with a needle model comprising 13 edges. The model is parameterized with Young's modulus $E = 200$ GPa, Poisson ratio $v = 0.3$, and a radius of 0.723 mm. We maintain a fixed frame rate of 50 Hz with a time step of 0.02 ms. Each algorithm undergoes three scenarios with 50 identical insertions. Although they have the same trajectory, tumor position, and entry point, due to the respiratory movement, the average results from 50 trials should be computed to evaluate the method. The EFF receives position updates at each time step under a fixed Cartesian translation velocity of 50 mm/s and rotation of 0.2 rad/s. The target moves along the trajectory at 2.5 mm/s, taking around 30 seconds to reach the tumor. Collected data include the distance from the needle tip to the moving target, the angle between the tip direction and trajectory tangent, and the average stress value of all tetrahedra at each time step. Data collection starts when the needle aligns correctly with the trajectory and ends when it reaches the tumor. This constraint accurately evaluates each algorithm's ability to follow the target, aiming for most of the needle to lie on the trajectory.



(a) Insertion with method in [15] (b) Insertion with iso-constraint

Fig. 5: insertion following the target in a curve trajectory

In the second experiment, we utilize a human body as depicted in Figure 1, with organs reconstructed from a CT scan of the database *3D-IRCADb-02*[1], including the respiration cycle, to simulate a realistic scenario. The Jacobian matrix $\mathbf{J}$ is computed in the inverse simulation, encompassing the needle and the liver. Concurrently, we execute the direct simulation, wherein organ and

[1]https://www.ircad.fr/research/data-sets/respiratory-cycle-3d-ircadb-02

needle interactions occur, avoiding additional errors during registration. As the inverse simulation is the small replica of the direct simulation where the robot's commands are optimized and then sent back to the direct simulation, the whole environment of the direct simulation is the real reflection of how the needles interact with the organs. Consequently, the liver's state is transferred from the direct simulation to the inverse simulation. The Young modulus $E = 8.0$ kPa and the Poisson ratio $v = 0.4$ are selected. The tumor position in both simulations is the same as defined by the barycentric coordinate inside the liver. The displacements of the tumor in direct simulation resulted in identical results in the inverse simulation. In the future, where real-time imaging is involved, the tumors will be segmented before the operation, and the location will be tracked using the FE Model and fiducial marker implanted. The finite element (FE) model of the skin consists of a mesh of 1836 triangles, characterized by parameters $E = 20.0$ kPa and the Poisson ratio $v = 0.3$. This model facilitates simulating needle interactions for insertion into both the skin and liver. To assess precision, we vary the sizes of segmented liver models, specifically 498, 1000, and 1500 nodes, to observe how precision improves with increased model detail. We also conduct these evaluations using the method in [15]. The insertion time from the entry of the liver to the tumor is from 12.5 to 14.0 seconds.

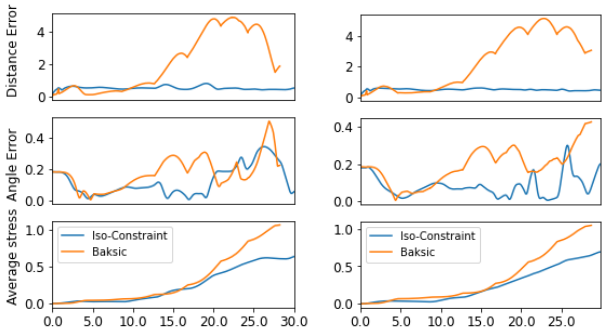
Due to the properties of the simulations, the experiments excluded the registration phase, which is required in the actual case. Nevertheless, in real insertion, the tissue motion will be measured based on the position displacement of the fiducial markers by the method in [21]. The ideal imaging for real-time tumor tracking would be fluoroscopy with C-Arm or ultrasound. The 2D positions of the markers on the image will be used to calculate the actual 3D positions, thanks to the extrinsic and intrinsic parameters.

V. RESULTS

In this section, the results from the experiments of two scenarios with two versions of the inverse loop (our and the methods in [15]) are compared regarding insertion precision, angular error of the needle tip direction to the tangent of the target, the stress applied to the gel caused by the insertion. Each result was obtained from 50 identical trials.

A. Distance, angular error and stress value on curve path

For a curve trajectory with a length of 7.5 cm, our method managed to force the tip to follow the trajectory target and have the error at the end at 0.5 mm after 30 seconds compared to 2.0 mm of the previous method (see figure 6). The angle error ranged from 0.08 to 0.13 rad compared to 0.18 using the method in [15]. The needle can maintain a close distance to the tip without deflecting. Therefore, the robot does not have to bend the needle to compensate for the distance error. This resulted in the end, and the needle caused only 0.6 MPa in stress compared to 1.0 MPa using the inverse loop in [15]



(a) Mesh size 500 (b) Mesh size 2000

Fig. 6: In the vertical axis, distance (mm), angular error (rad), and stress (MPa) value comparison with static curve trajectory. The time in seconds is reflected in the horizontal axis

B. Distance, average angular error under simulation with reconstructed human body

The results demonstrate an improvement in insertion accuracy, with enhancements ranging from 0.6 to 1.5 mm. The angle error improved by approximately 0.02 radians (see figure 7). This improvement can be attributed to the bilateral constraints on the skin model, which already restrict the movement of certain needle parts.

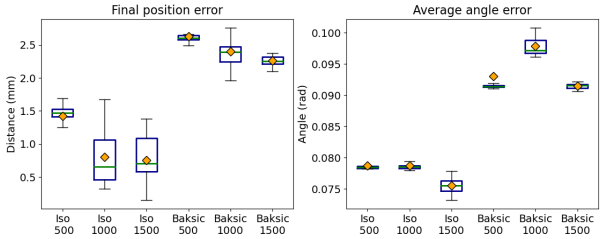
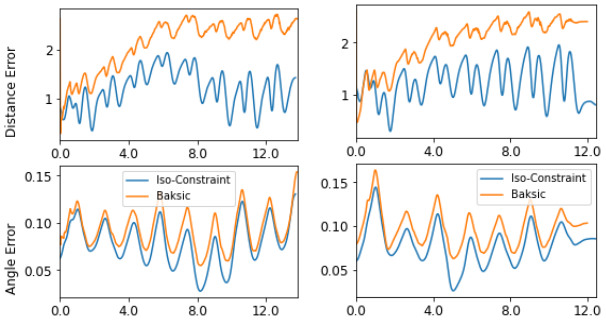


Fig. 7: Error comparison at the end of insertion for the straight trajectory inside the human body.



(a) Mesh size 500 (b) Mesh size 1000

Fig. 8: Under simulation inside human body simulation with respiration. The vertical axis represents distance (mm) and angular (rad) error value comparison. The time in seconds is reflected in the horizontal axis. .

Another observation from the experiments is the behavior observed in the line graph. When both θ_1 and θ_2 are included in the objective functions, the robot tends to wait for the alignment of the needle base direction with the tip's direction before pushing instead of attempting to bend. Therefore, the plot in figure 8 shows a high amplitude in the distance error

curve. However, the average error is significantly lower at 1.0mm than 2.25 mm using [15]. This behavior ultimately leads to reaching the tumor with a lower distance error.

C. Computational time

The computation time for two versions of the inverse simulations is calculated at the end of the insertion, where the maximum number of constraints is achieved. The constraint distance is 10 mm, and the mesh sizes used for testing are 500, 1000, 2000 vertices. Performance was measured using a computer with an AMD Ryzen 9 5900X processor and a RTX 3070 Ti graphics card. The experiments are processed under Assist Sofware, SofaFramework [22] and ROS2 and Ubuntu 22.04. While the percentage of the inverse loop in our method is around 3-4% higher, the total computational time for a simulation step is lower than that of the method in [15]. We use the GPU-based solver proposed in [19] to assemble $\mathbf{W_O}$. In table I, mechanical matrix definition and inverse loop have more proportion of time in one-time step due to the changes in lines 4 and 10 of the algorithm 1.

	Simulation step (%)						Computation Time (ms)	
	Iso-constraint			Baksic				
Mesh Size	MM	Inv	FE	MM	Inv	FE	Iso-constraint	Baksic
500	74.16	15.92	9.92	62.34	12.51	25.15	4.88	5.57
1000	78.84	10.1	11.06	70.49	5.76	23.75	9.09	10.64
2000	82.24	4.54	14.91	76.67	3.1	20.23	21.07	24.43

TABLE I: Computation time comparison with various mesh sizes. MM: Mechanical matrices definition (lines 1-4). Inv: Inverse Loop (Lines 6-12). FE: FE Resolution (lines 15-17)

VI. CONCLUSIONS

A strategy to fast compute the positions related to the objective function and the optimized inverse simulation loop have been introduced to steer a needle in deformable and moving tissues. This numerical method allows us to evaluate the non-linear objective functions under complex FE simulations, such as the angular, and precisely compute the linear error while maintaining the system's stability. The method was evaluated in a realistic framework generated from human CT scan data. The proposed solution allows steering the needle in real-time and fulfills recommendations for RFA procedures. Future work will apply the method to steer the needle with the robotic system.

REFERENCES

[1] D. A. McDermott, Shaunagh; Gervais, "Radiofrequency ablation of liver tumors," *Semin intervent Radiol*, vol. 30, no. 01, pp. 049–055, Feb 2013.

[2] A. Okamura, C. Simone, and M. O'Leary, "Force modeling for needle insertion into soft tissue," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 10, pp. 1707–1716, 2004.

[3] M. M. Arnolli, M. Buijze, M. Franken, K. P. de Jong, D. M. Brouwer, and I. A. Broeders, "System for ct-guided needle placement in the thorax and abdomen: A design for clinical acceptability, applicability and usability," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, 2 2018.

[4] S. DiMaio and S. Salcudean, "Needle steering and motion planning in soft tissues," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 6, pp. 965–974, 2005.

[5] D. Stoianovici, C. Jun, S. Lim, P. Li, D. Petrisor, S. Fricke, K. Sharma, and K. Cleary, "Multi-imager compatible, mr safe, remote center of motion needle-guide robot," *IEEE Transactions on Biomedical Engineering*, vol. 65, pp. 165–177, 1 2018.

[6] T. Hiraki, T. Kamegawa, T. Matsuno, T. Komaki, J. Sakurai, and S. Kanazawa, "Zerobot®: A remote-controlled robot for needle insertion in ct-guided interventional radiology developed at okayama university." *Acta medica Okayama*, vol. 72 6, pp. 539–546, 2018.

[7] A. Krupa, "3d steering of a flexible needle by visual servoing," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, Eds. Cham: Springer International Publishing, 2014, pp. 480–487.

[8] B. Maurin, J. Gangloff, B. Bayle, M. de Mathelin, O. Piccin, P. Zanne, C. Doignon, and L. Soler, "A parallel robotic system with force sensors for percutaneous procedures under ct-guidance," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*, France, Saint-Malo, Oct 2004.

[9] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[10] A. Krupa, "A new duty-cycling approach for 3d needle steering allowing the use of the classical visual servoing framework for targeting tasks," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2014, pp. 301–307.

[11] J. Chevrie, N. Shahriari, M. Babel, A. Krupa, and S. Misra, "Flexible needle steering in moving biological tissue with motion compensation using ultrasound and force feedback," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2338–2345, 2018.

[12] Y. Kobayashi, T. Sato, and M. G. Fujie, "Modeling of friction force based on relative velocity between liver tissue and needle for needle insertion simulation," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 5274–5278.

[13] Y. Lei, S. Du, M. Li, T. Xu, Y. Hu, and Z. Wang, "Needle-tissue interaction model based needle path planning method," *Computer Methods and Programs in Biomedicine*, vol. 243, p. 107858, 2024.

[14] 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, pp. 542–553, 2013.

[15] P. Baksic, H. Courtecuisse, C. Duriez, and B. Bayle, "Robotic needle insertion in moving soft tissues using constraint-based inverse finite element simulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2407–2413.

[16] Y. Adagolodjo, L. Goffin, M. de Mathelin, and H. Courtecuisse, "Robotic Insertion of Flexible Needle in Deformable Structures Using Inverse Finite-Element Simulation," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 697–708, Mar. 2019.

[17] C. Felippa, "A systematic approach to the element-independent corotational dynamics of finite elements," 01 2000.

[18] H. Courtecuisse, J. Allard, P. Kerfriden, S. P.-A. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical Image Analysis*, vol. 18, no. 2, pp. 394–410, Feb. 2014.

[19] Z. Zeng, S. Cotin, and H. Courtecuisse, "Real-time fe simulation for large-scale problems using precondition-based contact resolution and isolated dofs constraints," *Computer Graphics Forum*, vol. 41, no. 6, pp. 418–434, 2022.

[20] J. Li, Y. Xing, K. Liang, and S. Wang, "Kinematic Design of a Novel Spatial Remote Center-of-Motion Mechanism for Minimally Invasive Surgical Robot," *Journal of Medical Devices*, vol. 9, no. 1, p. 011003, 03 2015.

[21] Y. Adagolodjo, N. Golse, E. Vibert, M. de Mathelin, S. Cotin, and H. Courtecuisse, "Marker-based Registration for Large Deformations -Application to Open Liver Surgery," in *ICRA 2018 - International Conference on Robotics and Automation*, Brisbane, Australia, May 2018.

[22] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A Multi-Model Framework for Interactive Physical Simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, ser. Studies in Mechanobiology, Tissue Engineering and Biomaterials, Y. Payan, Ed. Springer, June 2012, vol. 11, pp. 283–321.