Autonomous tissue retraction with a biomechanically informed logic based framework

Daniele Meli[1]*, Eleonora Tagliabue[1]*, Diego Dall'Alba[1], Paolo Fiorini[1]

***Abstract*— Autonomy in robot-assisted surgery is essential to reduce surgeons' cognitive load and eventually improve the overall surgical outcome. A key requirement for autonomy in a safety-critical scenario as surgery lies in the generation of interpretable plans that rely on expert knowledge. Moreover, the Autonomous Robotic Surgical System (ARSS) must be able to reason on the dynamic and unpredictable anatomical environment, and quickly adapt the surgical plan in case of unexpected situations. In this paper, we present a modular Framework for Robot-Assisted Surgery (FRAS) in deformable anatomical environments. Our framework integrates a logic module for task-level interpretable reasoning, a biomechanical simulation that complements data from real sensors, and a situation awareness module for context interpretation. The framework performance is evaluated on simulated soft tissue retraction, a common surgical task to remove the tissue hiding a region of interest. Results show that the framework has the adaptability required to successfully accomplish the task, handling dynamic environmental conditions and possible failures, while guaranteeing the computational efficiency required in a real surgical scenario. The framework is made publicly available.**

***Index Terms*— Autonomous Robotic Surgery; Logic Programming; Biomechanical simulation;**

I. INTRODUCTION

One frontier of research in surgical robotics is the achievement of higher levels of autonomy, which has the potential to improve the quality of an intervention [1]. In particular, level 2 of autonomy as of [1] will introduce the autonomous execution of surgical sub-tasks, limiting human errors and the fatigue of surgeons, who shall monitor the overall execution and focus only on the most critical steps. An Autonomous Robotic Surgical System (ARSS) must guarantee an *interpretable* behavior, i.e. the input and output to the autonomous system must be easily understandable by a monitoring surgeon, for safety and reliability [2], [3]. Moreover, the surgical scenario is affected by high variability (e.g., patient's anatomy and tissue properties). This requires continuous reasoning on intra-operative data from sensors, in order to responsively adapt to the quick changes of the environment and to recover from possible failures.

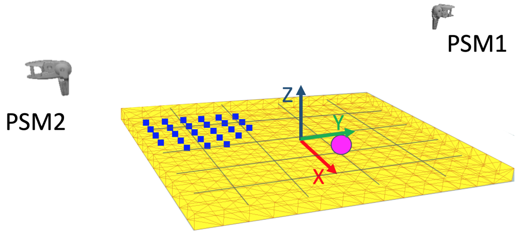


Fig. 1: The simulated environment considered for the tissue retraction task: deformable tissue (yellow), ROI (magenta sphere), attachment points (blue dots) and common reference frame. Gray lines delineate the 5 × 5 discretization for selection of grasping points.

In this paper, we approach the problem of reaching autonomy level 2 [1] in a surgical deformable environment, with a modular framework that integrates interpretable task planning and real-time situation awareness. Interpretability is guaranteed by both encoding prior expert knowledge in logic formalism at task level, and providing semantic interpretation of data from sensors with situation awareness. The proposed framework builds on our previous work [4], which exhibits fast computational performance compatible with the surgical time constraints, but has been tested only on a standard surgical training task in rigid domain. In this work, we consider a more realistic and significant surgical use case, consisting in Tissue Retraction (TR) to expose a region of interest (e.g., a tumor or other pathological areas). We extend the existing framework to operate in deformable environments by introducing a patient-specific biomechanical simulation that follows the actual condition throughout surgical task execution. The physics-based model estimates the applied forces, allowing to reduce the risk of tissue damage due to high solicitation. The capabilities of our framework are validated on a simulated TR, executed with the da Vinci Research Kit (dVRK) [5]. The specific contributions of this work are the following:

1) we show that our Framework for Robot-Assisted Surgery (FRAS) can adapt to the dynamic environmental conditions and re-plan when critical situations occur. To the best of our knowledge, this is the first attempt to failure recovery during TR;
2) we show that the integration of a biomechanical simulation allows to deal with specific problems of real surgery, e.g. lack of force feedback;
3) we show that FRAS generates interpretable plans (see attached video), guaranteeing computational efficiency even when operating in a deformable environment;
4) we publicly share the framework at https://gitlab.com/altairLab/tissue_retraction/-/tree/v1.0.

[1] Authors are with Department of Computer Science, University of Verona, Verona, Italy daniele.meli@univr.it

*Authors contributed equally and are reported in alphabetical order.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 742671 "ARS").

The paper is organized as follows: in Section II we review state of the art in autonomous TR, and we motivate the use of logic programming; in Section III we describe the TR task and the proposed autonomous framework; Section IV reports the experimental results of this paper, and Section V summarizes the main achievements and possible future research.

II. RELATED WORKS

Recent research has investigated the automation of basic operations in surgery, e.g. suturing [6], needle insertion [7] and dissection [8]. Autonomous TR combining standard motion planning algorithms with features for optimal grasping identification is performed in [9]. In [10], [11], authors have focused on the generation of optimal robotic trajectories and grasping point selection for TR, maximizing target visibility and minimizing tissue damage based on biomechanical simulation. These works assume the execution workflow is predefined, and do not deal with failure recovery and unconventionial situations which occur in real (surgical) robotic tasks.

Learning-based methods have recently shown the potential to capture variability in the surgical scenario. In [12], deep reinforcement learning is proposed to learn a policy for robotic TR in a simulated environment. [13] shows that efficient transfer of learned policies from simulation to the real system in uncertain environments can be obtained with the incorporation of few human demonstrations. [14] presents a learning strategy which incorporates human knowledge for the definition of task-specific features. However, learning-based methods fail to generalize to situations not seen at training time and do not guarantee any interpretability [15], which is essential for safety-critical scenarios as surgery.

Interpretability for autonomous robots can be guaranteed by *logic-based reasoning systems* [16]. They encode prior concepts and specifications about a robotic domain in a knowledge base (e.g. an ontology as proposed by [16]), using logic formalism which is easily readable by humans. Planning languages as PDDL then allow to apply logic inference on knowledge [17]. A logic-based task planner does not encode pre-defined workflows (as finite state machines do); instead, it performs human-like logic inference from knowledge and data from sensors, in order to generate an appropriate workflow for the current situation. Hence, the logic-based approach formally guarantees that specifications encoded in prior knowledge (which may be, e.g., safety constraints in surgery) are satisfied during the execution. However, [18] evidences the limits of ontology-based reasoning, which can only represent a static scenario, while real-time knowledge revision for re-planning is computationally inefficient. For this reason, ontologies are mostly used for supervision and assistance in surgery [19], [20]. In this paper, we instead use non-monotonic logic programming [21], which combines knowledge representation and reasoning, and permits online revision of incomplete or dynamic information. Although non-monotonic logic programming has never been applied to surgery, examples in other fields [22], [23] show its feasibility in safety-critical scenarios. In this work, we rely on *Answer Set Programming (ASP)* [24], a non-monotonic logic paradigm which is more computationally efficient and expressive if compared to its main competitor Prolog [25].

III. METHODS

In this Section, we describe and motivate the choice of the TR task as use case, and describe the proposed framework for ARSS.

A. Tissue retraction task

Tissue retraction (TR) is a common task in surgery. It aims at grasping and lifting an anatomical layer (e.g., adipose tissue) to reveal and make accessible an underlying region of interest (ROI) (e.g., a tumor). We assume that the 3D geometric model of the tissue and the 3D position of the ROI to expose are known from pre-operative data (Fig. 1). The task is executed by the two patient-side manipulators (PSMs) of the dVRK. A challenge of TR is the presence of *attachment points (APs)* between the tissue and the surrounding anatomy, which can be initialized based on statistical information [26], [27]. Avoiding APs is important in TR to prevent the application of excessive forces on highly constrained areas, thus reducing the risk of tissue tearing and/or rupture. For an efficient execution of TR, we choose the grasping point which maximizes the distance from known APs, while minimizing the distance from ROI for faster exposure.

The selected grasping point is considered *reachable* by one of the PSMs, depending on its position with respect to the tissue $x-$axis of symmetry (hence, distance on $y-$axis is considered, Fig. 1). After grasping, the PSM pulls the tissue up to expose ROI [28]. However, other strategies are needed in case the ROI is not exposed and/or critical conditions rise, e.g. excessive applied force, that can be recognized thanks to the feedback provided by a biomechanical simulation. The reasoner then generates an alternative strategy, e.g. lateral motion to expose the ROI (Fig. 4) or grasp re-planning (Fig. 5).

B. Framework for Robot Assisted Surgery (FRAS)

Fig. 2 depicts the main modules of our framework. FRAS is initialized with pre-operative information, that define the starting anatomical condition and the position of the target to reach. During task execution, a *situation awareness (SA)* module receives information from sensors and dynamic feedback from a biomechanical simulation of the environment. This information is translated to a semantic interpretation (*context*) of the environment state, and sent to the *task reasoner*. The task reasoner generates a plan, i.e. a sequence of actions towards the goal of the task. The actions are communicated to a *motion control (MC)* module, which translates them into motion commands for the robotic arms. The context and the plan can be monitored by an expert human supervisor, to guarantee the reliability of the ARSS.

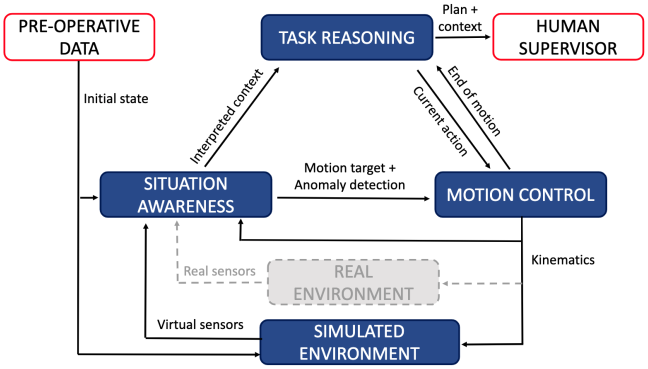


Fig. 2: Block diagram of the proposed FRAS. Blue blocks represent modules used in this paper, with arrows specifying exchanged information. Grey parts of the scheme allow extension towards application in a real surgical setup.

Algorithm 1 Situation awareness

```
Input: $S_E = \{P_t, P_f, p_{ROI}, \Sigma\}$, $S_R = \{j_{1,2}, p_{1,2}\}$; current action action; request for context $req$
Output: Set of context fluents $F$, target position $g$ for action and failure flag $fail$
Init: $fail = False$, $req = False$, action = $Null$, $P_{close} = \{p \in P_t \mid ||p - p_{ROI}||_2 < 10\,\mathrm{mm}\}$
repeat
    if $req$ then
        $F$ = compute_fluents($S_E$, $S_R$, $P_{close}$)
    if action == pull then
        $fail$ = check_failure($S_E$)
    $g$ = compute_target(action)
until end of task
```

1) Task reasoner: This module encodes task knowledge in the logic programming formalism of Answer Set Programming (ASP) [29]. An ASP program represents the domain of the task with a *description* $\mathcal{D}$ and a *history* $\mathcal{H}$. $\mathcal{D}$ comprises a *signature* Σ and axioms. Σ is the alphabet of the task, defining relevant attributes of the domain. Attributes may be *statics*, i.e., domain attributes whose values do not change over time; *fluents*, i.e., domain attributes whose values can be changed; and *actions*. Attributes may be *terms*; *atoms*, i.e. predicates of terms (e.g. `Atom(`$\mathtt{T}_1$`, ..., `$\mathtt{T}_n$`)` is an atom with terms $\mathtt{T}_{1,\dots,n}$ as arguments); and their logical negations. Values of terms are *constants* (either integers or strings). A term whose value is assigned is *ground*, and an atom is ground if its terms are ground. Axioms are logical relations between attributes. In this paper, we consider causal rules $\mathtt{A}_h$ `:-` $\mathtt{A}_{b1}$`, ..., `$\mathtt{A}_{bn}$, defining pre-conditions $\mathtt{A}_{b1,\dots,bn}$ (*body* of the rule) for grounding the *head* $\mathtt{A}_h$. Following the definition of the task introduced in Sec. III-A, statics are `block` (from now on `B`), representing a generic candidate grasping point on the tissue; and `arm` (from now on `A`), representing the robotic arm (either `psm1` or `psm2`). Fluents are relevant environmental conditions for the task. We define them as atoms with the variable `t` as argument, representing a discrete time step for temporal reasoning. They include `reachable(A, B, t)` (`B` reachable by `A`), `in_hand(A, B, t)`, `closed_gripper(A, t)` (gripper state), `distance(B1, B2, X, t)` (the distance between `B1`-`B2` is `X`), `visible_ROI(t)`, `max_height(A, t)` (pulling is no longer possible if excessive force is measured), `fixed(B, t)` (`B` is close to APs), `at(A, B, t)` (location of `A` on tissue) and `above_ROI(B, t)` (location of ROI under tissue). All fluents except `distance` are defined as *external* atoms, so they can be grounded from the SA module.

Actions for our task are `reach(A, B, t)`, `grasp(A, B, t)`, `release(A, t)`, `pull(A, B, t)` and `move(A, B, t)` (for lateral motion). Axioms express logical relations between atoms. For the task of TR, we consider axioms representing causal laws, executability constraints , choice rules and optimization statements. Causal laws define *effects of actions* on fluents (e.g., `A`'s gripper is closed after grasping, `closed_gripper(A, t) :- grasp(A, B, t-1)`). Executability constraints specify the set of actions and fluents that cannot hold concurrently (e.g., it is not possible to pull up tissue if the force applied to the tissue exceeds constraints, `:- pull(A, B, t), max_height(A, t)`). Choice rules specify the environmental *pre-conditions* which must be satisfied to execute actions. For our task, at most one single action can be executed at each time step. Then, we define a choice rule gathering pre-conditions for actions as `0 {action(t) : pre-cond(t)} 1`, which constrains the cardinality of the set of possible actions returned at each time step between 0 and 1. Optimization statements define preference in the set of possible actions, in particular for grasping point selection as defined in Sec. III-A. For instance, to maximize distance between grasping point and APs, we state `#maximize{X : reach(A, B1, t), distance(B1, B2, X, t), fixed(B2, t)}`. We finally define the goal of the TR task as the constraint that the ROI must be eventually visible within a predefined percentage (`:- visible_ROI(t)`).

Given the ASP program, an ASP solver grounds information from sensors and axioms at each time step, and returns them as $\mathcal{H}$, enhancing interpretability of the behavior of ARSS. For task planning, we select from $\mathcal{H}$ only atoms representing actions to command the robot (see attached video).

2) Situation awareness: The SA module continuously monitors the state of the robot and the environment, to provide high-level interpretation of the environmental context and reduce the risk of tissue damage. The functions of the SA module are described in Alg. 1. The SA module receives the state of the robot S_R, i.e. the 3D position of PSMs $p_{1,2}$ and joint angles of grippers $j_{1,2}$, from robot kinematics; and the state of the environment S_E, including the point cloud of tissue points P_t and APs $P_f \subseteq P_t$; the set Σ of mechanical forces at each tissue point from the biomechanical simulation; and the position of the ROI p_{ROI}. All the quantities in S_R and S_E are referred to the same

Algorithm 2 compute_fluents

```
 1: Input: $S_E = \{P_t, P_f, p_{ROI}, \Sigma\}$, $S_R = \{j_{1,2}, p_{1,2}\}$, $P_{close}$
 2: Output: Set of context fluents $F$
 3: Init: $F = \emptyset$, set $\delta \in [0, 1]$
 4: Compute $P_v = \{p \in P_{close} \mid ||p - p_{ROI}||_2 > 10\,\mathrm{mm}\}$
 5: % ROI visibility ($|\cdot|$ is the cardinality of a set)
 6: if $\frac{|P_v|}{|P_{close}|} > \delta$ then
 7:     $F$.append(visible_ROI)
 8: % Gripper state
 9: for $i \in \{1, 2\}$ do
10:     if $j_i$ <20° then
11:         $F$.append(closed_gripper(A$_i$))
12: % Finite set of points
13: $B$ = discretize($P_t$)
14: for $b \in B$ do
15:     % ROI location under tissue
16:     if $p_{ROI} \in b$ then
17:         $F$.append(above_ROI(b))
18:     % Reachability with respect to $x-$axis, see Sec. III-A
19:     $j = argmin_{i=\{1,2\}}||y_i - y_b||_1$
20:     $F$.append(reachable(A$_j$, b))
21:     % Points close to APs
22:     if $\exists p \in P_f : \ p \in B$ then
23:         $F$.append(fixed(b))
24:     for $i \in \{1, 2\}$ do
25:         % Arm location
26:         if $||p_i - b||_2$ <5 mm then
27:             $F$.append(at(A$_i$, b))
28:             % Tissue grasped
29:             if $j_i$ <20° then
30:                 $F$.append(in_hand(A$_i$, b))
    return $F$
```

kinematic frame (Fig. 1). External fluents for task reasoning are computed by the function *compute_fluents* as described by Alg. 2 (time `t` is omitted because it is assigned at ASP level). Specifically, Lines 7-8 compute ROI visibility. To this purpose, at the beginning of the task (Line 3 in Alg. 1) we compute P_{close} as the set of tissue points close to the ROI within a fixed threshold ($10\,\mathrm{mm}$ in this work). The ROI is assumed to be visible when a pre-defined percentage (above a threshold δ) of P_{close} becomes far enough from the ROI itself (e.g., more than $10\,\mathrm{mm}$ in this work), thus indicating that the tissue has been displaced such that the ROI is accessible. When an action has to be executed, function *compute_target* computes the motion target g for it. The target is needed only for actions `reach(A, B, t)` (the position of grasping point `B`, see Fig. 1) and `move(A`$_i$`, B, t)` ((x_B, y_B, z_i)). During the execution of `pull(A, B, t)` action, the function *check_failure* verifies that $max\Sigma < \epsilon$, where ϵ is a threshold depending on the tissue mechanical properties. Otherwise, failure is notified to the MC module to interrupt the execution, and the task reasoner re-plans (with grounding of `max_height(A, t)`).

3) Motion control: The MC module receives the current action in the plan from the task reasoner. For each action, it selects the appropriate motion primitive from a pre-defined set, as in [4]. For actions `reach(A, B, t)` and `move(A, B, t)`, it also receives the goal position from the SA module. Otherwise, the target is $(x_i, y_i, z_i + h)$ for `pull(A`$_i$`, B, t)`, i.e. the tissue is pulled up to a pre-defined height (h =50 mm in our task). Grasping and releasing of the gripper are direct joint closing and opening commands, respectively. In this work, we consider only linear trajectories, keeping a fixed orientation at the end-effector.

4) Simulated environment: A physics-based model of the anatomical environment is created starting from patient-specific geometry and mechanical properties extracted from pre-operative data. At each time instant, the simulation updates the tissue state based on the current robot configuration received from the MC module. To obtain a physically accurate result, we model the deformable behavior of anatomical tissues exploiting continuum mechanics laws, solved with the finite element (FE) method [30]. The choice of a physics-based method allows to continuously estimate forces exerted on the tissue, a crucial quantity to monitor and avoid unnecessary tissue damage. In the considered simulation environment, grasping is implemented by constraining the tissue points closest to the simulated PSM (distance below a small threshold, 5 mm in our task) to follow PSM motion, similarly to [10], [12]. Grasping is activated or released based on commands received from the MC module. Current implementation does not consider the possibility of slippage or breaking contacts between the tool and the tissue, that might happen during real retraction.

IV. RESULTS AND DISCUSSION

For the experimental evaluation of FRAS, we use the dVRK to control the surgical robot and receive kinematic information. The task reasoning module is implemented using the state-of-the-art ASP solver Clingo [31]. We assess performance of the framework on TR of a rectangular soft tissue sample of dimension $100 \times 120 \times 5$ mm, a realistic size for robot-assisted surgical applications [28], in a simulated environment. Hence, in this evaluation, all the information about the state of the environment S_E (including the point cloud of tissue points P_t) are provided to the SA module by the simulation itself (Fig. 2). The tissue is modelled as a St Venant-Kirchhoff material with Young's modulus 3 kPa and Poisson ratio 0.45, a common modelling choice for adipose tissues [32], [33], which are often responsible of hiding structures of interest during real surgery [28]. APs are defined as random tissue patches, following the strategy proposed in [33]. In order to ensure computational tractability of selecting the grasping point at the logic reasoning level, we consider a finite set of candidate grasping points, uniformly spaced on an $N \times N$ grid, centered on the tissue of interest (Fig. 1). In all tests, we set the threshold for visibility $\delta = 70\%$ in Alg. 2. Tests are run on a PC with 2.6 GHz Intel Core i7 processor and 16 GB RAM.

A. Autonomous tissue retraction

We run experiments to evaluate the capability of the framework to perform autonomous TR handling the different conditions that may occur. In this evaluation, we consider candidate grasping points distributed on a grid of resolution 7×7 (i.e., one possible grasping point every $15\,\mathrm{mm}$). The force limit is set to $\epsilon = 0.5\,\mathrm{N}$, to avoid any possible dangerous tissue manipulation. Success rate of FRAS is assessed by measuring the average visibility percentage as defined in Alg. 1, reached upon task completion.

We run 100 simulations initialized with different random environmental states (i.e., p_{ROI} and P_f). After the tissue is grasped, two possible situations might happen, depending solely on the initial state. In the simplest case (Scenario A in Fig. 3), the force applied to the tissue never exceeds the defined threshold ϵ; thus, the tissue is pulled upwards until the ROI is exposed. Instead, in case the force overcomes the safety threshold ϵ, the PSM moves the tissue laterally towards the ROI in order to reveal it (Scenario B in Fig. 4).

To better assess the contribution of the force feedback, we consider a third situation where the same 100 simulations are run again, but ignoring the constraint on the distance from APs for grasping point selection (Scenario C in Fig. 5). In this situation, the grasping point is selected as the one closest to the ROI, regardless of the position of APs. This scenario allows to evaluate how FRAS would react in case the chosen grasping point is very close to an AP, which may happen when there are many APs and the ROI is very close to them. We note that when grasping close to the APs, the force goes beyond ϵ immediately after grasping (i.e., when the $z-$position of the PSM is close to the tissue). Then, the task reasoner must re-plan grasping to another point, now maximizing also the distance from APs.

The three scenarios are illustrated in the video attachment to this paper, evidencing the interpretability of the plan and the environmental context description. Even though a simplified tissue model is considered for evaluation in this paper, it is worth noticing that our framework can provide a plan for any tissue geometry and state of the environment.

We have found that re-planning with Scenario B occurs in 36 simulations, while re-planning in Scenario C is needed in 46 cases. The achieved visibility percentage (mean ± standard deviation) is $96.89\% \pm 6.37\%$ (Scenarios A, B and C), demonstrating that, on average, our autonomous framework is able to expose the region of interest to a high extent, thus successfully accomplishing the task. Results show that re-planning does not affect the final visibility percentage (average visibility is 96.41% when re-planning is needed in Scenarios B and C, comparable to the average over all executions).

One of the contributions of this work lies in the integration of the dynamic feedback from the biomechanical simulation for enhanced situation awareness. Our results demonstrate that the force feedback provided by the simulation triggers re-planning in slightly more than one third of the cases. In real surgery, where the surgeon has no information about the force applied to the tissue, all these cases represent potentially dangerous situations where tissue damage is highly probable. In fact, the mechanical stress on the tissue can only be roughly evaluated from visual feedback or based on the surgeon's expertise. In particular, when grasping very close to APs (Scenario C), the force feedback allows to immediately recognize the high risk of tissue damage and adapt the behavior to successfully complete the task.

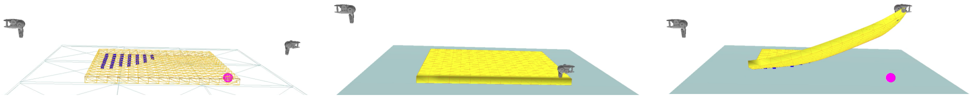
We further verify the importance of force feedback from simulation by running a set of 100 simulations where no force limit is set. In this batch of simulations, the maximum force applied to the tissue is (1.25 ± 0.96) N (mean ± standard deviation), i.e. almost 3 times the pre-defined threshold, with a peak of $8.50\,\mathrm{N}$ (when grasping occurs close to APs). As a consequence, integration of force information from the simulation is crucial to minimize the risk of tissue tearing or rupture.

B. Planning efficiency

An additional requirement of the task reasoning module is responsive reaction to the quickly changing anatomical environment. The computational performance of task reasoner is evaluated as the *planning time*, defined as the time required by Clingo to compute a plan, given a context interpretation from the SA module. Hence, it does not depend on the rate at which sensor data are received. The planning time includes time for grasping point selection; thus, it may be affected by two main parameters: the number of candidate grasping points and the percentage $f_{\%}$ of APs. In fact, they both influence the size of the search space for possible grasping points. In this evaluation, we consider different resolutions of the grid defining candidate grasping points $N \in \{5, ..., 10\}$. For each value of N, we run 25 simulations for each value $f_{\%} \in \{10\%, 30\%, 50\%, 70\%\}$. This results in 600 total simulations, each initialized with a different random p_{ROI} and P_f (according to the value of $f_{\%}$).

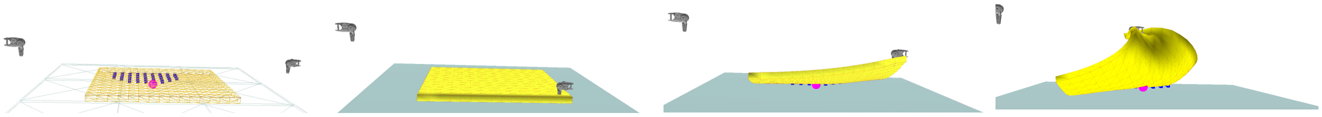
The second column of Table I reports the time required by the task reasoner to generate the initial plan. Obtained results show that the most influencing parameter is N, while $f_{\%}$ does not significantly affect the average planning time. The task reasoner is able to compute the grasping point and a plan within $1.5\,\mathrm{s}$ when using a 7×7 grid discretization for grasping point definition, demonstrating high efficiency. Although here we do not intend to provide a strategy for optimal grasping point selection, obtained results suggest that the proposed approach can be used in combination with state-of-the-art grasping point selection algorithms, to significantly improve their performance through search space shrinkage even in real surgery [11].

The computational capabilities of FRAS are also validated in terms of re-planning time (Scenarios B-C). Results are reported in the last two columns of Table I. We run 100 simulations for each scenario and for each value of N, setting constant $f_{\%} = 50\%$ (since $f_{\%}$ has shown not to influence the planning time). The average re-planning time for Scenario B is approximately half of the planning time obtained at the same value of N. This happens because the initial grasping



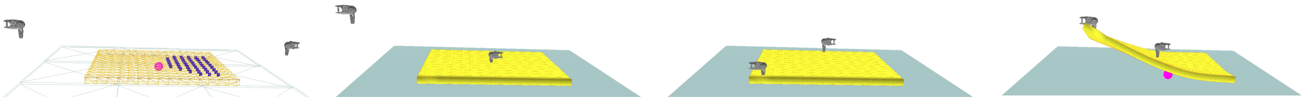
(a) Initial state (b) Grasping tissue (c) ROI revealed

Fig. 3: Scenario A, successful pulling with no replanning.



(a) Initial state (b) Grasping tissue (c) Pulling interrupted (d) ROI revealed

Fig. 4: Scenario B, re-planning with `move(A, B, t)` in c).



(a) Initial state (b) First grasping (c) Second grasping (d) ROI revealed

Fig. 5: Scenario C, re-planning after grasping close to an attachment point in b).

TABLE I: Planning time and re-planning times in Scenarios B and C (average ± standard deviation), with different tissue grid size N and percentage of APs $f_{\%}$.

N	**Planning time** [s]				**Re-planning time B** [s]	**Re-planning time C** [s]
	$\mathbf{f_{\%} = 10\%}$	$\mathbf{f_{\%} = 30\%}$	$\mathbf{f_{\%} = 50\%}$	$\mathbf{f_{\%} = 70\%}$	$\mathbf{f_{\%} = 50\%}$	$\mathbf{f_{\%} = 50\%}$
5	0.25 ± 0.02	0.24 ± 0.02	0.23 ± 0.01	0.24 ± 0.01	0.12 ± 0.04	0.32 ± 0.02
6	0.68 ± 0.02	0.65 ± 0.02	0.63 ± 0.01	0.62 ± 0.01	0.32 ± 0.09	1.01 ± 0.10
7	1.50 ± 0.02	1.48 ± 0.03	1.41 ± 0.02	1.42 ± 0.01	0.72 ± 0.19	1.95 ± 0.04
8	3.06 ± 0.04	2.97 ± 0.06	2.90 ± 0.04	2.88 ± 0.02	1.52 ± 0.49	4.03 ± 0.08
9	5.88 ± 0.08	5.70 ± 0.15	5.51 ± 0.03	5.51 ± 0.05	2.77 ± 0.60	7.74 ± 0.16
10	10.75 ± 0.20	10.21 ± 0.28	9.92 ± 0.09	9.87 ± 0.08	4.95 ± 1.05	16.00 ± 1.43

point is far from APs, hence the system does not need to select another grasping point, which represents a significant part of the ASP computational effort. On the contrary, in Scenario C the system requires more time to re-plan because tissue releasing and new grasping point selection are required. Overall, the re-planning time is below 2 s for a 7×7 grasping points discretization, which guarantees efficient adaptation to the quickly changing anatomical environment, introducing only a minor delay which is compatible with intra-operative times.

V. CONCLUSIONS

This work presents FRAS, a logic-based framework for autonomous surgical task execution, supporting real-time planning and failure recovery in deformable anatomical environment. Performance of FRAS is preliminarly assessed on a simulated tissue retraction task. Logic-based task reasoning and real-time situation awareness guarantee interpretability of the workflow of execution (environmental context interpretation and consequent plan generation), which is necessary for monitoring and to guarantee the reliability of the ARSS. Logics can also reduce the computational burden of standard state-of-the-art algorithms for grasping selection. A biomechanical simulation provides feedback on the forces applied on the tissue, reducing the risk of tissue damage. Future works will focus on the validation on a real scenario, where the role of the SA module and the simulation will be fundamental to close the control loop and ensure reliable execution. Furthermore, we plan to exploit the output from the SA module to continuously update the simulated environment and compensate for possible modelling uncertainties [34]. In particular, we will integrate strategies to update APs during interaction with the tissue, such as [27], [35], compensating for partial or imprecise knowledge about their position at the beginning of the task. Extension to more complex surgical tasks would also require the integration with more sophisticated motion planning strategies [4].

REFERENCES

[1] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, V. J. Santos, and R. H. Taylor, "Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy," *Science Robotics*, vol. 2, no. 4, p. 8638, 2017.

[2] M.-C. Fiazza and P. Fiorini, "Design for interpretability: Meeting the certification challenge for surgical robots," in *2021 International Conference on Intelligence and Safety for Robotics (ISR)*. IEEE, 2021.

[3] "Regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts," accessed: 2021-09-21. [Online]. Available: https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0001.02/DOC_1&format=PDF

[4] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini, "Autonomous task planning and situation awareness in robotic surgery," in *2020 IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3144–3150.

[5] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[6] S. A. Pedram, C. Shin, P. W. Ferguson, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous suturing framework and quantification using a cable-driven surgical robot," *IEEE Transactions on Robotics*, 2020.

[7] R. Muradore, P. Fiorini, G. Akgun, D. E. Barkana, M. Bonfe, F. Boriero, A. Caprara, G. De Rossi, R. Dodi, O. J. Elle, F. Ferraguti, L. Gasperotti, R. Gassert, K. Mathiassen, D. Handini, O. Lambercy, L. Li, M. Kruusmaa, A. O. Manurung, G. Meruzzi, H. Q. Nguyen, N. Preda, G. Riolfo, A. Ristolainen, C. Sanna, Alberto anda Secchi, M. Torsello, and A. E. Yantac, "Development of a cognitive robotic system for simple surgical tasks," *International Journal of Advanced Robotic Systems*, vol. 12, no. 4, p. 37, 2015.

[8] T. D. Nagy and T. Haidegger, "A dvrk-based framework for surgical subtask automation," *Acta Polytechnica Hungarica*, pp. 61–78, 2019.

[9] A. Attanasio, B. Scaglioni, M. Leonetti, A. F. Frangi, W. Cross, C. S. Biyani, and P. Valdastri, "Autonomous tissue retraction in robotic assisted minimally invasive surgery–a feasibility study," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6528–6535, 2020.

[10] S. Patil and R. Alterovitz, "Toward automated tissue retraction in robot-assisted surgery," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2088–2094.

[11] R. Jansen, K. Hauser, N. Chentanez, F. Van Der Stappen, and K. Goldberg, "Surgical retraction of non-uniform deformable layers of tissue: 2d robot grasping and path planning," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 4092–4097.

[12] E. Tagliabue, A. Pore, D. Dall'Alba, E. Magnabosco, M. Piccinelli, and P. Fiorini, "Soft tissue simulation environment to learn manipulation tasks in autonomous robotic surgery," in *2020 IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[13] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3875–3881.

[14] S. A. Pedram, P. W. Ferguson, C. Shin, A. Mehta, E. P. Dutson, F. Alambeigi, and J. Rosen, "Toward synergic learning for autonomous manipulation of deformable tissues via surgical robots: An approximate q-learning approach," in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2019, pp. 878–884.

[15] R. Goebel, A. Chander, K. Holzinger, F. Lecue, Z. Akata, S. Stumpf, P. Kieseberg, and A. Holzinger, "Explainable ai: the new 42?" in *International cross-domain conference for machine learning and knowledge extraction*. Springer, 2018, pp. 295–303.

[16] M. Tenorth and M. Beetz, "Knowrob – a knowledge processing infrastructure for cognition-enabled robots," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566–590, 2013.

[17] M. Fox and D. Long, "Pddl2.1: An extension to pddl for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.

[18] M. Ginesi, D. Meli, H. C. Nakawala, A. Roberti, and P. Fiorini, "A knowledge-based framework for task automation in surgery," in *2019 19th International Conference on Advanced Robotics (ICAR)*, Dec 2019, pp. 37–42.

[19] H. Nakawala, R. Bianchi, L. E. Pescatori, O. De Cobelli, G. Ferrigno, and E. De Momi, ""Deep-Onto" network for surgical workflow and context recognition," *International journal of computer assisted radiology and surgery*, vol. 14, no. 4, pp. 685–696, 2019.

[20] D. Katić, A.-L. Wekerle, F. Gärtner, H. Kenngott, B. P. Müller-Stich, R. Dillmann, and S. Speidel, "Ontology-based prediction of surgical events in laparoscopic surgery," in *Medical Imaging 2013: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 8671. International Society for Optics and Photonics, 2013, p. 86711A.

[21] Y. Dimopoulos, B. Nebel, and J. Koehler, "Encoding planning problems in nonmonotonic logic programs," in *European Conference on Planning*. Springer, 1997, pp. 169–181.

[22] M. Gebser, P. Obermeier, T. Schaub, M. Ratsch-Heitmann, and M. Runge, "Routing driverless transport vehicles in car assembly with answer set programming," *Theory and Practice of Logic Programming*, vol. 18, no. 3-4, pp. 520–534, 2018.

[23] A. Falkner, G. Friedrich, K. Schekotihin, R. Taupe, and E. C. Teppan, "Industrial applications of answer set programming," *KI-Künstliche Intelligenz*, vol. 32, no. 2-3, pp. 165–176, 2018.

[24] V. Lifschitz, "Answer set programming and plan generation," *Artificial Intelligence*, vol. 138, no. 1-2, pp. 39–54, 2002.

[25] A. Colmerauer, "An introduction to prolog iii," in *Computational Logic*. Springer, 1990, pp. 37–79.

[26] R. Plantefève, I. Peterlik, N. Haouchine, and S. Cotin, "Patient-specific biomechanical modeling for guidance during minimally-invasive hepatic surgery," *Annals of biomedical engineering*, vol. 44, no. 1, pp. 139–153, 2016.

[27] S. Nikolaev and S. Cotin, "Estimation of boundary conditions for patient-specific liver simulation during augmented surgery," *International Journal of Computer Assisted Radiology and Surgery*, 2020.

[28] S. B. Bhayani, "da vinci robotic partial nephrectomy for renal cell carcinoma: an atlas of the four-arm technique," *Journal of robotic surgery*, vol. 1, no. 4, pp. 279–285, 2008.

[29] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, and T. Schaub, "Asp-core-2 input language format," *Theory and Practice of Logic Programming*, vol. 20, no. 2, pp. 294–309, 2020.

[30] J. Zhang, Y. Zhong, and C. Gu, "Deformable models for surgical simulation: a survey," *IEEE reviews in biomedical engineering*, vol. 11, pp. 143–164, 2017.

[31] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and S. Thiele, "A user's guide to gringo, clasp, clingo, and iclingo," 2008.

[32] N. Alkhouli, J. Mansfield, E. Green, J. Bell, B. Knight, N. Liversedge, J. C. Tham, R. Welbourn, A. C. Shore, K. Kos, *et al.*, "The mechanical properties of human adipose tissues and their relationships to the structure and composition of the extracellular matrix," *American Journal of Physiology-Endocrinology and Metabolism*, vol. 305, no. 12, pp. E1427–E1435, 2013.

[33] E. Tagliabue, D. Dall'Alba, M. Pfeiffer, M. Piccinelli, R. Marin, U. Castellani, S. Speidel, and P. Fiorini, "Data-driven intra-operative estimation of anatomical attachments for autonomous tissue dissection," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1856–1863, 2021.

[34] J. Y. Wu, P. Kazanzides, and M. Unberath, "Leveraging vision and kinematics data to improve realism of biomechanic soft tissue simulation for robotic surgery," *International journal of computer assisted radiology and surgery*, vol. 15, no. 5, pp. 811–818, 2020.

[35] E. Tagliabue, M. Piccinelli, D. Dall'Alba, J. Verde, M. Pfeiffer, *et al.*, "Intra-operative update of boundary conditions for patient-specific surgical simulation," in *Medical Image Computing and Computer Assisted Intervention*. Springer International Publishing, 2021.