2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Real-time Robotic Needle Insertion In Deformable and Moving Structure Using Learning-by-Example Method

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

***Abstract*— This paper presents an innovative and practical method for robotic needle steering in radio-frequency ablation (RFA) to treat cancer. One of the main challenges in this process is that tissue shifts and deforms during needle insertion, making it difficult to accurately predict the needle's path in real time. Inverse finite element (iFE) simulations have been used to address this problem. While these methods are accurate, they often require further refinement for effective time performance in real-world robotic systems. This is because when the method is incorporated into a real robot, there can be a delay in command execution. To address this challenge, we propose a machine learning-based solution that learns from offline simulations, shifting the intensive calculations required by iFE methods to an offline training stage and enabling online prediction of tissue deformation with reduced computational time. Our network was trained on data from numerous simulated needle insertions to capture interactions among insertion forces, tissue properties, and resulting motion. Once trained, the model produces predictions almost instantaneously, making it suitable for real-time applications. We validated the approach by steering the needle in a simulated deformable, moving gel to compare it with numerical-based methods, and then performing needle steering within a reconstructed human body that involves multiple structures and integrates the robot's dynamics. The results demonstrated that the developed networks achieved slightly better accuracy in the first scenario while also running faster, resulting in improved performance under the robot's dynamics. These findings show that our method is a promising advancement toward real-time guidance systems for needle-based medical procedures.**

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

I. INTRODUCTION

Accurate needle steering is essential in many medical procedures, such as biopsies and minimally invasive surgeries. Traditional methods depend on the practitioner's skill, but manual control and rigid needles have inherent limitations that can make it challenging to precisely reach internal targets without damaging surrounding tissue.

A robotic solution to support needle placement in percutaneous interventions was introduced in [1] and has been certified for clinical usability and acceptance. With robotic arms, active needle steering becomes possible, enabling flexible needles to follow curved, complex paths [2]. Despite many robotic solutions for percutaneous procedures [3], operating within dynamic, deformable organs remains a challenging task. Accurate, real-time estimation of tissue deformation is crucial to prevent adverse outcomes, underscoring the need for robust robotic methods that can adapt on the fly [4].

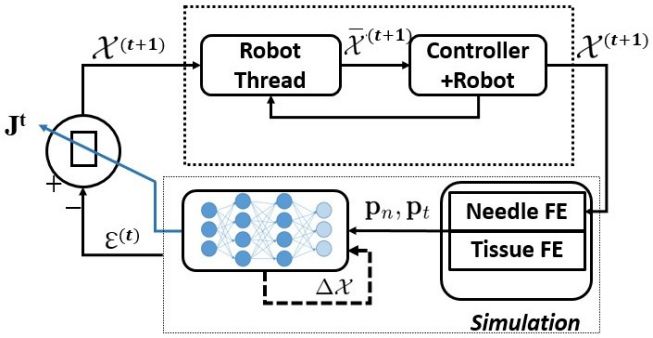


Fig. 1: Control Loop using Machine Learning

A common strategy involves solving the inverse kinematics problem and controlling a manipulator's motion using neural networks [5]. Additionally, machine learning models can predict optimal needle behavior and tissue deformation based on patient-specific data, thereby improving precision and reducing tissue damage. The recurrent neural network (RNN) approach was used by [6] to accurately estimate the roll angle of steerable needles during steering in various media, including tissue, ovine brain, and porcine lung. The proposed Long-Short-Term Memory (LSTM), based on an RNN, is designed to estimate the unsensed roll angle of the needle tip using partial-state sensing inputs. This method outperformed traditional model-based methods [7], [8], particularly in scenarios with significant modeling errors and complex system dynamics, such as those involving flexible needles with torsional compliance. It demonstrated highly accurate needle steering in both controlled gelatin experiments and in a real-tissue environment.

Data-driven methodologies, such as extreme learning machines (ELMs) [9], Gaussian mixture regression (GMR) [10], and K-nearest neighbors regression (KNNR), have been used to capture the complex, nonlinear inverse kinematics of TSM. Work by [5] on the control of an endoscopic device relied solely on a data-driven approach to learn a direct mapping from the task space to the actuator space.

The challenge of percutaneous surgery in treating liver tumors lies in needle tip deflection, which requires precise control. [11] introduced an ultrasound-guided puncture robot using a data-driven needle deflection correction method with a kernel extreme learning machine (KELM). Results showed an enhancement in accuracy. The study by [11] aimed to

*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

address limitations in current needle steering models by proposing a real-time correction approach.

Due to the limitations of numerical methods, such as computational time and convergence of the inverse problem, artificial neural networks (ANNs) are considered a promising alternative to conventional methods due to their flexibility, ability to capture nonlinear relationships, and strong learning capacity. Their fast computation makes them well-suited for real-time applications, such as needle steering, where timing is critical.

Residual MLP (RMLP) [12] is a neural network architecture for computer vision that relies entirely on multi-layer perceptrons (MLPs). The network consists of residual MLP blocks. Each block has two main parts: one linear layer that lets information flow across different image patches, and another that mixes information within each patch across channels. RMLP uses a lightweight learned affine transformation. Combined with residual skip connections, this design makes the network stable to train even at large scales. Our research explores the use of RMLP to approximate solutions to the inverse problem. Preliminary results indicate that neural networks can learn the mapping between the system's state and the control outputs derived from numerical models. While still exploratory, this work opens the perspective of replacing the computationally demanding model-based control step with data-driven prediction in future developments. The clinical motivation is to optimize our neural network for patient-based control by simulating numerous insertions with different trajectories. One possible extension of our methods is the adaptation to real-time medical image registration, which addresses the limitations of the fiducial marker method in [13]. So in that way, this article proposes:

- A method to reduce the computation time of the iFE method while ensuring the resolution of the non-linear constraints of the problem. Particularly for needle steering into a moving, deformable structure, by understanding the behavior of the iFE method, we can utilize a nonlinear objective function that can be computed in real time to derive the subsequent needle displacement, thereby steering it with high precision within the deformable structure.
- A data collection method based on iFE simulation, whose purpose is to cover the different scenarios that cause the non-linear interaction between the needle and the deformable structure.
- Two Neural Networks which can successfully solve the needle steering inverse problem, thereby controlling needle movements during insertion along a predefined trajectory.

The remainder of this paper is organized as follows: Section II provides a background on the needle steering in the finite element method. Section III outlines the primary network architectures, dataset generation methods, and network input schemes employed in this work. In Section IV, we present the experiments and analyze them. Finally, in Section V, we conclude this paper with insights on current limitations and future work directions.

II. BACKGROUNDS

Since the the needle and organ model have different degrees of freedom (DOFs), their interaction, encompassing both deformable body behavior and displacement constraints, is addressed using Lagrange multipliers [14]. The additional constraints introduced into the system must be enforced at the end of each time step to keep the needle aligned with its intended trajectory and to capture critical effects such as friction between the structures during insertion. When multiple deformable bodies interact, the mechanical formulation leads to a Karush-Kuhn-Tucker (KKT) system:

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

Here, $\mathbf{A}_i$ is the system matrix of object i, obtained from the sum of stiffness and mass matrices and updated at each time step; $\mathbf{x}_i$ denotes the nodal displacements; $\mathbf{H}_i$ is the Jacobian mapping constraints to motion; $\boldsymbol{\lambda}$ are the Lagrange multipliers; and $\boldsymbol{\delta}$ represents the constraint violations [15]. The solution starts by computing the *free motion*, i.e., the dynamics without constraints:

$$\mathcal{X}_i^{\text{free}} = \mathbf{A}_i^{-1} b_i, \quad \text{for } i = 1, 2 \tag{2}$$

Then, with $\mathbf{x}_i^{free}$, the system of constraints becomes:

$$\begin{aligned} \mathbf{W}\boldsymbol{\lambda} &= \boldsymbol{\delta}_{free} - \boldsymbol{\delta}, \\ \text{with} \quad \mathbf{W} &= \sum_{n\in\{1,2\}} \mathbf{H}_n \mathbf{A}_n^{-1} \mathbf{H}_n^T, \\ \text{and} \quad \boldsymbol{\delta}_{free} &= \sum_{n\in\{1,2\}} \mathbf{H}_n \mathbf{x}_n^{free} \end{aligned} \tag{3}$$

The matrix $\mathbf{W}$, referred to as the Delassus operator, represents the mechanical coupling between models in the constraint space. While its computation is costly, substantial acceleration can be achieved through GPU-based parallelization [16].

To solve Equation (3), a projected Gauss–Seidel method is employed to obtain $\boldsymbol{\lambda}$, which defines the force enforcing the relative displacement between the needle and the tissue model along the trajectory. Once $\boldsymbol{\lambda}$ is determined, the free motion is corrected to satisfy the constraints using

$$\mathbf{x} = \mathbf{x}_i^{free} + \mathbf{A}i^{-1}\mathbf{H}1^T\boldsymbol{\lambda} \tag{4}$$

The inverse Finite Element Method (iFEM) was first introduced in [17], [18] to enable needle steering in both static and deformable structures. To accelerate the minimization of objective values in moving organs, [19] extended the method by introducing objective constraints that linearize quantities such as angular deviations. However, this approach comes at the cost of reduced insertion accuracy. More recently, [20] proposed the isolated-constraints iFEM, which resolves nonlinear objective values in real time to improve precision.

The robotic needle insertion task is defined as a problem of non-linear function minimization [18], [19], [20]:

$$\mathcal{E}(\mathcal{X}^t, \mathbf{p}_n, \mathbf{p}_t, \mathbf{m}) = 0 \tag{5}$$

Here, $\mathcal{X}^t$ is the end-effector position at time t, p_n the needle tip position, and p_t the target position, determined by the liver model configuration. The objective function $\mathcal{E}(\mathcal{X}^t, \mathbf{p}_n, \mathbf{p}_t, \mathbf{m})$ is minimized to align the needle with the target, where $\mathbf{m}$ denotes the fiducial marker positions. The cost function E integrates three objectives:

- **Trajectory:** Minimizes the distance between the needle tip $\mathbf{q}_{tip}$ and the moving target $\mathbf{q}_{target}$ along a predefined path adapting to organ motion,

$$\mathbf{e}_p = \mathbf{q}_{target} - \mathbf{q}_{tip}. \tag{6}$$

- **Orientation:** Two angular constraints enforce alignment: the tip with the trajectory tangent, and the base (end-effector) with the entry-point vector,

$$\begin{aligned} \theta_1 &= \arccos(\vec{o} \cdot \vec{t}), \\ \theta_2 &= \arccos(\vec{e} \cdot \vec{n}), \end{aligned} \tag{7}$$

  where $\vec{o}$ is the trajectory tangent, $\vec{t}$ the tip direction, $\vec{n}$ the base direction, and $\vec{e}$ the entry-point vector. Minimizing θ_1 and θ_2 reduces bending and tissue stress.
- **Entry point:** A Remote Center of Motion (RCM) constraint keeps the needle entry fixed by projecting the initial point onto the shaft at each step.

The objective functions are solved at the same time in a multi-objective optimization problem using an iFE method. Because of the needle-tissue interactions' non-linear nature, the iFE method [17] is employed to compute the robot's EEF pose. At each time step, this method solves a minimization problem. A first-order Taylor expansion is used:

$$\mathcal{E}(\mathcal{X}^t + \Delta\mathcal{X}, \mathbf{p}_n, \mathbf{p}_t) \approx \mathcal{E}(\mathcal{X}^t, \mathbf{p}_n, \mathbf{p}_t) + \frac{\delta\mathcal{E}}{\delta\mathcal{X}} d\mathcal{X} = 0 \tag{8}$$

The optimal displacement $\Delta\mathcal{X}$ of the robot's end-effector is computed by:

$$\Delta\mathcal{X} = \mathbf{J}^{-1}\mathcal{E}(\mathcal{X}^t, \mathbf{p}_n, \mathbf{p}_t) \tag{9}$$

$\mathbf{J}$ denotes the Jacobian matrix of the simulation, corresponding to the gradient of the simulation. It is essential for this gradient optimizing the cost function $\mathcal{E}x$. However, because of the complex, nonlinear constraints which might not be continuous (the needle-tissue interactions), it is not possible to derive an analytical formulation of the Jacobian. Instead, we can compute a numerical approximation as:

$$\mathbf{J}^i = \frac{\mathcal{E}(\mathcal{X}^t + \Delta\mathcal{X}_i, \mathbf{p}_n, \mathbf{p}_t) - \mathcal{E}(\mathcal{X}^t - \Delta\mathcal{X}_i, \mathbf{p}_n, \mathbf{p}_t)}{2\|\Delta\mathcal{X}_i\|} \tag{10}$$

where $\Delta\mathcal{X}_i$ is a perturbation along the Cartesian axis i of the end effector (EEF) of the robot.

Since evaluating each column of $\mathbf{J}$ requires solving a full FE simulation step with various initial conditions , [18] proposed solving the inverse problem directly in the constraint space using the projection $\mathbf{A^{-1}H^T\lambda}$. Alternatively,

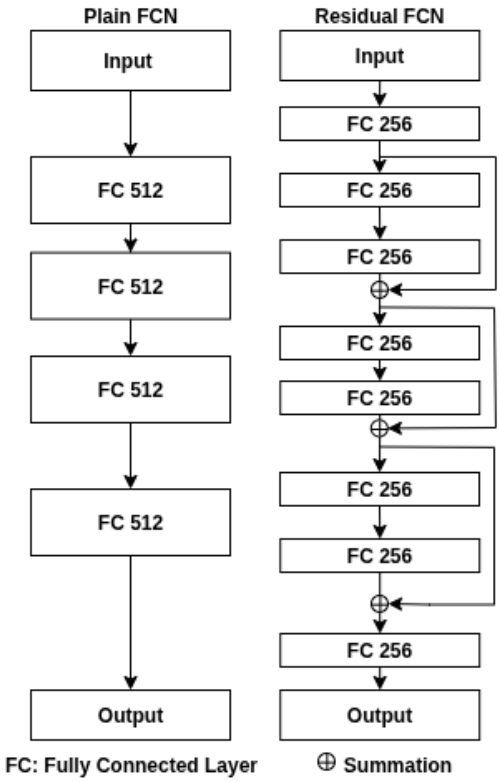


Fig. 2: Neural Networks investigated as inverse models; left: Plain Fully Connected Neural Network (FCN) for deformable gel insertion, middle: Residual-like FCN (RFCN) for needle insertion inside reconstructed human body

[19] extended the compliance matrix to incorporate *virtual objective functions*, enabling efficient evaluation of $\mathcal{E}$ while resolving constraints.

Each column of $\mathbf{J}$ is derived from FE simulations with varying robot positions, capturing local tissue–needle displacements and providing the gradient for control optimization.

Although the validity of the Jacobian is limited, on one hand by the mechanical parameters of the model, which are unknown, and on the other hand by the boundary conditions imposed by the registration constraints, it can still be used to determine a search direction to minimize errors in the objective function of equation (5) and derive a control command.

III. METHODS

A. Dataset Generation

The data for training and testing are generated from the inverse Finite Element (iFE) simulation [20] to minimize the objective function for the robot's end-effector position, thereby commanding the robot's pose. We aim to steer the needle to align the needle tip at a specific position in space, aligned with another point.

The data for inputs is the tuple of 14 values, containing

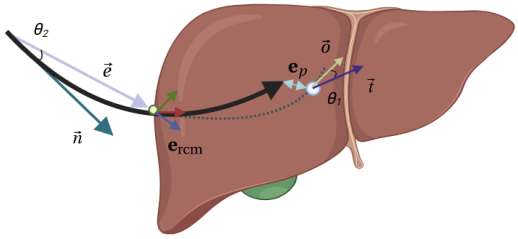


Fig. 3: Objective functions for needle steering inside liver's problem

the Robot's EEF pose:

$$\mathcal{X}_{pose}(x, y, z, r_x, r_y, r_z) \tag{11}$$

And the objective values which are illustrated in Fig. 3

$$\mathcal{T} = \begin{pmatrix} \mathbf{e}_x & \mathbf{e}_y & \mathbf{e}_z & \theta_1 & \theta_2 & \mathbf{e}_{\text{rcm}_x} & \mathbf{e}_{\text{rcm}_y} & \mathbf{e}_{\text{rcm}_z} \end{pmatrix} \tag{12}$$

The output is the robot's base displacement.

$$\Delta\mathcal{X}(\delta x, \delta y, \delta z, \delta r_x, \delta r_y, \delta r_z) \tag{13}$$

These values have been proven to be effective in steering the needle inside deformable objects and reconstructing the human body [20].

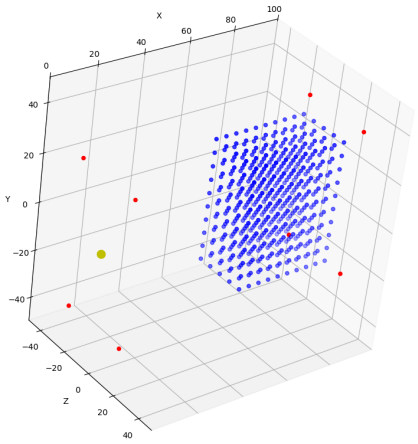


Fig. 4: Distribution of target points for data collection. Blue dot: equally distributed target points for each trajectory. Red dot: corners of the gel. Yellow dot: entry point of the gel

1) Needle Insertion into Deformable Gel: The first type of dataset is the need for insertion into a deformable gel. We define 500 target points in the right-most part of the gel and a sole insertion point (Fig. 4). The needle's starting pose is placed in two ways. The first one is perpendicular to the gel, with the needle tip at the entry point of the gel. Therefore, the needle direction is normal to the gel surface. The aim is to create the bending and non-linearity needed for the network to learn when it creates the bending to steer the tip to reach the target. This bending creates the non-linearity of the problem and makes it costly to predict. The second pose of the needle is that the needle lies along the line containing the trajectory, with the tip at the entry point. This aims to produce the data for straight insertion.

2) Needle insertion into simulated human model: The second type of dataset is the need for insertion into a reconstructed human model, which comprises the liver and the part from skin to before the liver (Fig. 5). We define 500 pseudo-tumor points around the intended target tumor. The trajectory is divided into two parts. The first one is from an insertion point to the surface of the liver. The second one is from the surface of the liver to the tumor. The data captured during the insertion process and under the respiration simulation, which was caused by the diaphragm from direct simulation. The purpose is to capture as much of the nonlinear behavior of the insertion.

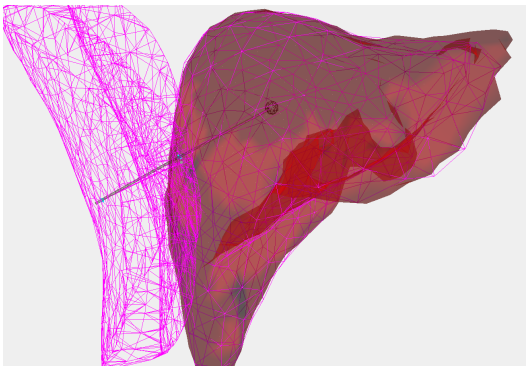


Fig. 5: Inverse Simulation for data collection, containing only the needle, liver, and skin's FEM (pink mesh)

B. Neural network as control loop

The Jacobian computation in equation 10 requires 12 simulations to be performed in one step. This is costly, and when integrating with a real robot, this could cause a delay. Therefore, there is a need to transfer the computing to offline learning for a needle steering inverse model, so that after receiving information from tissue-needle interaction, the subsequent displacement of the EEF can be derived in a shorter time.

We seek to address the inverse modeling of needle insertion by using a fully connected neural network (FCN) to steer a needle along a precisely defined trajectory. We evaluate the effectiveness of the neural networks in controlling robotic movements within a highly realistic biomechanical simulation. In this setting, the plain FCN inverse model is utilized for deformable gel insertion, and the Residual FCN [12] is used for insertion into the human body. These networks (Figure 2) help guide robot motion with higher precision in simulations but lower computational time.

Our simulation environment integrates the neural network with a forward model grounded in Finite Element Method (FEM) techniques, ensuring both physical realism and accuracy. This forward simulation is crucial for extracting the relevant variables that serve as inputs for our network. By

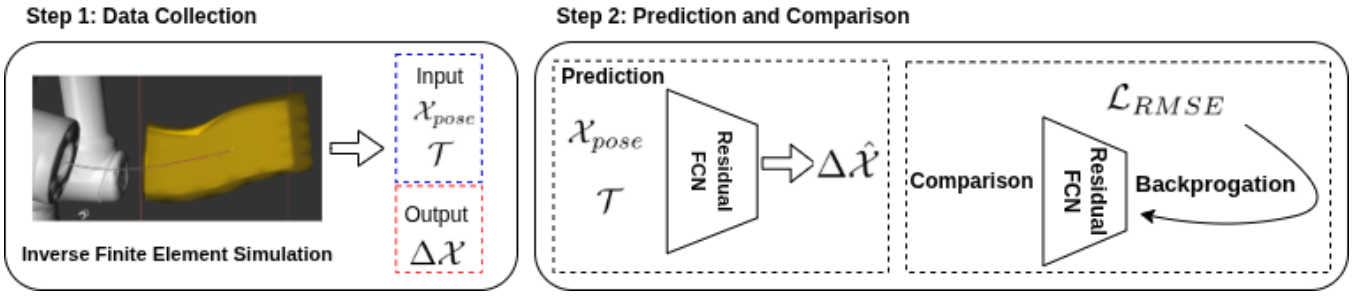


Fig. 6: Schematic of the different learning steps involved in learning by example framework on the needle insertion into deformable structure. Step 1: Data collection from different configuration of simulation to get the needle's pose, objective function's values and EEF's displacement to minimize these function. Step 2: Predict the next displacement and perform comparison with the values in the dataset

doing so, our neural network effectively solves the inverse problem, determining the necessary end-effector position to reach the next target point along the trajectory with the needle tip.

C. Loss function & Data Normalization

The networks were trained by minimizing the Root Mean Squared Error (RMSE) loss denoted here by $\mathcal{L}_{RMSE}$

$$\mathcal{L}_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\Delta\mathcal{X} - \Delta\hat{\mathcal{X}}\right)^2}. \tag{14}$$

Where $\Delta\mathcal{X}$ and $\Delta\hat{\mathcal{X}}$ are respectively the ground truth and predicted displacement of the DOFs. Moreover, N is the total number of samples in the training batch.

Also, before training, as shown in Fig. 6, both the input and output from the datasets are normalized using the StandardScaler from the Scikit-Learn [21] library. The purpose is to rescale each feature from input or output to have a zero mean and unit variance. This helps the training converge more quickly and compare features accurately.

D. Inverse models to compare & Evaluation metrics

We compare the results of insertion into a deformable gel using our method with those in [20] and [19] under three scenarios: static gel with a straight trajectory, static gel with a curved trajectory, and moving gel with a straight trajectory.

As performance metrics, the tip to target distance (eq. 6) and the average error of orientation (eq. 7) are considered along the needle insertion.

IV. EXPERIMENTS AND RESULTS

A. Learning Result

A small MLP of 3 layers with 512 weights is sufficient for learning the necessary steering behavior within the gel. For the human body, we have used the Residual FCN of 3 residual blocks with 2 FC layers of 256 weights each. To evaluate the learning process, the average loss is computed on the test set of randomly generated configurations. The data obtained consists of 1,000,000 samples for each scenario, divided into 75% for training, 15% for testing, and 15% for the validation dataset.

TABLE I: Training results: initial vs. final loss and best epoch.

Scenario	Initial Loss	Final Loss	Best Epoch
Deformable gel	0.45	0.21	72
Human liver	0.20	0.07	55

B. Needle Insertion into Deformable Gel

(a) Static curved trajectory

(b) Moving straight trajectory

Fig. 7: Needle insertion into deformable gel under two trajectory conditions.

We define three scenarios to test the insertion into the simulated deformable gel: a static straight trajectory, a static curve trajectory, and a moving straight trajectory (Fig. 7). The targets are 200 randomly distributed points, each with a 10cm distance from the entry points, ensuring the same insertion length in both straight and curved trajectories. The results in Fig. 8 show the improvement of precision and non-linear errors from the methods in [22] and [20]

The needle's model is composed of 13 nodes, parameterized with Young's modulus E = 200 GPa, a radius of 0.723 mm and Poisson ratio v = 0.3. We keep a 50 Hz fixed frame rate with a time step of 0.02 ms. Each type of scenario has 200 different insertions, each with a different trajectory for different target points. These trajectories are not presented in the dataset. At each time step, the robot's EFF receives position updates with a fixed Cartesian translation velocity of 0.05 cm/s and rotation of 0.2 rad/s. The virtual target to guide needle's tip slides along the trajectory at 2 mm/s.

For a straight trajectory, our method has a slightly better error than [20] but achieves the same error in the end: 0.12

mm, which is far better than [19], which is 0.24mm—the same result for the angle evaluation. For curve trajectory, our method and [20] converge better than [19] and have around 0.4 - 0.6mm for distance error For moving gel, our method is more stable with keeping distance from the trajectory's target than [20] while has the same error at then end, and performing better than [19]. As show in Fig. 9, our method give the nearly the same targeting error at the end of trajectory and slight better angle error than [20] which is intended, and the computational time is show in next part.

The dataset generated from synthetic simulation under static insertion with intended bending to create non-linear condition. The results show that even with curve trajectory and moving gel, the robot successfully learned from the data coming from numeric method and perform the insertion in dynamic environment with cause greater bending of the needle and have higher degree of non-linearity.

C. Computational time

The computation time of two inverse simulation variants is compared to the FCN approach at the end of insertion, when the number of constraints is at its maximum. Tests are performed with a constraint distance of 10 mm and mesh sizes of 1000. Experiments were run on a workstation equipped with an AMD Ryzen 9 5900X CPU and an RTX 3070 Ti GPU, using Ubuntu 22.04, SofaFramework [23], and ROS2. Our method has a significantly lower computational time for the inverse step, 4.06%, and a lower overall computational time, as shown in Table II.

TABLE II: Computation time comparison with various methods. MM: Mechanical matrices definition, Inv: Inverse loop, FE: FE resolution.

Step \ Metric	Computation (%)	Time (ms)
Iso-constraint: MM	78.84	7.167
Iso-constraint: Inv	10.10	0.918
Iso-constraint: FE	11.06	1.005
Obj-constraint: MM	70.49	7.501
Obj-constraint: Inv	5.76	0.612
Obj-constraint: FE	23.75	2.525
MLP: MM	78.41	5.036
MLP: Inv	4.06	0.260
MLP: FE	17.53	1.562

D. Needle Insertion into reconstructed human model

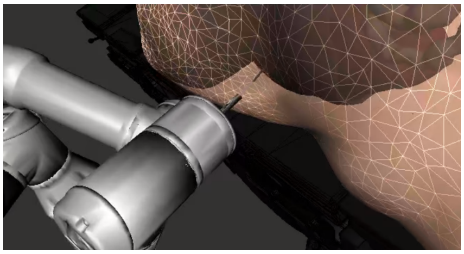


Fig. 10: Direct Simulation where reconstructed human's body with respiration is used to validate the needle insertion

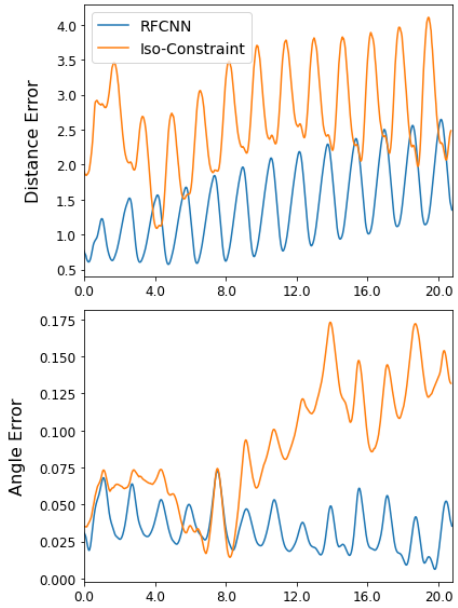


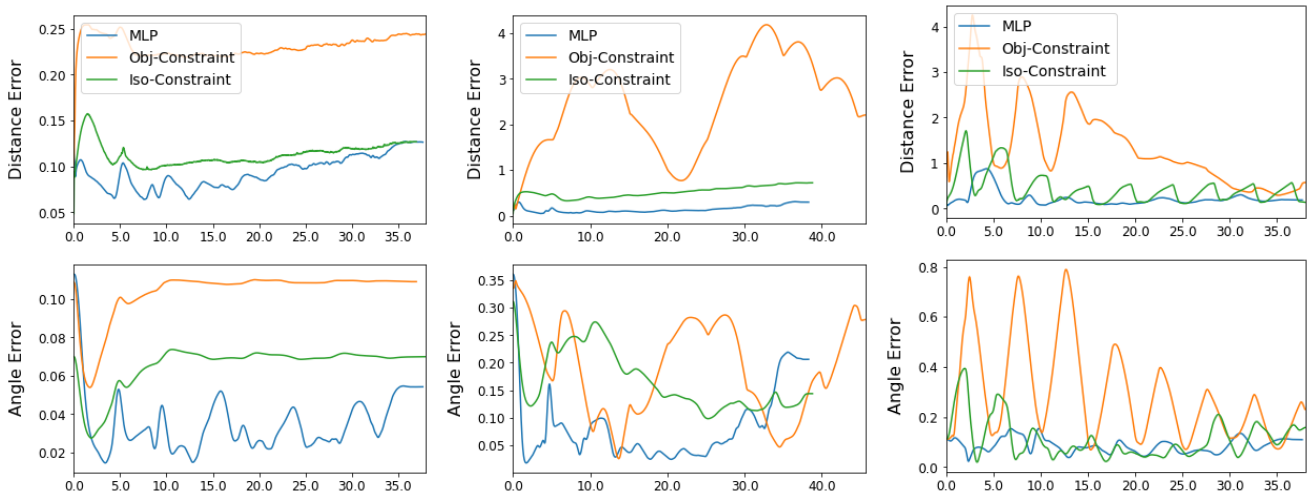
Fig. 11: Needle Insertion results inside human body simulation with respiration and robot's dynamics. The vertical axis represents distance (mm) and angular (rad) error value comparison. The time, in seconds, is reflected on the horizontal axis. (RFCNN: our RFCN. Iso-Constraint: The method in [20])

Now with the high precision obtained from the gel's insertion and the reduced computational time for both overall and specific inverse step a, we used the trained Residual FCN model (Fig. 2) to perform insertion into a reconstructed human body from a CT scan of the database *3D-IRCADb-02*[1], including the respiration cycle, to simulate a realistic scenario.

The validation in [19], [17], [20] use a simulation called Direct Simulation (DS) where the control is calculated and sent from the Inverse Simulation. The DS reflects a realistic environment, where the whole body's organs, as well as the robot's arm and imaging devices, are visible (Fig. 10). In this scenario, we assume that the needle-tissue interaction (Fig. 1) from Section II can be deduced from a registration method using a real-time imaging device. Moreover, the needle's state and trajectory could be derived from the registration. Our Residual FCN, therefore, takes this information along with the robot's pose, which is also calculated from forward kinematics or from visual tracking. This means we can opt out of the Inverse Simulation and steer the needle directly, in this case, Direct Simulation. This helps save further computation time needed.

In DS, the liver model consists of 498 triangles. The Poisson ratio $v = 0.4$ and the Young modulus $E = 8.0$ kPa are the parameters for the model. The skin's model is a mesh of 1836 triangles, with parameters chosen as $E = 20.0$ kPa and the Poisson ratio $v = 0.3$. The reason for this choice is that the Residual FCN performs better with two

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



(a) Straight Trajectory (b) Curve Trajectory (c) Straight Trajectory Moving Gel

Fig. 8: At the vertical axis, comparison of distance (mm), angular error (rad) value between methods for static straight, static curve, and moving straight trajectories. The time, in seconds, is reflected on the horizontal axis. (MLP: our FCN. Obj-Constraint: The method in [19]. Iso-Constraint: The method in [20])

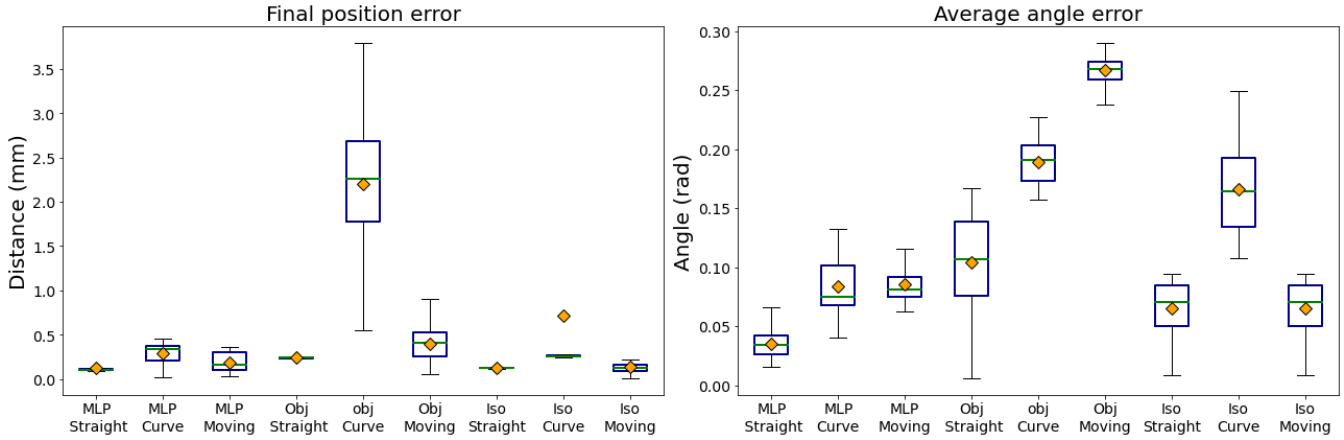


Fig. 9: Error comparison at the end of insertion for the different trajectories into the gel. (MLP: our FCN. Obj-Constraint: The method in [19]. Iso-Constraint: The method in [20])

different structures. Additionally, we need to consider the dynamics of the UR3e robot, including delays and EEF pose errors. As the inverse simulation loop produces the required Cartesian coordinate in the simulation's coordinates, there is a requirement to convert it to the robot's coordinates. Once the Cartesian pose is calculated, the simulation is sent to the robot in Docker's environment via ROS2 [24] to perform insertion using the Cartesian controller developed by [25]. This could evaluate how the delay from the robot's controller affected the insertion precision. The result in Fig. 11 shows that our method has better error along the insertion and at the end, achieve the precision lower 1mm than [20]. This could be explained by the less computation time for the inverse step lead to the robot received the cartesian pose command faster to cope up with the respiration cycle.

V. LIMITATIONS

Based on the simulation, several limitations in our paper remain unaddressed. As the trajectory is based on a straight line and assumes the liver is homogeneous, the simulated liver lacks vessels. In addition, more detailed tissue parameters, such as elasticity, anisotropy, and other biomechanical properties, should be incorporated to improve simulation fidelity. However, this limitation could be overcome by characterizing the parameters during preoperative scanning to build a simulation model for training before the actual surgery. An important aspect for the medical environment is the model registration based on real-time imaging, which prevents discrepancies between simulated outcomes and real-world performance. The dynamic responses of robots, influenced by factors such as inertia, friction, and real-time environmental interactions, are critical for the practical deployment of robotic systems. Incorporating these dynamics

into our simulations is crucial for adapting our research to real-world conditions and will necessitate further adjustments and testing once integrated with actual robotic platforms.

VI. CONCLUSIONS

This work demonstrates that Fully Convolutional Networks (FCN) and Residual FCN networks present a valid and competitive alternative to iterative numerical methods for solving inverse models. They achieve comparable levels of accuracy while significantly reducing computational time. The primary contribution of this study lies in its successful reduction of computational time without compromising accuracy, regardless of the shape of the trajectories or the mechanical properties of the insertion tissue. Notably, the minimal computational time required makes this approach suitable for integration into the control loop of automated needle steering systems. However, the necessity for forward simulation to obtain the required inputs can be a limitation, as it involves performing segmentation and reconstruction on the patient's model. But consider the benefits for presented, this could be a good trade-off. Our method is particularly applicable in the medical field, achieving sub-millimeter targeting accuracy that exceeds the required threshold of 3 mm for percutaneous procedures. Therefore, the outcomes of this study are especially promising, despite the existing limitations. Future works will deal with the learning needle-tissue interactions to complete the control loop, including organ registration from medical images.

REFERENCES

[1] M. M. Arnolli, M. Buijze, M. Franken, I. A. M. J. Broeders, and D. M. Brouwer, "A precision system for computed tomography-guided needle placement in the thorax and abdomen—technical design and performance analysis," *Journal of Medical Devices*, vol. 12, no. 2, p. 021003, 04 2018.

[2] 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.

[3] 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.

[4] 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, no. 6, pp. 539–546, 2018, publisher Copyright: © 2018 Okayama University Medical School.

[5] W. Xu, J. Chen, H. Y. Lau, and H. Ren, "Data-driven methods towards learning the highly nonlinear inverse kinematics of tendon-driven surgical manipulators," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 3, p. e1774, 2017, e1774 RCS-16-0056.R1.

[6] M. Emerson, J. M. Ferguson, T. E. Ertop, M. F. Rox, J. Granna, M. Lester, F. Maldonado, E. A. Gillaspie, R. Alterovitz, R. J. Webster, and A. Kuntz, "A recurrent neural network approach to roll estimation for needle steering," *International Symposium on Experimental Robotics (ISER)*, vol. 19, pp. 334–342, 2020.

[7] K. B. Reed, A. M. Okamura, and N. J. Cowan, "Modeling and control of needles with torsional friction," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2905–2916, 2009.

[8] V. Kallem and N. J. Cowan, "Image guidance of flexible tip-steerable needles," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 191–196, 2009.

[9] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006, neural Networks.

[10] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 2012, pp. 323–329.

[11] Z. Jiang, M. Lv, Z. Wang, B. Zhao, L. Lei, G. Yu, B. Li, and Y. Hu, "Correction control of needle deflection based on kernel extreme learning machine," in *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2023, pp. 575–580.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] T.-L. Ha, J. Verde, J. Bert, and H. Courtecuisse, "Toward fluoroscopy guided robotic needle insertion for radio frequency ablation," *Computer Animation and Virtual Worlds*, vol. 36, no. 3, p. e70025, 2025.

[14] P. P. Alart, "Méthode de newton généralisée en mécanique du contact," *Journal des Mathematiques Pures et Appliquees*, vol. 76, pp. 83–108, 1997.

[15] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical Image Analysis*, vol. 18, pp. 394–410, 2014.

[16] 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.

[17] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016)* . Daejeon, South Korea: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016), Oct. 2016.

[18] 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, 2019.

[19] 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 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 2407–2413.

[20] T. L. H, J. Bert, and H. Courtecuisse, "Real-time robotic flexible needle insertion in deformable living organs using isolated objective constraint," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2024, pp. 10 998–11 003.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] P. Baksic, H. Courtecuisse, and B. Bayle, "Shared control strategy for needle insertion into deformable tissue using inverse finite element simulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 442–12 448.

[23] 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.

[24] S. Scherzinger, "Cartesian controller," https://github.com/fzi-forschungszentrum-informatik/cartesian_controllers.

[25] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (fdcc): A new approach to cartesian compliance for robotic manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4568–4575.