A First Evaluation of a Multi-Modal Learning System to Control Surgical Assistant Robots via Action Segmentation

Giacomo De Rossi, Marco Minelli, Serena Roin, Fabio Falezza, Alessio Sozzi, Federica Ferraguti, Francesco Setti, Marcello Bonfè, *Member, IEEE*, Cristian Secchi, *Senior Member, IEEE*, and Riccardo Muradore, *Member, IEEE*

***Abstract*—The next stage for robotics development is to introduce autonomy and cooperation with human agents in tasks that require high levels of precision and/or that exert considerable physical strain. To guarantee the highest possible safety standards, the best approach is to devise a deterministic automaton that performs identically for each operation. Clearly, such approach inevitably fails to adapt itself to changing environments or different human companions. In a surgical scenario, the highest variability happens for the timing of different actions performed within the same phases. This paper presents a cognitive control architecture that uses a multi-modal neural network trained on a cooperative task performed by human surgeons and produces an action segmentation that provides the required timing for actions while maintaining full phase execution control via a deterministic Supervisory Controller and full execution safety by a velocity-constrained Model-Predictive Controller.**

***Index Terms*—Medical robotics, cognitive robotics, R-MIS, action segmentation, model-predictive control.**

I. Introduction

Any ROBOTIC Minimally-Invasive Surgery (R-MIS) system has to comply with tight requirements to be allowed within an operating room, where the actuated instruments must interact with both soft tissues and hard surfaces, such as needles, clips, and between themselves. Currently, all robotic platforms within an operating room primarily rely on surgeons to provide all guarantees through their experience and direct instrumental control via teleoperation. For instance, the most advanced robotic platform available today in the operating room is the daVinci Surgical Platform, a remote teleoperation platform for laparoscopic surgery that does not present any automation degree and provides only video as feedback to the surgeon to maintain the highest possible level of control stability.

The autonomous execution of a task by robots is mostly relegated to industrial applications where robotic platforms execute repetitive tasks with minimal to no cooperation with humans: the focus is on executing precisely the same motions in the most efficient way when positioned in a highly structured environment. The research in robotics, however, is pushing for the introduction of cooperative tasks in which both the motion accuracy and cognition level need to be robust under any condition [1]. In medical robotics, the main effort is nowadays spent in the development of autonomous and semi-autonomous technologies to R-MIS. A comprehensive study performed in [2] evaluates the impact of autonomous technologies on medical/surgical practice and emphasises the need of human cooperation and supervision in the future of autonomous robotic surgeries. Among many applications available in literature, the most relevant ones are the recognition of the different phases in an endoscopic surgery addressed with deep neural networks [3] and the implementation of a knowledge-based ontology approach [4]. Other works apply unsupervised learning technique to overcome the problem of human labelling of data [5], [6], with the latter focusing specifically on its potential for robotics application. The necessary level of interaction to achieve full cooperation with surgeons will push the dexterity, perception and cognition capabilities beyond the current limits of robotics applications.

The SARAS[1] solo-surgery platform will be a very sophisticated example of a shared-control system: a surgeon operates remotely a pair of robotic laparoscopic tools (i.e., the daVinci Surgical Platform) and cooperates with the two novel SARAS autonomous robotic arms inside a shared environment to perform complex surgical procedures. The goal of the project is to define the required technologies to provide an experimental

Manuscript received December 24, 2020; accepted May 4, 2021. Date of publication May 20, 2021; date of current version August 26, 2021. This article was recommended for publication by Associate Editor D. Elson and Editor P. Dario upon evaluation of the reviewers' comments. This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 779813 (*SARAS*). *(Corresponding author: Giacomo De Rossi.)*

Giacomo De Rossi, Serena Roin, Fabio Falezza, Francesco Setti, and Riccardo Muradore are with the Department of Computer Science, University of Verona, 37134 Verona, Italy (e-mail: giacomo.derossi@univr.it; serena.roin@univr.it; fabio.falezza@univr.it; francesco.setti@univr.it; riccardo.muradore@univr.it).

Marco Minelli, Federica Ferraguti, and Cristian Secchi are with the Department of Engineering Sciences and Methods, University of Modena and Reggio Emilia, 42122 Reggio Emilia, Italy (e-mail: marco.minelli@unimore.it; federica.ferraguti@unimore.it; cristian.secchi@unimore.it).

Alessio Sozzi and Marcello Bonfè are with the Department of Engineering, University of Ferrara, 44122 Ferrara, Italy (e-mail: alessio.sozzi@unife.it; marcello.bonfe@unife.it).

Digital Object Identifier 10.1109/TMRB.2021.3082210

[1] SARAS is an EU founded project and stands for Smart Autonomous Robotic Assistant Surgeon, details at www.saras-project.eu

robotic platform that intends to effectively substitute the assistant surgeon next to the patient within the operating room.

Within the classification of autonomy grade in a surgical system [7], this work locates at a level 2: the system is bounded to operate reactively to the surgeon's actions and follow their lead during the operation while providing assistance to complete the tasks. The general architecture of the cognitive control has been formalised in our previous work [8]. This paper refines the architecture to fulfil the requirements for completing a laparoscopic pick-and-place cooperative task, which is a standard training procedure for trainee surgeons, in a semi-autonomous manner using the novel SARAS robotic minimally-invasive tools. It represents a solid basis to reach the further goals for a cooperative robotic platform for an entire surgical procedure.

II. Problem Statement and Architecture Description

This paper contributes to the state-of-the-art primarily by integrating multiple perception and control technologies with specific attention given to the safety of operation. Safety has obvious implications in the field of surgical robots and is reflected in how the majority of publications are dedicated to overcome issues that arise during both manual and teleoperated surgeries [9], [10], [11]. Specifically, previous applications of Model Predictive Controllers (MPCs) can be found to improve visual servoing control of underactuated devices within the confined environment of the human anatomy [12], [13], [14]. For the advancement of autonomous controls in this scenario, most of the literature presents case study applications of classic control [15], [16], [17], with only the work presented in [18] formulating a control strategy in line with the goal of this work. However, to the best of our knowledge, this work and its predecessor [8] represent the first attempts at direct cooperation between a surgeon and an autonomous laparoscopy manipulator using high-level cognition with improved safety control strategies.

Figure 1 shows the block diagram of the overall system. The main surgeon is the central figure with control over the entire process: they teleoperate the daVinci Surgical System which produces images $\mathcal{I}$ and Cartesian poses ξ. These are processed by the AI module along with the Cartesian poses of the SARAS arm x using the knowledge of the training data $(\hat{\mathcal{I}}, \hat{x}, \hat{\xi})$. The evaluated action $\bar{A}$ and confidence $\bar{\alpha}$ are passed to the supervisory controller that formalizes in a deterministic manner the task knowledge, thus missing only the correct temporal execution and unexpected events. Finally, the MPC receives the current goal x_g and confidence level $\alpha(k)$, with k as the discrete time variable, needed to control the SARAS arm.

The entire system represents an initial evaluation for a semi-autonomous robotic surgical assistant system that aims at the integration of perception, decision, planning, and action. It has been evaluated over a surgical training scenario that is clearly a simplification of a surgical operation, yet it is still realistic and challenging.

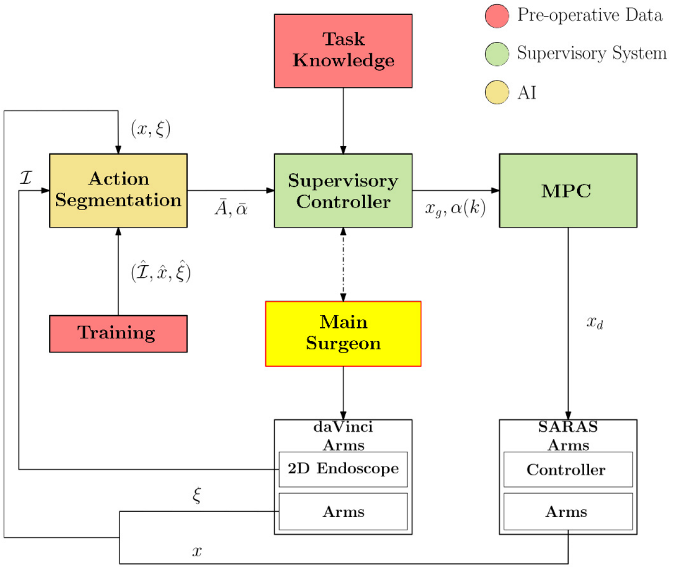
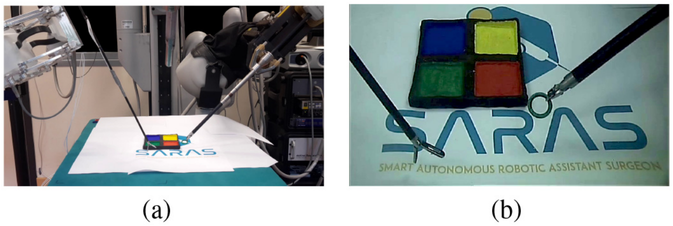


Fig. 1. Control architecture schematics. The dashed line indicates an event-based information stream between the Supervisory Controller and the Main Surgeon, i.e., an user input request after displaying an error condition. MPC stands for *Model-Predictive Controller*.

The foremost attention has been dedicated to the design of an Action Segmentation neural network architecture: it uses multi-modal learning capabilities over image data and kinematic trajectories of the robots to provide high level of confidence for a correct real-time temporal sequencing. The network topology is designed to be easily adapted to more complex tasks than the one presented hitherto. As the neural network provides the estimated timing for the action execution, a *hybrid automaton* formalizes the pre-operative task knowledge into a sequence of sub-tasks by controlling the robot with required goal points and grasp directives.

We introduce the sensible concept that a manipulator should move faster whenever it is confident on what it is expecting to do in the scene and with caution (slower) every time it is not sure on what movement it has to perform. Therefore, the correct control velocities for the SARAS arm's lower level controller are computed by a Model Predictive Control by modulating over both the confidence of the action segmentation module and the distance to obstacles in the scene, primarily the surgeon's teleoperated arm. The former constraint assures both continuity and safety in execution by restraining the velocity in the event of misidentified actions; the latter guarantees that the autonomous arm and the teleoperated one maintain a minimal safety distance to minimize interference and unintended contacts between them. This becomes a non-trivial problem when applied to the standard laparoscopy instruments mounted on the robot and their relative mechanical limitations, the foremost being the requirement for a remote center-of-motion (RCM) at the instrument entry point (trocar). The RCM represents a prerogative for laparoscopic surgery since it guarantees the safety of the patient as the pivot point remains fixed relative to the epidermis, thus preventing tissue tears. Together with the RCM constraint, the absence of a spherical wrist in laparoscopic instruments on the SARAS



(a) (b)

Fig. 2. Experimental setup. (a) daVinci® arm (right) and SARAS arm developed by Medineering™ (left). (b) same scene seen through the left endoscope camera.

arm at the end-effector limits the degrees of freedom available for obstacle avoidance and grasping.

A. Robot Description

Along with the commercial daVinci robot, which is teleoperated by the surgeon, SARAS employs for the task a novel, specifically-designed assistant robot to operate in minimally invasive scenarios of laparoscopic surgeries. Indeed, the SARAS robot is based on an actuated dual-trapezoid parallel structure that allows to set a software RCM constraint at any point along the laparoscopic tool. The tools mounted on the SARAS robot are standard laparoscopic tools, i.e., rigid cylinders $\approx$ 400mm long and a varying diameter (5 $\approx$ 12mm), with an actuated instrument on the tip, as shown in Figure 2. Finally, the scene is captured through a stereoscopic *endoscope* held by the dedicated endoscopic arm of the daVinci platform, which operates under the same constraints as the other arms.

B. A Semi-Autonomous Cooperative Task

Validation is performed by completing an experiment that consists in a pick-and-place exercise where one daVinci arm is teleoperated and one SARAS arm is autonomous. The user is instructed to pick up a colored ring placed in the scene, either red, blue or green, and to bring it closer to the camera for color identification. The SARAS arm, using both cognitive and geometrical information inferred from image and kinematic data, moves towards the ring; after grasping it, the robot waits until the other arm releases the ring and, finally, leaves the exchange area to deliver the ring to the corresponding target by color.

Each data acquisition session was prepared over multiple sittings with the intent of avoiding overfitting by excessive duplicates: both the orientation of the target square, shown in Figure 2, and the initial position of the ring were randomized, along with the light conditions and the endoscopic camera angle. Moreover, the final dataset contains five recordings per ring color to provide a sufficient and differentiated amount of data to the learning process. The process was divided into 8 different fine-grained actions for the main surgeon (MS) and the assistant surgeon (AS):

A01	MS moves to the ring;
A02	MS picks the ring;
A03	MS moves the ring to the exchange area;
A04	AS moves toward the ring;
A05	AS grasps the ring and MS leaves the ring;
A06	AS moves with the ring to the correct delivery area;
A07	AS drops the ring in the corresponding target;
A08	AS moves back to the starting position.

The task can be also divided into three distinct phases:

- The **surgeon phase**, where the daVinci moves to the ring and picks it up (actions A01-A02);
- The **cooperation phase** where the ring is brought to the exchange area and the SARAS arm moves there and picks the ring (actions A03-A04-A05);
- The **execution phase** in which SARAS, autonomously, brings the ring to the correct target area and moves away (actions A06-A07-A08).

C. Contributions

This paper showcases a valid approach to design *shared control* architectures for semi-autonomous, robotic minimally-invasive surgery that adopts high level semantics deduced from the surgical scene to coordinate with a human surgeon. The main contributions over the state-of-the-art in surgical controls, and specifically the results in [8], can be summarized in:

- a multi-modal cognitive system that improves action segmentation performance in terms of edit score by encompassing both visual and kinematic information and the adoption of a Temporal Convolutional Neural Network that allows for a better identification of actions of different duration;
- a re-designed supervisory controller that guarantees a higher safety level through the adoption of thresholds on the confidence of predicted actions.

As in [8], a Model-Predictive Controller computes the optimal control velocities modulated by both the confidence level of the cognitive system and a minimum safety distance among the tools; its improved computational performance from the version adopted in [8] allows to operate at higher control frequency over a longer prediction horizon. The choice of the weighting matrices and the control horizon within the MPC metric have not been addressed in detail as they are not the focus of this paper.

III. Action Segmentation

The *action segmentation* has to operate within stringent timing and performance requirements to be applied online as a soft-sensor. Indeed, the underlying model must:

- be *reliable*, which can be verified by the low incidence of false positives and negatives, and the percentage of correctly evaluated sequences;
- be *robust*, which is tested under varying conditions for the experimental setup (lighting, camera orientation, target variation etc.)
- provide *real-time evaluation* for its application as an advanced soft-sensor taking as input fast-changing signals and providing as output commands to lower-level controllers. This requires both data buffering operations and a small memory footprint not to hinder cyclic computations.

To comply with these requirements, we chose to implement a neural network, called *EdSkResNet*, that integrates

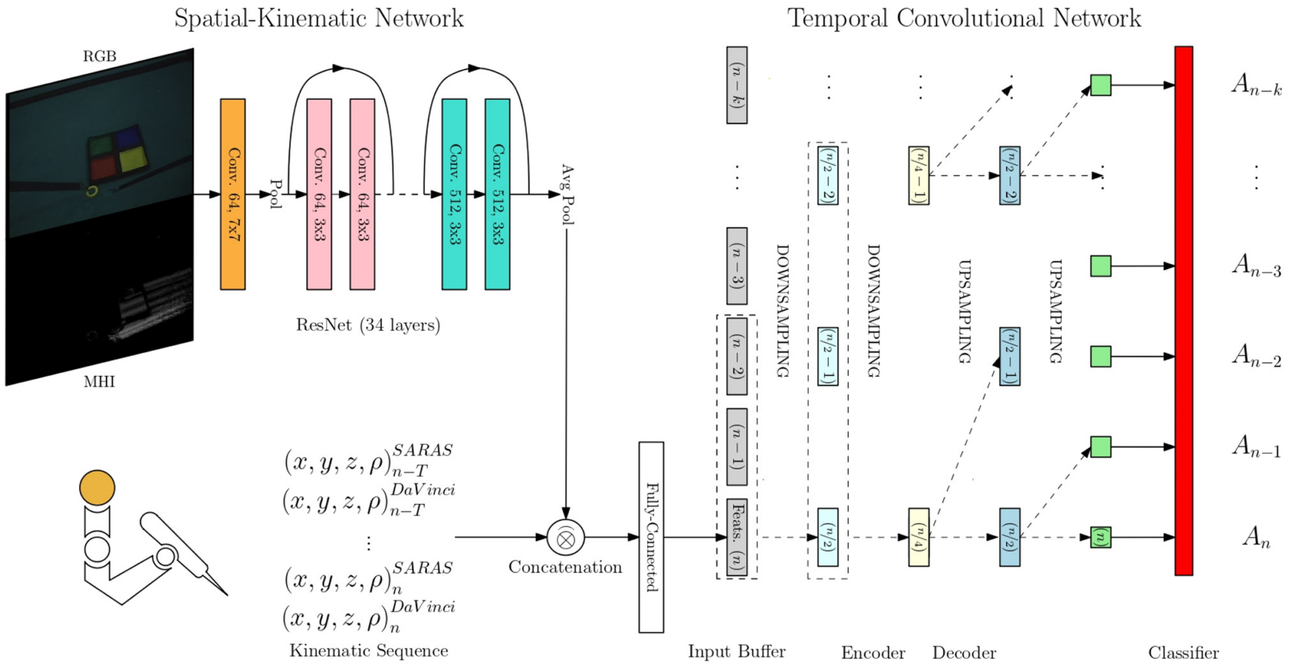


Fig. 3. Neural network schema for action segmentation: the RGB and MHI images are processed simultaneously as a 4-channel enhanced frame.

multi-modal learning over the data available during a robotic minimally-invasive surgical operation. It outperforms many state-of-the-art solutions for both spatial, i.e., the instantaneous description of the scene, and temporal information analysis.

A. Neural Network Specifications

The resulting neural network architecture (Figure 3) is composed of two sub-networks: the *Spatial-Kinematic Network*, which produces high-level features by processing image and kinematic data, and the *Temporal Convolutional Network* [19], which filters such features temporally over a sliding window to stabilize their changes over time.

The backbone of the Spatial Kinematic Network structure is the Deep Residual Network (*ResNet* [20]) with 34 layers. Its task is to process each image taken from the endoscope (in this case, the left image of the stereo camera assembly) at a rate of 10 frames per second to produce meaningful features. Additionally, it represents one of the few structures capable of scaling according to the data, i.e., its depth can be easily increased or decreased depending on the scene complexity without suffering from model overfitting during training. Its structure is composed by a cascade of convolutional filters increasing in number layer after layer; the residual paths allow the gradient not to vanish during training, which would decrease its effectiveness. The kernel size $(3, 3)$ is maintained throughout all layers to improve feature detection at different scales.

To further enhance the capabilities of *ResNet* for the specific problem of action segmentation, we introduced:

- an additional image channel called the *Motion History Image* (MHI) [21], [22], implemented as a decay factor that weights more recent and older grayscale frames over a temporal window T.
- a sequence of kinematics position, also with duration T, of the end effector for both the SARAS and daVinci arms including the closing percentage of the graspers at the end effector.

The features computed by the enhanced *ResNet* are concatenated to the temporal sequence of kinematic positions to generate an expanded feature vector. This concatenation is performed to balance the information produced by the spatial and kinematic sides. The output for *ResNet-34* is constrained by its own architecture to 512 features. Conversely, the kinematic sequence, which is a succession of normalized Cartesian coordinates and opening percentage of the tool for both arms, is restricted only by the operative frequency of the robots. To generate 512 data points and balance the spatial output, the latter has been, indirectly, set to 6.4 Hz to match the temporal window T and the video acquisition frequency of 10 Hz. The combination of image and kinematic information allows the network to better discriminate actions that appear too similar in either the image data or the relative motions to be classified correctly.

All the feature tensors computed by the spatial-kinematic network in real-time are pushed into a sliding window buffer containing up to 100 samples (10 s) to be processed within the Temporal Convolutional Network. The buffer is designed not to interrupt the training of the neural network since it allows to maintain the gradient needed for the backpropagation end-to-end, i.e., from the labeling at the end of the Temporal Convolutional Network to the input sequence of the Spatial-Kinematic Network. There are multiple benefits in the use of the temporal network:

1) it stabilizes the output relative to input changes, which has a considerable impact for online use;
2) it increases robustness to the segmentation of actions of different duration;

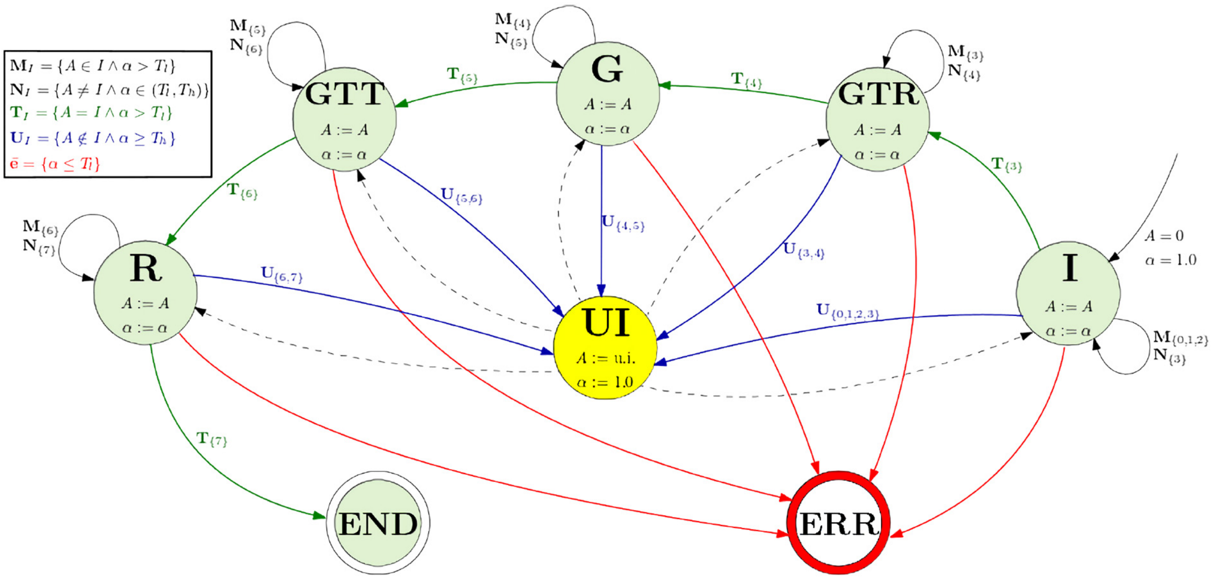

$\mathbf{M}_I = \{A \in I \wedge \alpha > T_l\}$
$\mathbf{N}_I = \{A \neq I \wedge \alpha \in (T_l, T_h)\}$
$\mathbf{T}_I = \{A = I \wedge \alpha > T_l\}$
$\mathbf{U}_I = \{A \notin I \wedge \alpha \geq T_h\}$
$\bar{e} = \{\alpha \leq T_l\}$
$\mathbf{M}_{\{5\}}$ $\mathbf{N}_{\{6\}}$
GTT
$A := A$
$\alpha := \alpha$
$\mathbf{T}_{\{5\}}$
$\mathbf{M}_{\{4\}}$ $\mathbf{N}_{\{5\}}$
G
$A := A$
$\alpha := \alpha$
$\mathbf{T}_{\{4\}}$
GTR
$A := A$
$\alpha := \alpha$
$\mathbf{M}_{\{3\}}$ $\mathbf{N}_{\{4\}}$
$\mathbf{T}_{\{3\}}$
$\mathbf{T}_{\{6\}}$
$\mathbf{U}_{\{5,6\}}$
$\mathbf{U}_{\{4,5\}}$
$\mathbf{U}_{\{3,4\}}$
$\mathbf{M}_{\{6\}}$ $\mathbf{N}_{\{7\}}$
R
$A := A$
$\alpha := \alpha$
$\mathbf{U}_{\{6,7\}}$
UI
$A :=$ u.i.
$\alpha := 1.0$
$\mathbf{U}_{\{0,1,2,3\}}$
$A = 0$
$\alpha = 1.0$
I
$A := A$
$\alpha := \alpha$
$\mathbf{M}_{\{0,1,2\}}$ $\mathbf{N}_{\{3\}}$
$\mathbf{T}_{\{7\}}$
END
ERR

Fig. 4. Scheme of the hybrid automaton supervisor.

3) it allows to obtain a prediction horizon by simply shifting the temporal output sequence during training.

Regarding the second point, the issue with non-balanced labelling, which occurs with actions of different duration across the dataset, could still have an impact on the Spatial-Kinematic network side as it risks overfitting over replicated data. However, the use of both kinematic and motion history images naturally differentiates among the input data due to the different velocities of the instruments. Additionally, a simple solution to respond to this issue is to pre-train the Spatial-Kinematic network with scaling coefficients applied to the labels to improve the single-shot recognition capabilities. Regarding the third point, the prediction results were not used explicitly in this work, yet they demonstrate the *shift-invariance* [23] capability of this network architecture and represent another proof of how the overall system is optimized for processing streaming data. This property also reduces sensitivity to the relative lengths of tasks, which can be observed in the different total temporal lengths of the offline and online tests, with the former being less than half the duration of the latter. This network operates following the *autoencoder* principle of filtering the least relevant information over time by size reduction and expansion. As the name suggests, it employs temporal convolution with a kernel size of $k_t = 60$ and stride $s = 2$ to reduce halve the tensor length for each level in the downsampling (for this task, only one level has been found necessary); upsampling operates in a opposite manner to restore the temporal sequence length. Finally, the classifier adopts a *softmax* operator to fit the probability density function for a categorical distribution. The loss function computes the *categorical cross-entropy* between the predicted ($\hat{y}$) and expected (y) values

$$L(y, \hat{y}) = -\sum_{i=0}^{k} \sum_{j=0}^{C} \big(y_{ij} \log(\hat{y}_{ij})\big) \quad (1)$$

with k and C being the temporal length for the current batch and the number of classes, respectively.

IV. Supervisory Control

For each iteration, after the action segmentation module estimates the current action $\bar{A}$ and confidence level $\bar{\alpha}$, the next task to be performed by the autonomous robot, i.e., the next goal position x_g and confidence level $\alpha(k)$ (as the non-modified output of the neural network at time sample k) (Fig. 1), is determined by a *supervisory controller*.

For the scenario described in Section II, this controller can be implemented as the *hybrid automaton* shown in Figure 4. It differentiates from a classic *Finite-State Machine* from its dependency on the time-varying variables A and $\bar{\alpha}$. We define three distinct thresholds that control the next-state function of the automaton. The first one is the lower threshold T_l, which represents the minimum value for trustworthiness below which the network is producing defective results (i.e., below the random extraction probability, approximately 12% for 8 actions). The second is the higher threshold T_h, which is the minimum level of confidence to discriminate among actions; it can be empirically set to 85% since it relates to the output of the *softmax* layer used for classification learning. Finally, the M_{tol} which discriminates over the amount of time the segmentation output remains within the two confidence thresholds ($T_l \leq \alpha(k) \leq T_h$); it has been introduced to avoid having the classification stuck in uncertainty and it has been calibrated on the time-steps required to complete the grasping action. A nominal execution of the task would see the neural network producing confidences over T_h and the next-state function using only the segmented action A to trigger a transition; a non-nominal execution would see a confidence profile that rises and drops over such threshold, thus requiring additional supervision to operate safely. The threshold T_l acts as a safety switch that indicates a computation or communication failure within the system since the neural network cannot produce values lower than the random extraction chance by design.

Guided by these thresholds, the automaton presents five states in which the SARAS robot acts autonomously.

I **Idle**, the initial state in which the system needs to remain until the detected action corresponds to tasks performed by the daVinci arm (i.e., A01-A02-A03);

GTR **Go To Ring**, when the fourth action A04 is detected, the supervisor directs the SARAS arm to move towards the ring by changing the goal position x_g;

G **Grasp**, corresponding to the A05 action, the robot is required to grasp the ring (direct control over the graspers);

GTT **Go To Target**: once the robot arm has grasped the ring, it needs to reach the delivery target as defined by action A06;

R **Release**: as soon as the target is reached and the action segmentation module detects the releasing action (i.e., A07), the supervisor orders the SARAS arm to release the ring.

Three additional control states are necessary to fulfill the description. The **End** state follows the Release state and signals the SARAS arm that it can move away from the target: this is identified with action A08. From each state, the next state is described by **ERR** (Error) whenever $\bar{\alpha} \leq T_l$. Finally, the state **UI** (user input) acts as a safeguard measure to ensure that complete control over the task is given to the surgeon whenever the condition for the maximum tolerance time is met (M_{tol}): the system will stop all activities and the surgeon is required to manually input the action to be executed next whenever the confidence level remains below the threshold T_h. The authors remark that, since the implemented supervisory control model is clearly deterministic in its formulation, the requirement of user input in the event of task failure (for instance, the token slipping from the grasper) is the most conservative and safest approach. The error state is considered as a last resort only for catastrophic system failures, such as software or hardware failures.

V. Model Predictive Control

The requirement of working in restricted environments, such as the abdomen of a patient in a laparoscopic setup, alongside human-operated tools bearing hard to predict motions leads to the implementation of reactive control methods to guarantee safety [24]. Specifically, MPC-based control methods allow formulating constraints for the robot motion planning that can consider limitations forced by both the environment and the physical characteristics of the robotic manipulator that needs to interact with it [25].

A. Constraints Formulation

As the controller is intended to maximize both the performance and the safety of the autonomous arm within the operative scenario, it was designed to incorporate both:

1) a collision avoidance formulation;
2) a velocity modulation based on the uncertainty of the action segmentation module.

For the first point, given the mechanical structure of laparoscopic tools, the constraint is formulated as a minimum of the distance between capsule-like bounding shapes that approximate the laparoscopy tools. Considering the two tools a and b, respectively the autonomous and teleoperated arms, we can consider the constraint

$$d_a^b(k) = d_{ax(a)}^{ax(b)}(k) - r_i - r_j > d_s \tag{2}$$

assuming $d_{ax(a)}^{ax(b)}$ being the distance between the instruments when collapsed into a line, r_i, r_j the corresponding radii of the respective tools, and d_s the safety distance. The velocity constraint follows the simple principle of modulating the motion following the certainty profile over the action to be executed. This is computed by the *action segmentation* module as $\bar{\alpha}$ at each discrete time. Let $u^{\max}$ be the maximum allowable velocity for the tool, the velocity modulation can be simply expressed as

$$|\bar{u}| \leq \alpha(k) u^{\max} \tag{3}$$

taking into account any desired maximum tool velocity.

B. Control Model

The robot Cartesian kinematic model is expressed by the simple motion of a material point in space located at the end effector, i.e., a discrete time-domain integrator

$$X(k+1) = X(k) + Bu(k) \tag{4}$$

where $X = [x, y, z, \theta] \in \mathbb{R}^4$ is the state vector containing the coordinates and the rotation of the tool, and $B = diag(\Delta T) \in \mathbb{R}^{4\times4}$ is the input matrix integrating the input velocity u on the discrete-time domain $t = k\Delta T, k \in \mathbb{Z}$, with ΔT the sampling time. This system formulation is sufficient to move the end effector of the robot given the intrinsically limited dynamics of the slow-moving tool; the motion towards the goal always follows the shortest line with the current position. However, this does not guarantee that the resulting motion is feasible for all the goal positions with the presence of obstacles along the path and the remote center-of-motion constraint. Therefore, a geometrical solution has been developed to provide the MPC with a real-time sequence of waypoints towards the goal [26]. This solution exploits the fact that a laparoscopy tool always presents an obstacle-free motion along the instrument axis towards the insertion trocar. The algorithm operates in two steps: (1) it samples uniformly a circle centered around the obstacle, and (2) it selects the point which is closest to the trocar. Afterwards, this point is projected onto the plane of the motion to find the nearest waypoint to avoid the obstacle. Figure 5 shows a solution found by this algorithm to generate a waypoint around the obstacle.

This simple solution fails only whenever the required goal position is non-approachable *a priori*.

C. Optimization

Given the foregoing requirements and constraints, we evaluate the optimal control problem over a finite temporal horizon

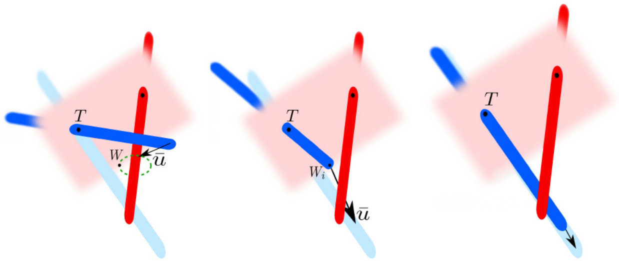


Fig. 5. Tool position (blue capsule), desired tool position (light-blue capsule) and obstacle (red capsule). Legend: *T* indicates the *trocar* entry point; *W* is the waypoint computed over the circumference around the obstacle (first image to the left, green dashed line); $\bar{u}$ is the new velocity vector pointing to the waypoint.

of length p and generate a sequence of velocities $\hat{\mathbf{u}}^*(k+\cdot)$ by minimizing the Euclidean distance between the target and the predicted states,

$$\begin{aligned} \hat{\mathbf{u}}^* = \arg\min_{\hat{u}} \; & \sum_{i=0}^{p-1} |x_w(k) - \hat{x}(k+i)| \\ s.t. \; & \hat{x}(k+i+1) = \hat{x}(k+i) + B\hat{u}(k+i) \\ & |\hat{u}(k+i)| \leq \alpha(k) u^{\max} \\ & d_a^b(k) \geq d_s. \\ & \hat{x}(k) = x(k) \end{aligned} \tag{5}$$

in which p indicates the length of the prediction horizon (in discrete steps), x_w is the waypoint state (x_g whenever this state corresponds to the final goal position) as evaluated in (4), and $\hat{x}(k+i), \hat{u}(k+i)$ are the state and the velocity predicted i steps ahead within the future horizon, respectively. The obstacles and the action segmentation confidence level are both considered frozen over all $|p|$ time steps. By repeating the optimization process at every time step, the result is the optimal control velocity for the current discrete time $u(k) = \hat{\mathbf{u}}^*(k+0)$ that satisfies the requirements and adapts to the environment containing human-controlled tools as moving obstacles.

VI. Experiments

A. Neural Network Training

The neural network for action segmentation has been trained on a customized dataset of videos acquired using the setup shown in Figure 2. When acquiring training data, both the daVinci and SARAS arms were teleoperated by two operators. Videos are recorded using the left camera of a stereo endoscope mounted on a robotic arm with the poses of both robots synchronized to each frame via ROS [27]. In total, 15 videos of approximately 200 frames each at 10 frames per second have been taken, all representing the same cooperative task, with the corresponding ground truth labelling. Each action lasted for an average of 1.04s to 6.24s with a standard deviation between 30% and 50% of their lengths. This statistics indicates that the trained model can generalize over high variances in duration. To facilitate the training phase, the parameters have been initialized with weights from the *ImageNet* competition [28]. We adopted a data acquisition protocol in the training set to improve the robustness that involves varying, for each acquisition session,

- the image acquisition perspective through the endoscope camera, with orientations kept within a 20° cone approximately,
- the lighting of the scene, by turning on and off both ambient and endoscope illumination,
- the position of the objects.

B. Ablation Studies

To better understand the net contribution of each term included in the proposed network architecture, called *EdSkResNet* shown in Figure 3, we tested several networks by turning on and off single parts of the architecture to identify each specific contribution to the overall result. The sub-networks are:

- *ResNet*, the standard RGB-only ResNet34 image classification network [20];
- *sResNet*, the ResNet34 computed over the RGB + MHI enhanced frames (which is similar to [22]);
- *skResNet*, the sResNet with the addition of kinematic sequences.

We included a baseline result, identified by the name *kClass*, which is a simple kinematics classifier composed of two fully-connected layers (a similar structure has been tested also in [29] for the JIGSAWS dataset);

The results of each network have been evaluated using three statistical indices related to the segmentation reliability:

- the *Accuracy Score*, computed as the percentage of correctly labelled samples relative to the ground truth,

$$acc = \frac{\#\text{ true samples}}{\#\text{ total samples}}$$

- the *Edit Score*, i.e., the normalized Levenshtein distance [30] between the longest of two strings $(s, \hat{s})$, thus it rewards the capability of the network to produce the correct sequence of actions; the distance is computed as

$$\mathrm{L}_{s,\hat{s}}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} \mathrm{L}_{s,\hat{s}}(i,j-1)+1 \\ \mathrm{L}_{s,\hat{s}}(i-1,j)+1 \\ \mathrm{L}_{s,\hat{s}}(i-1,j-1)+\mathbb{1}(s_i \neq \hat{s}_j) \end{cases} & \text{otherwise} \end{cases}$$

with $\mathbb{1}$ being the indicator function;

- the F_1 *Score* is the harmonic mean of the *precision* (which is the ratio between correct positive results and totally positive results) and *recall* (that is the number of correct positive results divided by the number of samples that should have been identified as positive). It is calculated as

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Testing was conducted following a Leave One Sample Out (*LOSO*) cross-validation approach for every trial to be trained over full sequences of actions. The median neural network model has been maintained to improve generalization in online usage where conditions can differ relative to the acquired data. The optimization algorithm for training is *Adam* [31] with a

TABLE I
ABLATION STUDIES RESULTS OF THE MEDIAN MODEL FROM THE LOSO EVALUATION (%)

Network	Accuracy	Edit Score	F_1
kClass	77.90	94.12	78.68
ResNet	83.98	76.19	83.85
sResNet	77.90	84.21	77.41
skResNet	90.05	100.00	90.25
EdSkResNet	93.37	100.00	93.32

validation strategy that saves the weights of the model with priority given to increments in edit score over increments in accuracy and reloads these weights for the following iterations. The best results have been obtained using a 2.0 seconds history time window for both the kinematic position trace and the MHI. Table I reports the median model's results for each score and network topology when segmenting at the latest timestamp (without a prediction horizon). The average percentage values for Accuracy, Edit Score, and F_1 are, respectively, 92.93 ± 1.53, 96.86 ± 3.63, and 92.21 ± 3.02. As expected, the scores confirm the assumption that the combined contribution of video and kinematic data overcome the limitations of either when they are used separately, with the *skResNet* gaining over both the simple kinematic classifier and the enhanced *sResNet*. Finally, the introduction of the temporal convolutional filter provides

1) an additional increase in recognition, mainly over the accuracy score since the edit score was already maximized by the *skResNet* network alone;
2) increased continuity and stability in recognition when used online for controlling the robot, as shown in Section VI.

The scores presented in Table I are better visualized in Figure 6. It shows the sequence of actions as color boxes encoded following the convention in Section II-B: most notably, the segmentation around the critical phase changes, indicated in the figure by the black dashed vertical lines, is closer in timing to the ground truth, hence the improved accuracy score obtained in training. The temporally-filtered model produces, therefore, increasingly stable results that are more suitable to be used as an online soft-sensor.

Additional information can be extracted by looking at the confusion matrix for the same results, presented in Figure 7. The most uncertain actions can be identified as A05, A06, and A07. The relative error clearly falls within the respective temporally-adjacent actions, with A05 being confused in all occurrences with A04. This will be more evident in the real-time segmentation of the task, as presented in Section VI-C, Figure 8. Thanks to the buffering nature of the Temporal Convolutional Filter, it is possible to introduce a look-ahead action prediction. This is not a requirement for the task at hand, but it proves how the temporal convolution reacts to being trained with time-shifted labels. The results show an expected decrease in both accuracy and edit score as the horizon is pushed further; nevertheless, with a prediction horizon of 1.0 *s* the overall segmentation quality remains acceptable according to both metrics (as shown in Table II and Figure 6).

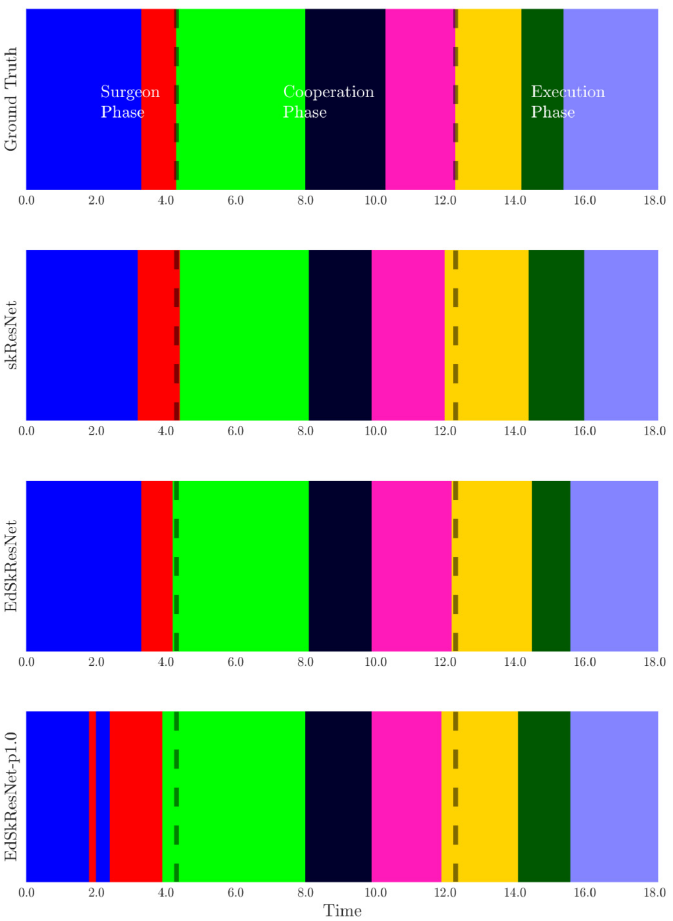


Fig. 6. Segmentation graphs for kernel size $k_t = 60$ performed on RGB + MHI enhanced images and kinematic data. From the top: the ground truth labelling; the results respectively without (*skResNet*) and with (*EdSkResNet*) temporal filtering. The bottom plot is the estimate via *EdSkResNet* with a look-ahead horizon of 1.0 seconds. The dashed lines separate the three main phases (Surgeon, Cooperation, and Execution) relative to the ground truth.

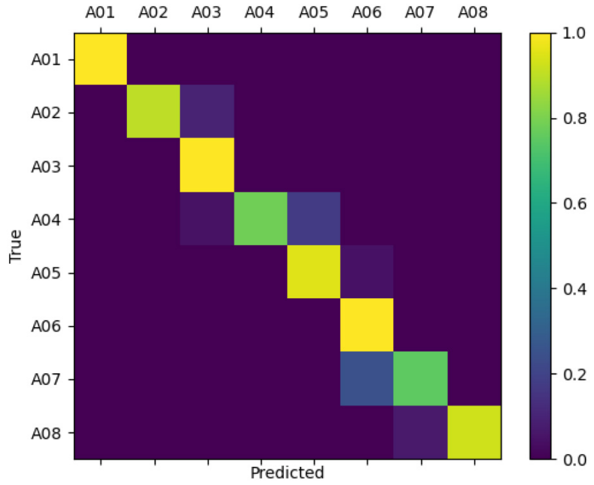


Fig. 7. Normalized confusion matrix of the median model obtained by the *Leave One Sample Out* cross-validation.

The look-ahead prediction can be used in the Model Predictive Controller to provide an estimate of the confidence level during optimization instead of maintaining a steady state condition;

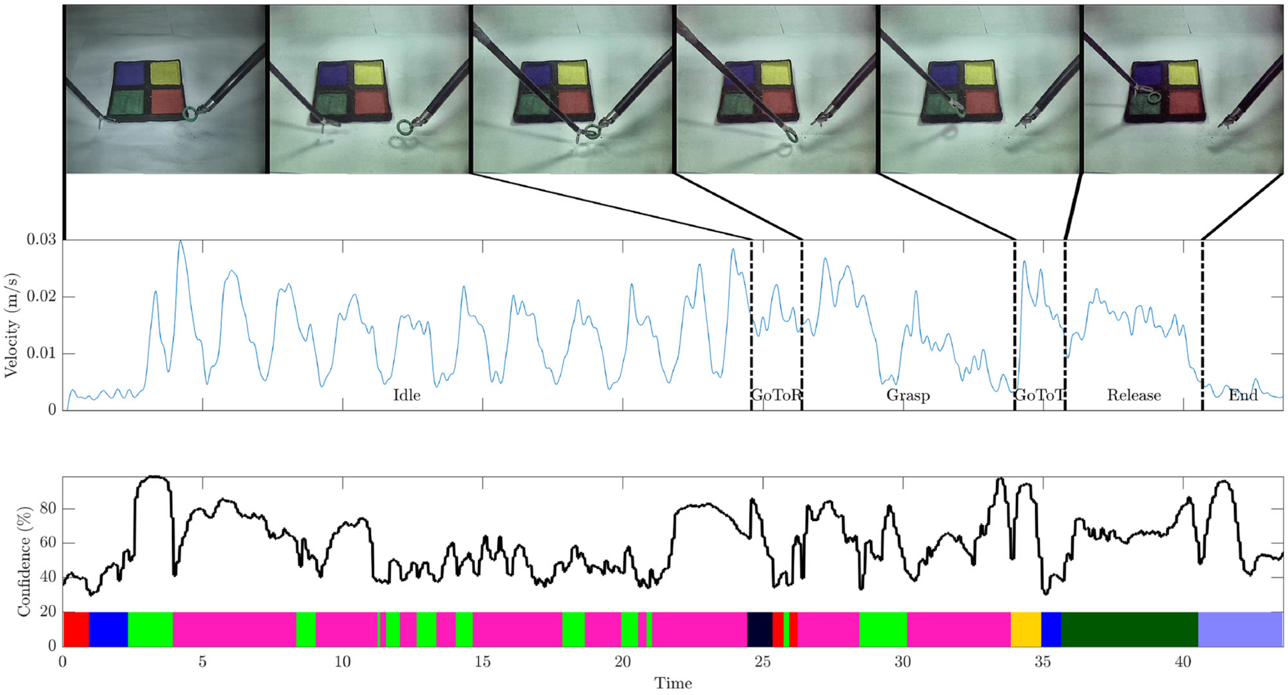


Fig. 8. Plot of an experimental task instance performed autonomously by the SARAS arm: the middle plot shows the norm of the Cartesian velocity vector with superimposed automaton states; the bottom plot shows the confidence level with the corresponding identified actions.

TABLE II
LOOK-AHEAD LABELLING ON EDSKRESNET (%)

Horizon	Accuracy	Edit Score	F$_1$
0.5 s	87.29	88.89	87.86
1.0 s	85.63	88.89	86.22

TABLE III
RESULTS FOR THE MEDIAN USER OF THE JIGSAWS SUTURING FOR THE LOUO EVALUATION (%)

Algorithm	Accuracy	Edit	F$_1$
ED-TCN [19]	81.4	83.1	87.1
Sym. Dilation w/ pooling + attn [32]	**90.1**	89.9	**92.5**
EdSkResNet [our]	81.71	**91.74**	80.08

the MPC would still evaluate the prediction horizon at each computation cycle to properly update all command velocities. To validate the neural network model over the state-of-the-art, we provide the readers with a comparison of the *EdSkResNet* over the JIGSAWS dataset [29] for two of the best performing solutions (Table III). The improvement in Edit Score indicate the prowess of the model to identify the correct sequence of actions as performed by human operators teleoperating laparoscopy instruments.

C. Robot Control Results

The combined contributions of action segmentation, supervisory controller, and model-predictive controller allow the cooperation task as presented in Section II to be completed successfully. The autonomous arm understood whenever the teleoperated arm requires the exchange to happen and delivers the ring to the required colored patch. Specifically, within the critical *cooperation phase* (actions A03, A04, and A05), the reduced level of confidence for the prediction, as presented by the confusion matrix, is correctly handled by both the supervisory control, through the correct evaluation of the "Idle" state, and the MPC, which modulates the velocity. Therefore, the uncertainty of the network does not ultimately hinder the execution of the task. The full execution can be seen in Figure 8. The top plot shows the view from the endoscope camera. The plot in the middle shows the velocity profile of the SARAS arm, computed as the magnitude of the Cartesian velocity vector, in response to the optimal input velocities produced by the MPC. The states of the automaton are superimposed over the profile. The upper limit u_{max} for the MPC has been set to $0.03\,\mathrm{m\,s^{-1}}$. The lower plot presents the confidence profile of the action segmentation module with the corresponding actions, highlighted using the same colour convention of Section II. The relationship between the automaton states and the recognised actions is evident since the robot reacts to the correct perceived user action. It is worth noting how the confidence modulation affects the maximum velocity during the robot movement in the states **GoToRing** and **GoToTarget**. During the **Idle** control state, the SARAS arm is kept in motion in order to simplify the identification of action A04 (the recognition appears uncertain between actions A03 and A05); as soon as the pick-up action is completed by the surgeon, the system recognises action A04 (at approximately second 25) and SARAS enters the state **GoToRing**, thus executing the correct action. After a few seconds of low action confidence, the task proceeds nominally with the grasp and delivery of the ring to the target, the approximate coordinates of which is located by using both cameras of the stereo endoscope to triangulate the center of the matching bounding boxes.

The test has been performed under different conditions of light and endoscope angle to verify the behavior within possible imaging conditions for laparoscopy operations with good overall performance by the system.

D. Discussions

The goal for this architecture is to perform tasks in a high-risk scenario, therefore all the uncertainties occurring in the decision-making for the task need to be reduced as much as possible. The *EdSkResNet* has been designed with the possibility of computing the spatial and temporal networks to address the issue of oversegmentation through temporal filtering only for faster fine-tuning. As presented in Table I, once empirical choices have been made for the MHI and kinematic queue length depending on the granularity of the desired actions, the *skResNet* already achieves high performance in offline action segmentation after fine-tuning from a non-correlated dataset. However, the *Spatial-Kinematic Network* acts as a single-shot detector without considering temporal correlation, which usually is a source of segmentation noise. The output stabilises with the introduction of the *Temporal Convolutional Network*, especially for online evaluations as presented in Figure 8.

It is necessary to address the difference between the offline and online testing results. During the training of the model for the neural network, the test results, which drive the choice for the final parameter set to be applied, are inevitably higher than the online results appearing over the real-time experiment. This could be attributed to the sensitivity to the user performing the task, with the SARAS arm was teleoperated during data acquisition, whereas it operated autonomously during real-time experiments. The different motions performed by the human and autonomous operators explain, at least in part, the higher relative confusion between actions A04 and A05 (Figure 7), which generates the oversegmentation that occurs at the beginning of the task (Figure 8). In fact, during this phase, the SARAS robot was purposefully kept in motion with a minimal sine wave motion to try to mimic the inevitable movements present in the data acquired through teleoperation. In addition to this occurrence, the authors address that, despite the previously mentioned precautions taken during data acquisition, the lighting condition had a remarkable effect on the task execution as the latter progressed only after dimming the light of the endoscope. The overall uncertainty, however, reduces the confidence level for any single action which makes it manageable through the supervisory controller. Finally, an initial uncertainty is present also between the actions A01 and A02 that did not occur during the offline evaluations and that manifested during the online evaluation. This can be attributed to the initialization of the daVinci kinematics that read the forceps as closed at the very beginning, a fact that confused the network and induced the swap between the actions. Indeed, the grasping angle is a strong feature for the multi-modal learning process to evaluate the effective pick up action. Under the constraints of the experimental conditions, the model predictive controller formulation and the pre-operative task knowledge, represented by the finite-state machine, provide the required level of safety and control stability to avoid damage in the event of incorrect action evaluation, thus it operates as a safe reactive cognitive system.

VII. Conclusion

In this paper we proposed a control architecture that satisfies the requirements for a semi-autonomous assistant. It integrates the necessary task determinism to operate in a surgical scenario by means of the Supervisory Controller, the motion safety offered by the velocity-constrained Model Predictive Controller formulation, and the adaptability to human task execution timings provided by the Action Segmentation (and, possibly, prediction) module. The combined efforts of these three elements managed to complete the cooperative pick-and-place task successfully without external intervention on the autonomous part. The experiments presented in this work showcase the difficulties in the adoption of neural networks to surgical scenarios, a fact that induces to think that the way forward is represented by different task-specific neural network models orchestrated by supervisors rather than a single model for entirely different classes of tasks.

As future works, the system needs to be tested on more realistic surgical scenarios with a greater amount of data to be processed by the neural network to increase robustness under all possible experimental conditions.

Acknowledgment

The authors would like to thank Johann Wigger and Sabine Hertle at Medineering GmbH for the development of the SARAS robots used in the experiments.

References

[1] M. Bonfè *et al.*, "Towards automated surgical robotics: A requirements engineering approach," in *Proc. 4th IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Rome, Italy, 2012, pp. 56–61.

[2] F. Ficuciello, G. Tamburrini, A. Arezzo, L. Villani, and B. Siciliano, "Autonomy in surgical robots and its meaningful human control," *Paladyn J. Behav. Robot.*, vol. 10, no. 1, pp. 30–43, 2019.

[3] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, "EndoNet: A deep architecture for recognition tasks on laparoscopic videos," *IEEE Trans. Med. Imag.*, vol. 36, no. 1, pp. 86–97, Jan. 2017.

[4] O. Dergachyova, X. Morandi, and P. Jannin, "Knowledge transfer for surgical activity prediction," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, pp. 1409–1417, Apr. 2018.

[5] M. J. Fard, S. Ameri, R. B. Chinnam, and R. D. Ellis, "Soft boundary approach for unsupervised gesture segmentation in robotic-assisted surgery," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 171–178, Jan. 2017.

[6] S. Krishnan *et al.*, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," *Int. J. Robot. Res.*, vol. 36, nos. 13–14, pp. 1595–1618, 2017.

[7] G.-Z. Yang *et al.*, "Medical robotics—Regulatory, ethical, and legal considerations for increasing levels of autonomy," *Sci. Robot.*, vol. 2, no. 4, 2017, Art. no. eaam8638.

[8] G. De Rossi *et al.*, "Cognitive robotic architecture for semi-autonomous execution of manipulation tasks in a surgical environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Macau, China, 2019, pp. 7827–7833.

[9] Y. Kouskoulas, D. Renshaw, A. Platzer, and P. Kazanzides, "Certifying the safe design of a virtual fixture control algorithm for a surgical robot," in *Proc. 16th Int. Conf. Hybrid Syst. Comput. Control*, 2013, pp. 263–272. [Online]. Available: http://doi.acm.org/10.1145/2461328.2461369

[10] L. Cheng, J. Fong, and M. Tavakoli, "Semi-autonomous surgical robot control for beating-heart surgery," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Vancouver, BC, Canada, 2019, pp. 1774–1781.

[11] M. Minelli, F. Ferraguti, N. Piccinelli, R. Muradore, and C. Secchi, "An energy-shared two-layer approach for multi-master-multi-slave bilateral teleoperation systems," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 423–429.

[12] B. Calli and A. M. Dollar, "Vision-based model predictive control for within-hand precision manipulation with underactuated grippers," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, 2017, pp. 2839–2845.

[13] M. Khadem, C. Rossa, R. S. Sloboda, N. Usmani, and M. Tavakoli, "Ultrasound-guided model predictive control of needle steering in biological tissue," *J. Med. Robot. Res.*, vol. 1, no. 1, 2016, Art. no. 1640007. [Online]. Available: https://doi.org/10.1142/S2424905X16400079

[14] M. Minelli *et al.*, "Integrating model predictive control and dynamic waypoints generation for motion planning in surgical scenario," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, 2020, pp. 3157–3163.

[15] N. Preda *et al.*, "A cognitive robot control architecture for autonomous execution of surgical tasks," *J. Med. Robot. Res.*, vol. 1, no. 4, 2016, Art. no. 1650008.

[16] R. Muradore *et al.*, "Development of a cognitive robotic system for simple surgical tasks," *Int. J. Adv. Robot. Syst.*, vol. 12, no. 4, p. 37, 2015.

[17] A. Leporini *et al.*, "Technical and functional validation of a teleoperated multirobots platform for minimally invasive surgery," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 2, pp. 148–156, May 2020.

[18] O. Weede, A. Bihlmaier, J. Hutzl, B. P. Müller-Stich, and H. Wörn, "Towards cognitive medical robotics in minimal invasive surgery," in *Proc. Conf. Adv. Robot.*, 2013, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/2506095.2506137

[19] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, *Temporal Convolutional Networks: A Unified Approach to Action Segmentation* (Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics 9915)). Cham, Switzerland: Springer, 2016, pp. 47–54. [Online]. Available: http://arxiv.org/abs/1611.05267

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[21] M. A. R. Ahad, *Motion History Images for Action Recognition and Understanding* (SpringerBriefs in Computer Science). London, U.K.: Springer, 2013. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-4730-5

[22] C. Lea, A. Reiter, R. Vidal, and G. D. Hager, *Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation* (Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics 9907)). Cham, Switzerland: Springer, 2016, pp. 36–52. [Online]. Available: http://link.springer.com/10.1007/978-3-319-46487-9_3

[23] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[24] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Auton. Robots*, vol. 32, no. 4, pp. 433–454, 2012.

[25] M. Cefalo, E. Magrini, and G. Oriolo, "Sensor-based task-constrained motion planning using model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 220–225, 2018.

[26] A. Sozzi, M. Bonfè, S. Farsoni, G. De Rossi, and R. Muradore, "Dynamic motion planning for autonomous assistive surgical robots," *Electronics*, vol. 8, p. 957, Aug. 2019.

[27] M. Fleder. (2012). *ROS : Robot Operating System*. [Online]. Available: http://www.ros.org

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, 2009, pp. 248–255.

[29] Y. Gao *et al.*, "JHU-ISI gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in *Proc. Model. Monitor. Comput. Assist. Interventions Workshop M2CAI*, vol. 3, 2014, p. 3.

[30] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1091–1095, Jun. 2007.

[31] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[32] J. Zhang *et al.*, "Symmetric dilated convolution for surgical gesture recognition," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, A. L. Martel *et al.*, Eds. Cham, Switzerland: Springer Int., 2020, pp. 409–418.