UNIVERSITA' DEGLI STUDI DI VERONA

DEPARTMENT OF

Computer Science

GRADUATE SCHOOL OF

Natural Sciences and Engineering

DOCTORAL PROGRAM IN

Computer Science

Cycle XXXII

Gesture Recognition and Control for Semi-Autonomous Robotic Assistant Surgeons

S.S.D. INF/01

Coordinator: Prof. Massimo Merro

Advisor: Dott. Riccardo Muradore
Co-Advisor Dott. Francesco Setti

Doctoral Student: Dott. Giacomo De Rossi

Gesture Recognition and Control for Semi-Autonomous Robotic Assistant Surgeons –
Giacomo De Rossi
PhD thesis
Verona, 22 April 2020
iris.univr.it

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 thesis explores the solutions adopted in pursuing automation in robotic minimally-invasive surgeries (R-MIS) and presents a novel 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.

Contents

1

Introduction

The current trend in robotics is to push cooperation with humans to support and improve capacity, skills, and safety whenever needed. The term *cobotics* (a portmanteau of *collaborative* and *robotics*) is being pushed within the research community to highlight all the applications in pursuit of enhancing human activities with autonomous machines. Thus, the trend is to move robots out of factories and beyond accurate but repetitive tasks towards environments shared with humans and tasks of compliant object manipulation.

To achieve real cooperation effectiveness between a human and a robot, it is necessary to operate simultaneously on the same three aspects that define full autonomy, namely *scene understanding*, *autonomous reasoning*, and *compliant control*. With respect to scene understanding, the focus verges on the comprehension of *human actions* and their future evolution. The automated analysis of human gestures is indeed a major research area in computer vision thanks to its potential to improve traditional human-machine interfaces and to create true cooperative robotic platforms that overcome the limitation of available technologies. For instance, this technology is being evaluated for usage in autonomous driving systems that require a comprehension on the intentions of other vehicles on the road. Autonomous reasoning encompasses all the strategies exploited to achieve an independent task execution from the robot, ranging from deterministic *Hybrid Automata*, where the dynamics of the task have a profound influence over the execution of pre-defined sequences of actions, to probabilistic models based on *Markovian assumptions*, up to *genetic algorithms* to achieve a more generalized artificial intelligence. Finally, the requirement of interacting with environments ranging from hard to soft contacts brings up the necessity of compliant controls, i.e. algorithms that abandon the classic position-oriented placement of robotic manipulators to optimize the safety of energy exchange among bodies both rigid and flexible.

The integration of these three components can be achieved in two main design strategies. The first is called the *engineering stack*, which develops the three capabilities separately to maintain high levels of supervision over the prowess of all parts to ensure adherence to stringent operational requirements but, inevitably, moves the difficulty over to the integration. By the definition of the required interconnections early in the design phase, it is still possible to achieve a seamless integration. The second approach is the *end-to-end* model which intends to develop a single unit capable of direct analysis of sensory inputs to produce the required control output to the robot. It intentionally masks the contributions of all components to provide the highest integration and role superposition, but it makes overtly complex the definition of fine-tuned constraints.

Looking into surgical applications, 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 System, a remote teleoperation platform for minimally-invasive surgery that does not present any automation degree and provides only video as feedback to the surgeon to ensure control stability under all circumstances. Notable exceptions available on the market are ROBODOC [53], CyberKnife [15], or NeuroMate [76], but either their operative scope is restricted to specific and well structured body portions or they are designed to operate only on rigid tissues using offline planning.

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. Regarding the specific use in a surgical scenario, many approaches have been proposed for scene recognition [74, 13, 18, 38], autonomous reasoning [23], and compliant control [21, 71, 9]. Consequently, the necessary level of interaction will push the dexterity, perception and cognition capabilities beyond the current limits of robotics applications.

1.1 Lexicon

A few lexical notes are required to proceed onwards with the document. As this work will present solutions developed to address the problem of action segmentation also outside the scope of surgical applications for laparoscopy, the use of "action" and "gesture" is considered interchangeable. Formally, by the taxonomy that will be defined in Chapter 2.2.1, a *gesture* corresponds to the concept of *motion*, whereas an *action* corresponds to a *surgéme*; in practice, the distinction between the two terminologies is based primarily on semantics rather than a formal distinction on the underlying data used to represent them.

It is also necessary to specify what this work intends with *Action Recognition* and *Action Segmentation*. In fact, the distinction between the two problems is subtle and the literature sometimes mixes the terminologies. all algorithms that produce a one-shot identification of an action instance within time series data (for example a set of videos) are classifiable as action recognition. Whereas the temporal context of the action to be identified is essential and it differentiates the process of action recognition from a static image classification, the temporal coordinates themselves, hence the beginning and the end of each action, are not relevant for the task. Conversely, action segmentation involves the continuous identification of action changes and their location in time, in addition to their classification. Action segmentation requires to solve both a *classification* and *regression* problem, therefore all methods that perform action segmentation can be used for recognition but the opposite is not always possible.

1.2 SARAS European Project

The work of this thesis is aligned to the effort pursued by the EU funded *Smart Autonomous Robotic Assistant Surgeon* (SARAS) Project (`saras-project.eu`) and shares with it both the platform and data. The goal of the project is to define the required technologies and to pursue the development of an effective robotic substitute to the assistant surgeon that currently works next to the patient within the operating room during R-MIS operations. All the instruments involved will be general-purpose products for minimally invasive surgery, like scissors, graspers, clip appliers. However, to effectively validate the SARAS concept, the project focuses on *radical prostatectomies*, i.e. the resection of the whole prostate gland in male patients with prostate cancer while preserving urinary continence and erectile function, and *partial* or *radical nefrectomies*.

The project aims at developing three increasingly complex autonomous platforms to assemble a data-driven cognitive control architecture in which the surgeon and the robots operate seamlessly together. In the first, called MULTIROBOTS-SURGERY platform (Figure 1.4), the main surgeon controls the daVinci® tools from the console, whereas the assistant surgeon teleoperates standard laparoscopic tools mounted at the end effectors of the SARAS robotic arms from a remote control station equipped with *virtual reality* and *haptic devices*. The assistant surgeon will perform the same actions as in standard robotic surgery, but this time by teleoperating the tools instead of moving them manually. The MULTIROBOTS-SURGERY platform is an example of *multi-master/multi-slave* (MMMS) bilateral teleoperation system, where two users cooperate on a shared environment by means of a telerobotic setup. This setup already improves over standard *robot-assisted radical prostatectomies* as the assistant surgeon controls a sophisticated system that emulates standard laparoscopy tools and provides force feedbakc and virtual fixtures to the user. Moreover, the platform allows to

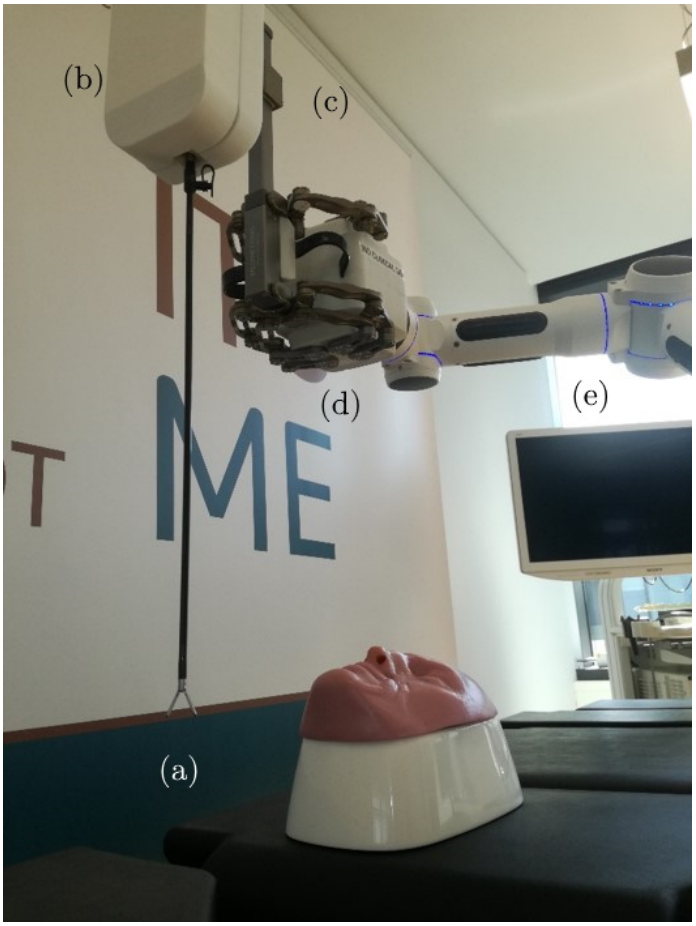

Fig. 1.1: SARAS Assistant Robot for Laparoscopy: (a) is the end effector gripper; (b) is the adapter holding the actuation of the gripper and rotation θ over the z-axis; (c) is the linear actuator for z; (d) is the x, y parallel robot; (e) is the passive positioning arm.

acquire the relevant video and kinematic data from expert operators. In the second architecture, called the SOLO-SURGERY platform (Figure 1.5) and the intended case study for the work of this thesis, the assistant surgeon will be replaced by the *cognitive control architecture* controlling the SARAS arms and adapting to the operator's actions to provide assistance. This platform will be a very sophisticated example of a shared-control system: a surgeon operates remotely a pair of robotic laparoscopic tools (*e.g.* the daVinci® Surgical Platform) and cooperates with the two novel SARAS autonomous robotic arms inside a shared environment to perform complex surgical procedures [65, 55]. This architecture, of which Figure 1.3 presents the main block components, represents a highly sophisticated example of *embodied A.I.*, i.e. a robotic platform relying on artificial intelligence technologies to comprehend both the operator and the environment. Finally, in the LAPARO-2.0 platform (Figure 1.6), the only robot operating next to the patient will be the SARAS assistant robot as the surgeon handles standard laparoscopy instruments instead of robotic tools. The removal of the robot from the operator's side increases the challenges of controlling the collaborative robot as it introduces the requirement of visual tracking for all the instruments that, otherwise, can be achieved by exploiting the robots' kinematics.

A customized robot has been developed by *Medineering*™ GmbH for the SARAS project to mimic the dynamics of the laparoscopy tools being driven by the assistant surgeon. The robot is sustained over the operating table by a passive positioning arm and has four actuated degrees of freedom split between a parallel robot for movements over the x, y plane, a linear actuation for the z axis, and an adapter containing an actual laparoscopy tool providing z-axis rotations θ; a motor actuates the opening and closing actions of grippers or scissors

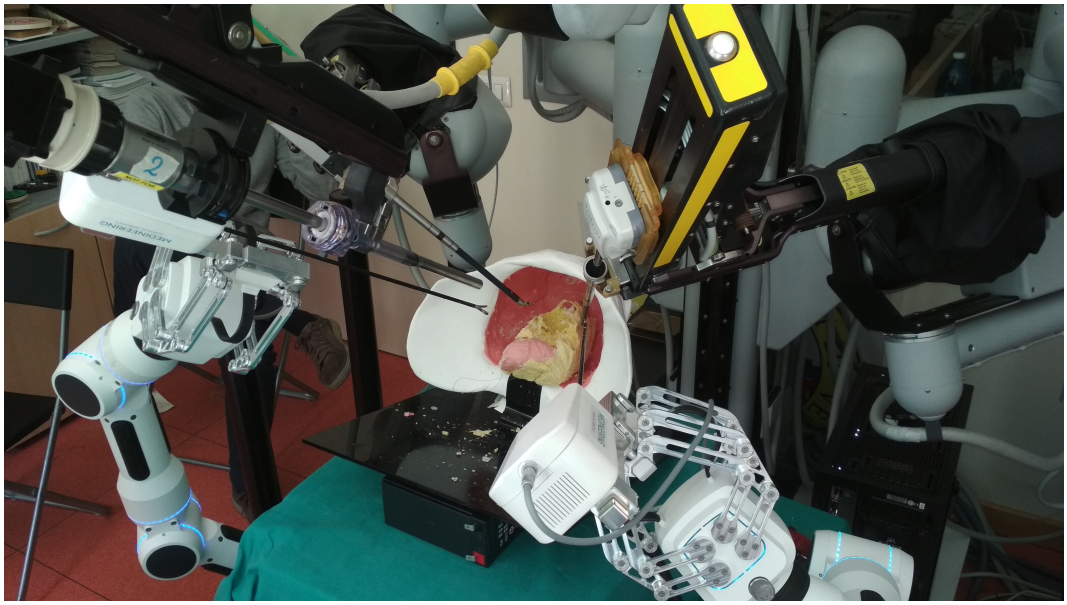

Fig. 1.2: SARAS Assistant Robot for Laparoscopy and DaVinci Research Kit arms after performing a radical prostatectomy on a phantom.

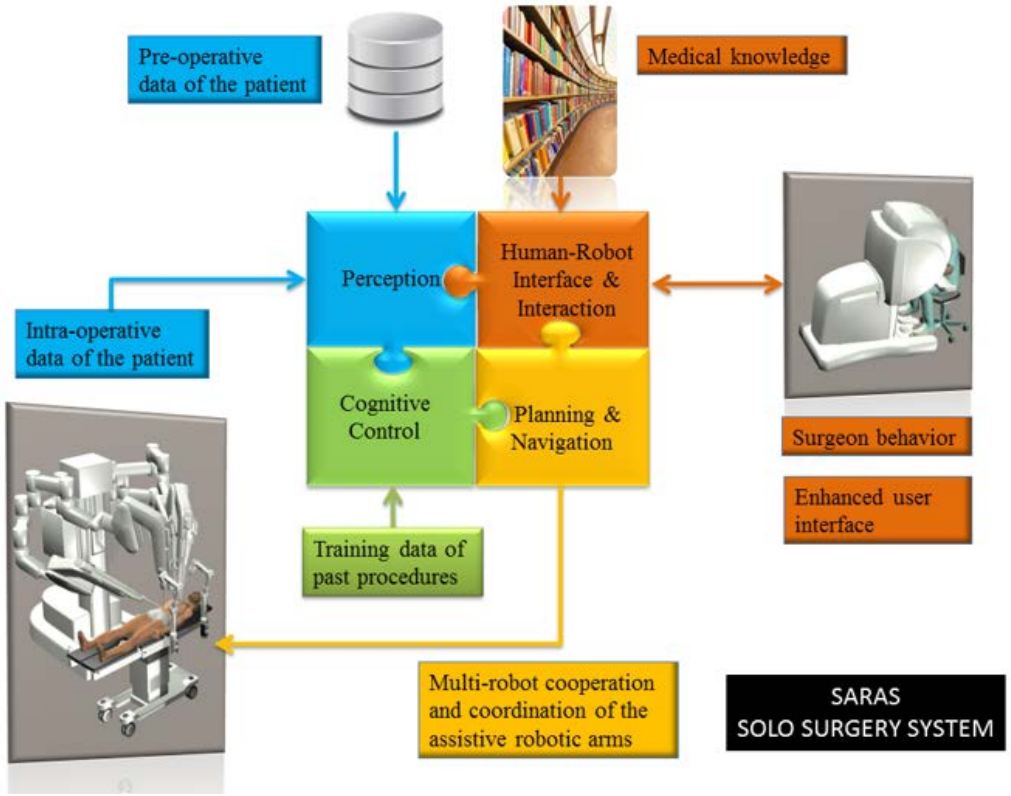

Fig. 1.3: Macro components of the general solo-surgery architecture

at the end effector. The robot is constrained by a *remote center-of-motion* (RCM) on the entry point to the patient (*trocar*) maintained by the solution of the inverse kinematic solver (i.e. via software). Figure 1.1 presents a view of the full robot arm and Figure 1.2 provides a view of an experimental setup prepared for the simulation of a radical prostatectomy in support to the DaVinci Research Kit [32] arms.

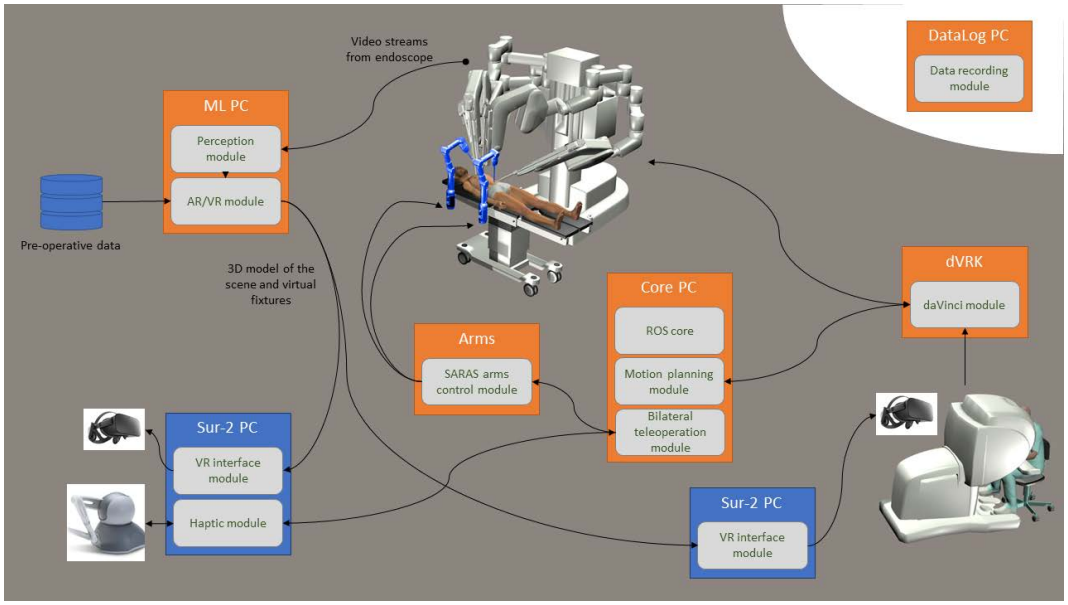

Fig. 1.4: SARAS Multirobots-Surgery platform.

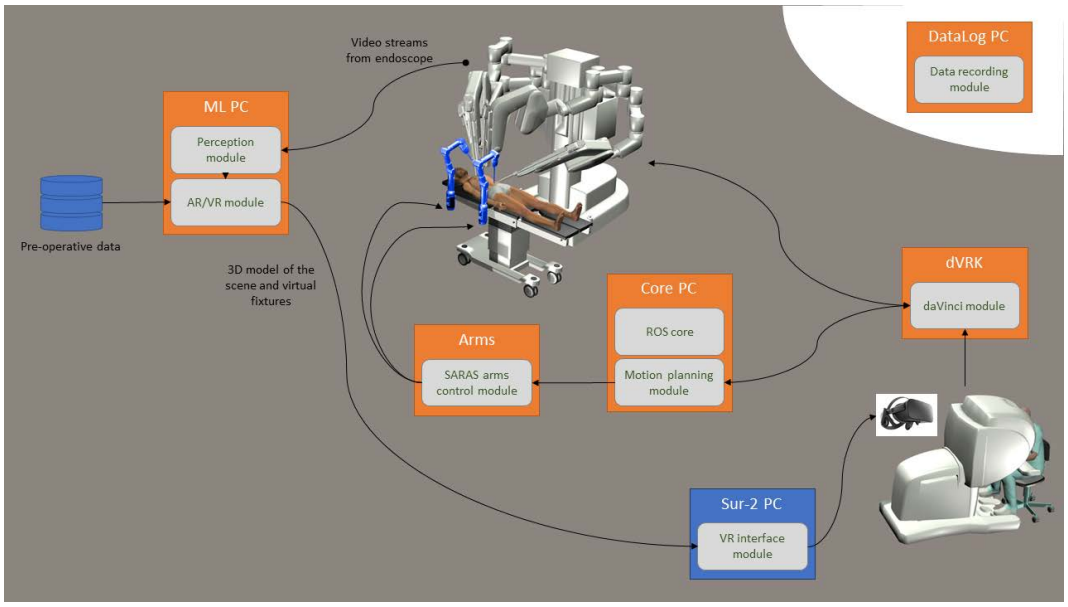

Fig. 1.5: SARAS Solo Surgery platform.

1.3 Thesis Contribution

This thesis aims at integrating an improved neural network model for online temporal segmentation and recognition of surgical gestures, called *surgémes* [62], with the control of a Robotic Minimally Invasive Surgical System to create a cognitive cooperative system capable of assisting a surgeon in manipulation tasks with laparoscopy instruments.

This thesis focuses on three aspects that are deemed fundamental for a cooperative surgical robot:

1. the *interpretation* of the surgical gesture to comprehend the intention of the surgeon at any given time (i.e. perception);
2. the *control* of surgical robots, which requires to integrate decision process and planning;
3. the *integration* with a supervisor system where the *a priori* knowledge is encoded.

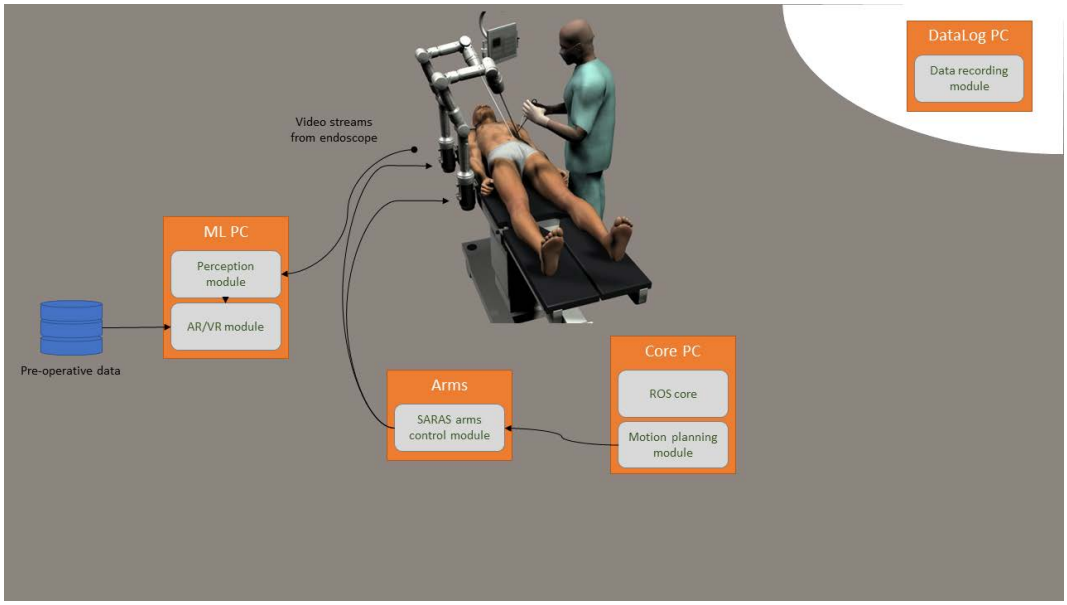

Fig. 1.6: SARAS Laparo-2.0 platform.

The first contribution can be found in the formulation and testing of a fast neural network architecture for gesture segmentation using multi-modal learning through the exploitation of both kinematic and video data in an *online* manner. The proposed solution, called the *Time-Interpolated Fully-Convolutional* (*TiFC*) network, is a neural network which uses convolution strides to maintain the sampling period constant during temporal downsampling, which avoids the use of more common max pooling operators, and an interpolation function to reconstruct the temporal dimension. Its goal is to improve over most commonly used recurrent networks for signal segmentation, like those presented in [73, 69, 16], by following novel approaches in signal regression and analysis as it implements a Convolutional Neural Network also on the temporal domain. Additionally, it intends to overcome the computational inefficiency of *autoencoder* structures by avoiding convolutions in the *decoder* phase. This structural change reduces the required training time given the intrinsically parallel nature of independent convolution filters when compared to recurrent neurons, while improving overall performance [3]. This neural architecture takes as input a sequence of temporal features related to an action performed by a human, or by a robot manipulator, acquired using a single camera. It proposes a variation on the *Encoder-Decoder Temporal Convolution Network* (*ED-TCN*) topology presented in [43], in which an encoder-decoder neural network is trained to operate over a sampled time axis of video features. As a new feature of the proposed network, after filtering the resulting signals are interpolated in the time domain to match the initial signal length. This is achieved via a single linear interpolation step, rather than by using stacks of deconvolution filters, consisting in zero-order hold operators in combination with standard convolutions. This effectively halves the number of required parameters with basically no detrimental impact on the final segmentation results, thus representing an efficient strategy for temporal action segmentation. Linear interpolation is simple and efficient to compute, but presents well-known drawbacks in terms of both precision and differentiability at the boundary. Nevertheless, the results improve over those of traditional encoder-decoder architectures and are comparable to the current state-of-the-art. This paves the way to further improvements by implementing more advanced classes of interpolating functions such as basis splines. An additional benefit in lowering the number of required convolution filters is the reduced sensitivity of the network to variations in temporal kernel size, which represents the most delicate choice of parameters for temporal convolution network structures. This hyperparameter relaxation represents an advantage when implemented within a real-time controller and in scenarios that could involve a stretched temporal execution of actions relative to those available in the training set.

The second contribution is the development of a control system that employs the information provided by the action segmentation to correctly time the autonomous part of the task execution to the direction given by the surgeon while avoiding interference to the surgeon's job through undesired collisions between the tools and operating at a constrained speed whenever the *confidence* over the interpreted action being performed is low to avoid both possible execution deadlocks due to low confidence and an excessive velocity towards wrong targets. This has been achieved by the development of both a *hybrid automaton* supervisor to define the appropriate reactions to the actions of the human operator and a *Model Predictive Controller* (MPC) that provide the optimal control velocities as the result of a bounded optimization process.

In summary, the work presented in this thesis presents a novel cognitive control system for assistant robots in minimally-invasive surgery that improves the current state-of-the-art by

1. the development of an improved gesture recognition algorithm based on convolutional neural networks that can operate *online* and is less susceptible to changes in the speed of execution of any performed action;
2. the adoption of a *Model Predictive Control* to define constraints on the movement of the robots. This allowed to avoid potentially harmful collisions based on geometrical information of the workspace and to bind the velocity to the cognitive information of the gesture recognition.

The experimental setup provided by the SARAS project allowed the entire system to be tested, at least partially, on advanced laparoscopy robotic platforms and realistic anatomical phantoms to empirically prove the effectiveness of the entire control architecture.

1.4 Thesis Outline

The organization of this document follows a logical progression towards the description and test results of the architecture developed to achieve a semi-autonomous surgical task using laparoscopy robots. At first, Chapter 2 presents the formalizations adopted to describe surgical procedures in engineering to position the work within the larger research area of *Surgical Process Modeling*. The following Chapter explores the state-of-the-art for gesture recognition and introduces one of the main contributions of this work to the field: a neural network to filter data time series to generate a temporal segmentation of actions with reduced computational footprint and improved performance. Chapter 4 describes the technologies applied to control surgical robots and introduces the Model-Predictive controller designed to overcome the limitations of control technologies available in literature and on the market regarding cooperation with humans in critical scenarios (*e.g.* laparoscopic surgery). Finally, Chapter 5 delineates the *cognitive control architecture* capable of performing semi-autonomous surgical tasks as an assistant to a primary human operator, the experimental setup, the validation task, and the results obtained by applying the foregoing technologies.

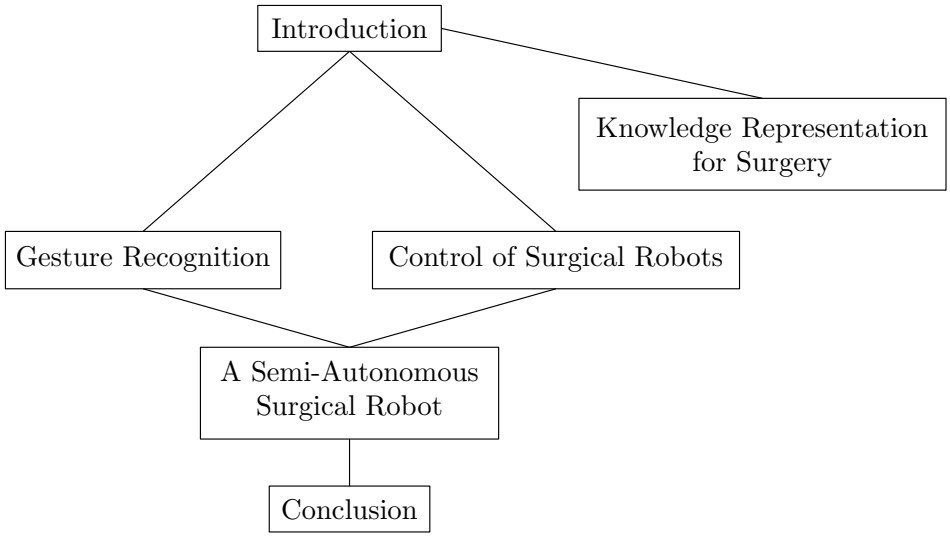

Fig. 1.7: Organigram of the Chapters.

The thesis that gesture recognition applied to the control of surgical robots is not only viable but has positive repercussions is, therefore, empirically demonstrated by both simulated qualitative experiments on the separate components that and the successful execution of a semi-autonomous pick-and-place task using both the daVinci® Surgical System and the SARAS Platform.

In the Conclusion, future developments and improvements for the architecture are discussed towards the execution of a full surgical procedure alongside a surgeon. In particular we will focus on solutions that could increase performance and safety. Figure 1.7 presents a reading organigram of the Chapters.

2

Knowledge Representation for Surgery

2.1 Introduction

The analysis of any specific model that contains the desired expressivity to conduct a surgical procedure in an autonomous manner has to be collocated within the general concept of *knowledge representation* for surgical procedures, which is the study of both data and form required to define a procedure from the pre-operative to the post-operative phases. The description of surgical procedures in literature has been defined with a variety of formalisms and taxonomies depending on the abstraction level. The necessity to find a structure in the data involved in surgical applications arises from the increased use of advanced machinery and automation in the operating room. The medical staff has to review the large amount of data that modern minimally-invasive diagnosis tools, such as 3D computerized tomography, advanced ultrasound probes, intra-luminal cameras, while future autonomous machines will need to process all the information before taking appropriate actions. Data organization and representation are especially critical to effectively connect information and formulate a corresponding *knowledge graph* [59].

This chapter introduces the formalisms available in literature regarding both knowledge organization and extraction from low-level data to contextualize the work of this thesis within the wider field of surgical knowledge representation. Specific attention will be given to the description of formalisms for *Robotic Minimally-Invasive Surgery* (R-MIS) given the focus of this thesis.

2.2 Taxonomy

A surgical procedure can be organized in a hierarchical structure depending on both the domain and the type of data involved. We regard the work in [40] that reviews the formalisms applied to surgical applications and medicine in general. The formalism laid out by the authors describes the surgical process within five domains:

- *Application*, the specific clinical application being described within the surgical field
- *Acquisition*, how and when the relevant data has been acquired;
- *Modeling*, the definition and formalization of the work domain;
- *Analysis*, the description of how data is compared, aggregated and mathematically described;
- *Validation*, the metrics adopted to compare and evaluate both data and models.

Application and *Acquisition* are sufficiently self-explanatory: for instance, a specific scenario is that of a Robot-Assisted Laparoscopy operation and the data acquired is intra-operative video from an endoscope and the synchronized kinematic measurements available for the robot being controlled by the surgeon. The other domains require a more in-depth analysis to be described.

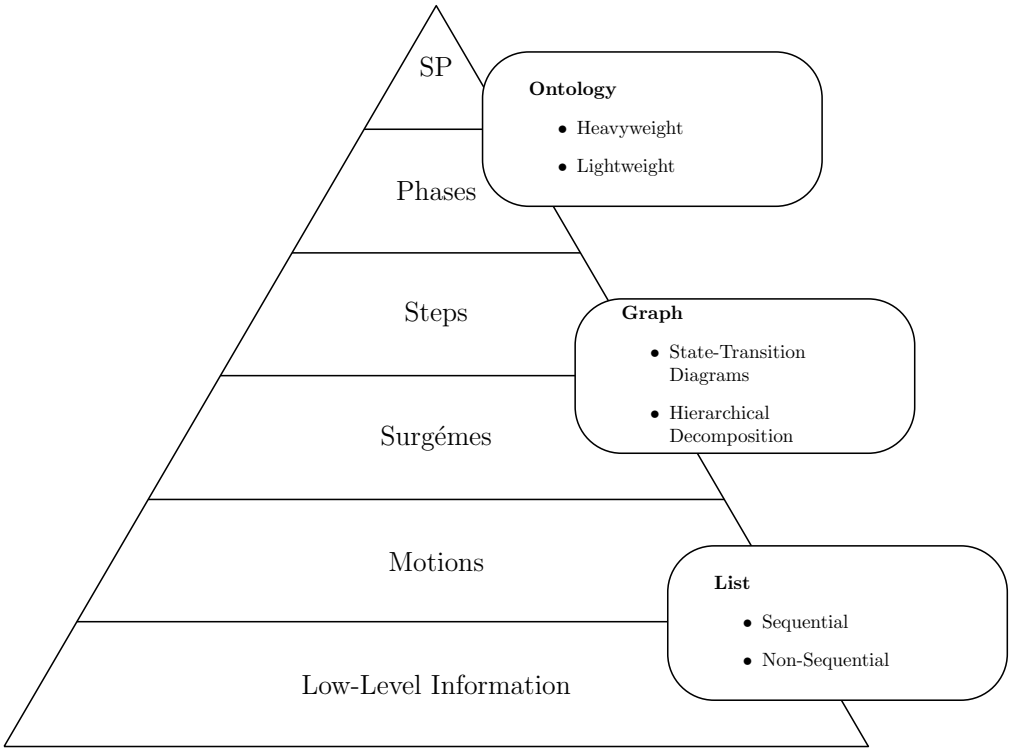

Fig. 2.1: Surgical Process Representation granularity pyramid with examples of corresponding formalisms.

2.2.1 Modeling

The modeling of a surgical procedure involves the definition of the formulation that better describes the required *granularity* of the surgical item that is being studied. In fact, a surgical procedure can be seen as a pyramid structure (Figure 2.1) based on low-level information, i.e. all the data generated by the sensors in the operating room such as videos of laparoscopy cameras, kinematic sequences of robots. Such data can be structurally organized into higher-level units that encompass increased semantics:

- *motions*, which describe single-hand tasks involving only one hand trajectories;
- *surgémes* [62], which represents a semantically well-defined surgical motion unit (Figure 2.2 show a few examples of surgémes for suturing a wound with robotic laparoscopy tools);
- *steps*, a sequence of surgémes used to achieve a surgical objective;
- *phases*, the highest level semantics representing major distinct stages for the procedure.

Depending on whether the model being defined climbs or descends the pyramid, the result is said following respectively a *bottom-up* and *top-down* modeling approach.

The subdivision is inherently mapped in the two main approaches found in modeling, i.e. *data-driven* and *model-driven*. In the former, the acquisition and processing of lower-level data is what defines the separation between surgical processes; in the latter, the separation of entities is defined *a priori* by the surgical staff and the engineers with the higher-level data as the result of the execution of such process.

2.2.2 Analysis

The analysis of a surgical process defines the technologies implemented to process data and models with the intent of automating the execution of the *knowledge processing*.

For Top-Down models, the analysis available in literature employs *formal and description logic*, *inference and workflow engines*, and the *hybrid automaton*. Among the description

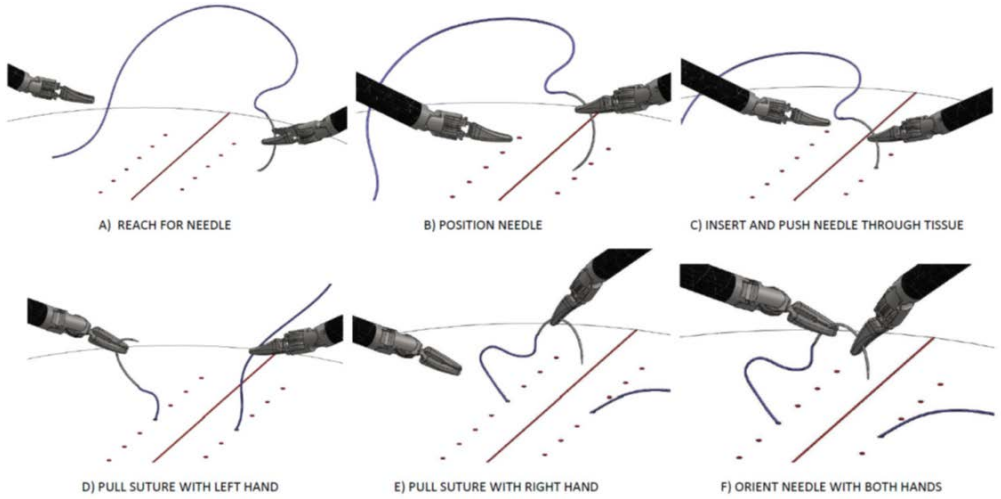

Fig. 2.2: Surgémes for a line suture of a wound using R-MIS laparoscopy tools [5].

logic formalisms, *ontologies* have been the most studied for applications to the surgical domain, for instance the *OntoSPM* [29]. Ontologies are a engineering artifacts consisting of a *vocabulary*, which models a set of real-world phenomena in a given domain, and an *explicit* and *machine-processable* specification of the intended meaning given by "*is a*" relations between the vocabulary and constraints that capture part of the semantics. Workflow Engines rely on classical logic specifications of "*and*" and "*or*" operators augmented by temporal logic to relate tasks in an executable flowchart; an example is found in the *Surgical Workflow Model* [60]. Applications of *hybrid automata* to the surgical workflow have been applied with success to robot-assisted surgical procedures as a formal verification method for the process, such as the *Hybrid Input/Output Automaton* model presented in [7].

Bottom-Up approaches exploit low-level data to construct higher-order knowledge structures: they rely on *comparison* and *aggregation* mechanisms to define relations. These mechanisms usually apply statistical and/or geometrical tools to compare and cluster data points. An example of aggregation methods is found in [48] between two surgical procedures described in the *Unified Medical Language System* (UMLS); Figure 2.3 presents an aggregation example being performed on four sub-procedures of the same surgical process.

The definition of distance metrics is clearly required to compare data for aggregation: the work in [54] proposes five similarity metrics to be applied at any level of surgical procedures: the similarity of granularity and of content, of duration (temporal), of transition, and of transition frequency. Figure 2.4 presents said metrics in a schematic format.

Bottom-Up approaches can operate aggregations and comparisons in both a *supervised* or *unsupervised* manner depending on the availability of labeling data to allow direct ground-truth comparisons which can supervise the learning process. Unsupervised algorithms combine metrics at varying levels to achieve data clusters to be quickly labeled with a semantic meaning. Examples are provided by the *Dynamic Time Warping* performed in [25], which operates on sequential lists to autonomously identify cost-based graph similarities, or the *Soft Boundary* approach to gesture segmentation [18] which provides a fuzzy-logic formulation of boundary conditions for temporal sequences. Among the supervised algorithms, *neural networks* have provided the highest performance and adaptability degree so far for featurization, classification, and regression, at the cost of a reduced control over the quality of results due to their weak theoretical foundation.

2.2.3 Validation

The validation domain directs the choice of evaluation strategy that, inevitably, alters the quality of the results for the entire surgical process. In practice, it can be performed in three different scenarios:

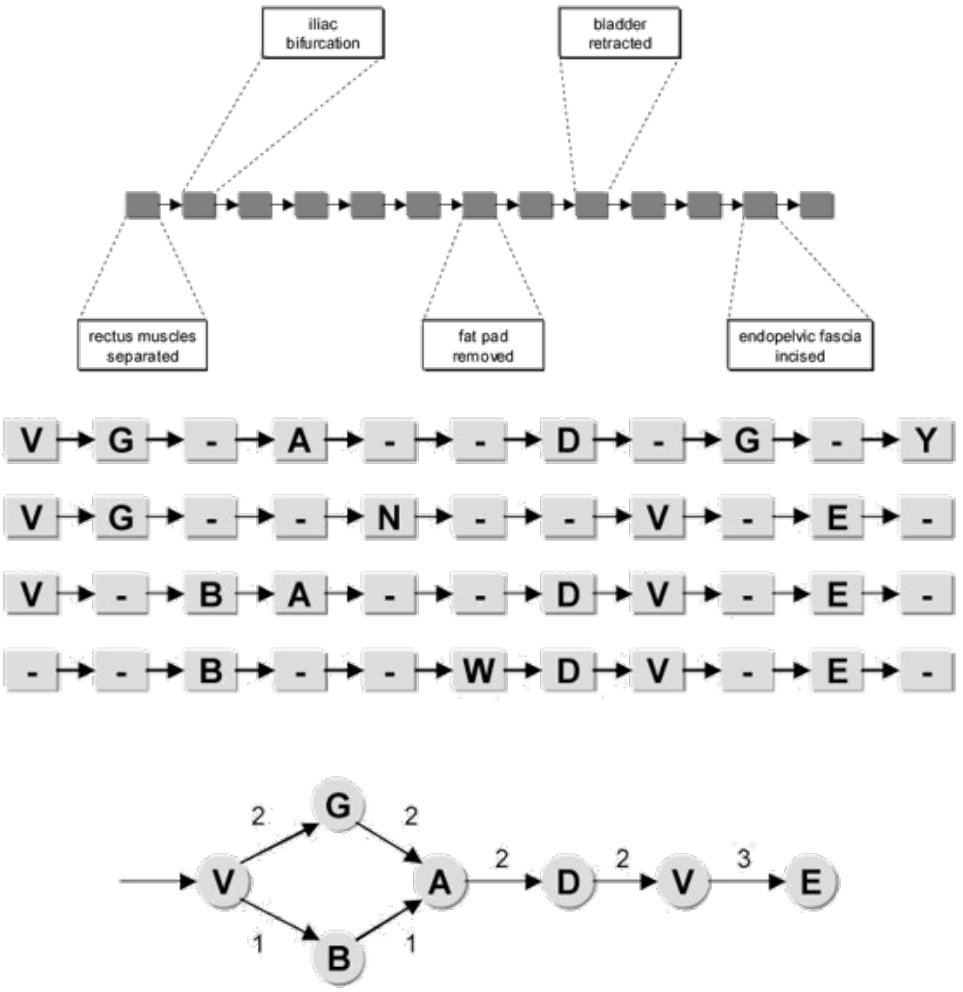

Fig. 2.3: UMLS aggregation example [48].

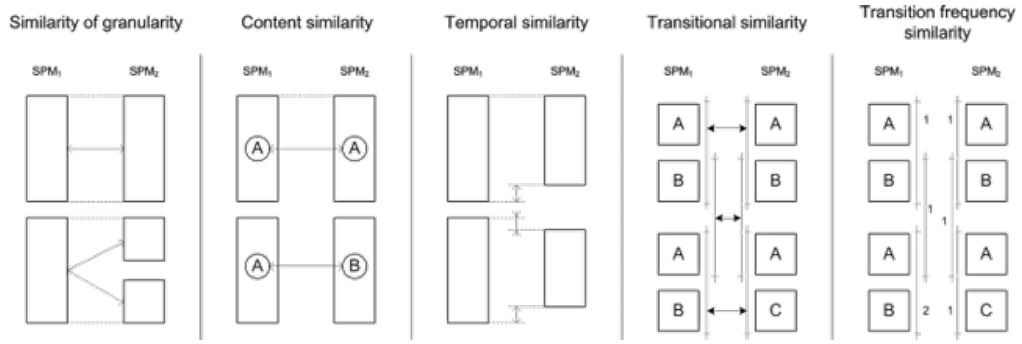

Fig. 2.4: Schematic principles of SPM similarity metrics [54].

- Computer-Simulated environments,
- Simulated Operating Rooms, working over anatomical phantoms,
- Real data acquired during surgical interventions.

The three scenarios are not mutually-exclusive: any well proven techniques has been tested under all these conditions, usually in incremental steps.

Validation has to provide a set of metrics and procedures to provide quantitative measurements of the quality achieved by SPMs. For supervised methods, the foremost validation

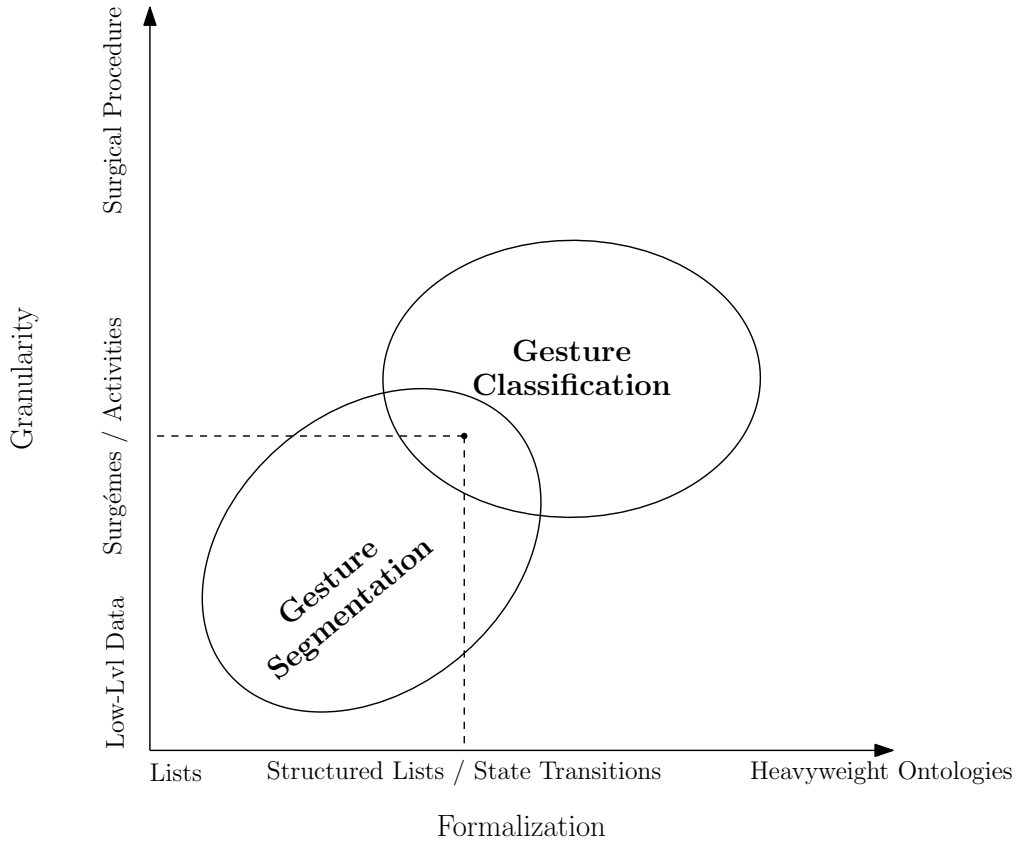

Fig. 2.5: Position of the work in this thesis according to the Surgical Process Modeling Taxonomy.

technique is represented by the *K-Fold* Cross-Validation, which breaks the labeled data into sets, usually corresponding to users, ad executes validation in a *Leave One Out* manner to certified the generalization capabilities of the tested model.

For unsupervised clustering techniques, in the absence of reference ground-truth data to compare, one of the most used metrics is the *Davies-Bouldin Index* [10]: let $\mathbf{X}$ be a n-dimensional feature vector assigned to a cluster i and let A_i indicate the *centroid* of said cluster; the DB-index is computed as

$$S_i = \left(\frac{1}{T_i} \sum_{k=1}^{T_i} |x_j - A_j|^p \right)^{\frac{1}{p}}$$

with p as the distance order (usually $p = 2$ for the Euclidean distance).

2.3 Discussions

From the discovery performed in this Chapter regarding the formalisms applied to surgical processes, it is possible to position the contribution of this thesis in the *Formalization/Granularity* space (Figure 2.5): the goal is to obtain a combined segmentation and classification of surgical gestures.

In this thesis, we combine data-driven and model-driven approaches to process lower-level data as recorded through the instruments and examine the progress of the semi-autonomous surgical task to coordinate the robot's actions. The execution of the task is controlled through the available knowledge by a model-driven *hybrid automaton*.

3

Gesture Recognition

3.1 Introduction

The ability of transferring information in a non-verbal way is a major advantage for social interaction. People can successfully understand the environment they navigate without any kind of explicit language input. This cognition ability remains a major distinction between biological beings and machines, and yet a *cobot* will be required to perform this feat to achieve full cooperation capabilities. In surgery, and in every other highly skilled job, medical teams operate together for months or even years to reach a full understanding of how each other works.

Wherever autonomous systems will be able to match (or even surpass) the innate human cognition ability is a pure speculative argument. Nevertheless, the exponential improvement in computational power in the past few decades paved the way for statistical analysis of large datasets in pursuit of autonomous pattern recognition, a specific task where machines do appear to have the edge on the average person, although still with significant limitations.

Once the capability of extracting higher level semantics from raw data has been established, the development of more and more advanced action and/or gesture recognition models has progressed rapidly. Progress has also been aided by the improvements of *artificial intelligence* algorithms dedicated to improve understanding of the task and the surrounding environment in which it is defined. The first examples of artificial intelligence algorithms have, historically, been *knowledge based* modeling approaches, *e.g.* the ontology models presented in Chapter 2, that describe the task as formal languages to discover relationships between items by means of logical inference rules.

Any algorithm that allows a machine to extrapolate patterns from data privy of any *a priori* programming falls within the research field of *machine learning*. The idea is to combine computing power and exploit geometric, probabilistic and statistical results to explore the unorganized data in search for structures that defy the capabilities of classic data analysis paradigms. The adoption of data-driven machine learning technologies for gesture recognition is driven primarily from necessity to match complex human motions in space with their corresponding semantics. Indeed, the analysis of gestures involving human agents is non-trivial due to unpredictable motions involving an environment or other agents acting on each other, all of which often posses an unknown dynamical model. Most of the research revolving around this issue has primarily focused on adapting and testing existing machine learning algorithms to improve *featurization* (i.e. the reduction of the search space), *semantic classification*, and *temporal regression* of actions.

The most prominent algorithms for featurization employ *Convolutional Neural Networks* designed to encode large amounts of data into compact representations to be more easily analyzed. For single shot detection of static images, the analysis is, usually, a classification problem; action segmentation requires solving both a classification and a regression to introduce the semantic meaning through labeling and to identify the underlying modes.

This chapter is based on the paper "Efficient Time-Interpolated Convolutional Network for Fine-Grained Action Segmentation", which has been submitted by the author to *Pattern Recognition Letters* as of the writing of this thesis.

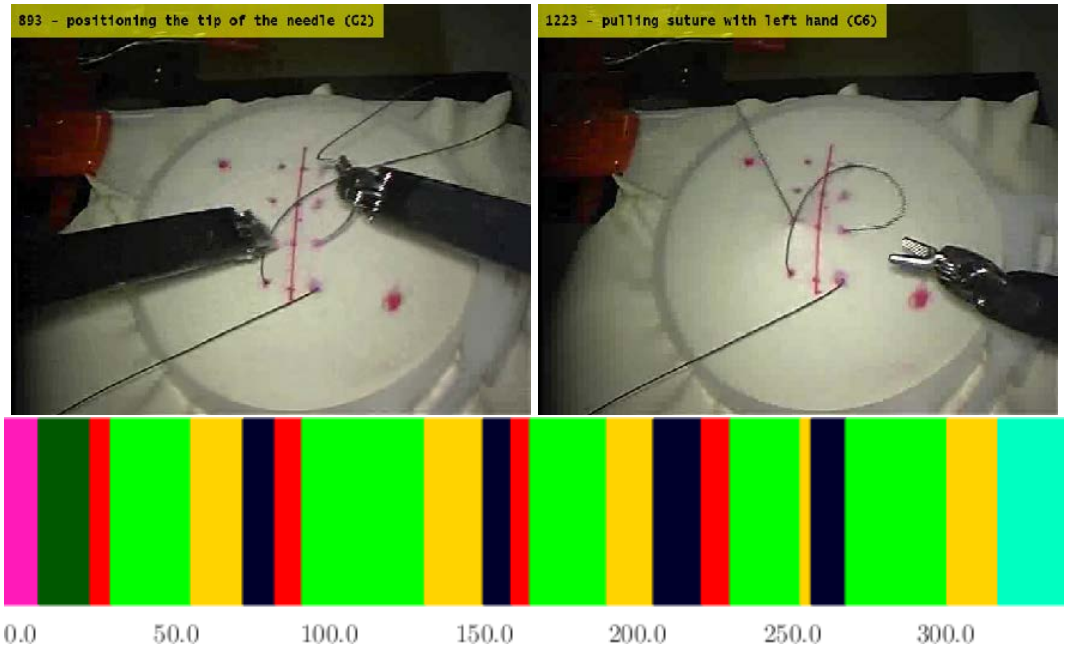

Fig. 3.1: Samples of frames taken from the JIGSAWS Suturing dataset with superimposed action class labels. The colored plot below is an example of gesture segmentation on these action classes (see Table 3.1 for colors coding).

The classification and regression problems can be solved in two major techniques for learning, *Supervised* and *Unsupervised*. Within the limits of a fully known dataset of inputs and outputs $\Gamma = \{\mathbf{X}, Y\}$, with *features* $\mathbf{X}$ and labels Y, *Supervised* techniques explicitly learn a set of parameters Φ for a model $\mathcal{M}$ that minimize a loss function $\mathcal{L}(Y, \mathcal{M}(\mathbf{X}, \Phi))$. Inevitably, both the selection of a correct loss function and the labeling quality of the dataset play a role as important as the selection of the appropriate model to the overall results. Conversely, *Unsupervised* techniques rely only on geometric or stochastic measurements, from the simple *Mean Squared Error* to more complex *probability distribution functions* to cluster data points within distinct sets, for classification. Albeit being virtually immune to issues with labeling quality, unsupervised-only algorithms tend to produce lower quality results when compared to algorithms that include supervised classification.

Another attribute for machine learning systems is whether they can operate *online*, i.e. the processing takes place on a sample-by-sample basis, or *offline*. Online methods are similar to *Auto-Regressive* linear models in that the computation for the current sample is dependent only on the previous ones, thus the overall model can operate in a *causal* manner. Methods based on clustering could be difficult or even impossible to apply to real-time processing depending on either the required computational load or the metrics defined to cluster the data which tend to analyze the entire domain without *causality*.

This chapter presents a selection of the most relevant technologies available in literature to address the issue of action/gesture segmentation (and, in general, action/gesture recognition). It introduces the *neural network* developed for feature extraction from both images and kinematic trajectories and filter them temporally to produce an action segmentation that is compatible with the requirements of a real-time controller.

3.2 Problem Formulation

We will present a valid algorithm to perform real-time action segmentation, which involves the simultaneous classification in the search space of actions represented by a categorical representation and temporal regression of actions as they change. The starting point is a higher-level representation of the underlying data involved (*e.g.* video and kinematic streams) that

already highlights the most prominent latent variables. For the solution adopted to extract features, we refer to Chapter 5 where the complete system working on low-level data is analyzed and validated.

The desired output for the algorithm is visualized by Figure 3.1: the example frames are extracted from the Suturing subset of the JIGSAWS dataset [26] on which are superimposed the labelling corresponding to the action being performed at the specific frame (Table 3.1 contains the specific definition of actions for the dataset). The color barcode below the frames is the entire video segmented in discrete sequences depending on the highest probability that the action being performed matches one of the pre-defined labels. To perform this *multi-class* classification, the best model to fit is represented by a *categorical probability distribution* and, specifically, a *Gibbs sampling* over conditional distributions,

$$\begin{aligned} \alpha &= (\alpha_1, \dots, \alpha_C) \\ \mathbf{p}|\alpha &= (p_1, \dots, p_C) && \sim \mathrm{Dir}(C, \alpha) \\ \mathbb{X}|\mathbf{p} &= (x_1, \dots, x_C) && \sim \mathrm{Cat}(C, \mathbf{p}) \end{aligned} \tag{3.1}$$

where α is the set of concentration hyperparameters for the conjugate prior *Dirichlet distribution* ($\mathrm{Dir}(C, \alpha)$) and $\mathbb{X}$ is the set of *observation nodes*; the expected value for such model is computed as

$$\mathbb{E}[p_i|\mathbb{X}, \alpha] = \frac{c_i + \alpha_i}{N + \sum_c \alpha_c} \tag{3.2}$$

with N as the total number of observations. Operatively, the discrete-time input for the system at time $t \in T$ is a collection of signals $\mathbf{u}_t \in \mathbb{R}^D$ representing the evolution of actions in a sequence of T samples from a D-dimensional feature space. The ground truth $\mathbf{y}_t \in \{0, 1\}^C$ is a discrete classification of the actions in a categorical C-dimensional space:

$$\mathbf{y}_t = [\mathbf{y}_{t,i}, i = 1, \dots, C] = [0, 1, 0, 0, 0, \cdots, 0]. \tag{3.3}$$

As only one action can be active at any given time $t \in T$, we have that

$$\sum_{i=1}^{C} \mathbf{y}_{t,i} = 1$$

.

For each input $\mathbf{u}_t$ there exists an associated multi-dimensional likelihood output $\mathbb{P}(\hat{\mathbf{y}}_t) \in [0, 1]^C$ to be used to drive the estimation of $\hat{\mathbf{y}}_t$, for instance:

$$\begin{array}{lccccccc} \mathbb{P}(\hat{\mathbf{y}}_t) = [0.1 & 0.8 & 0.05 & 0.05 & 0.0 & \cdots & 0.0] \\ \qquad\qquad\quad \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \cdots & \downarrow \\ \hat{\mathbf{y}}_t \quad = [0 & 1 & 0 & 0 & 0 & \cdots & 0] \end{array} \tag{3.4}$$

Table 3.2 summarizes the notation used in this Chapter.

3.3 Related Works

Historically, the first approaches to gesture recognition exploited variations of *Markov Chain* probability distributions that require ground truth data for parameter estimation, which typically is performed by *Expectation-Maximization*. They are based on the assumption that the segment being analyzed at each time step depends only on the previous segment and the current observations [2]. However, this assumption could introduce errors in the results, usually in the form of over-segmentation, since it tends to disregard longer action occurrences. To address this issue, studies were conducted by using Skip-Chain Conditional Random Fields [41, 47], which improved the overall results, while others authors considered more advanced Markovian models formulations [72].

Only few examples of fully unsupervised methods can be found in literature, one of which is presented in [38], where kinematic features from the JIGSAWS dataset is clustered

Table 3.1: JIGSAWS action labels as presented in [26]. Labels indicated with (*) do not appear in the Suturing subset being considered.

Label	Action
G1	reaching for the needle with right hand;
G2	positioning the tip of the needle;
G3	pushing needle through the tissue;
G4	transferring needle from left to right;
G5	moving to center of workspace with needle in grip;
G6	pulling suture with left hand;
* G7	pulling suture with right hand;
G8	orienting needle;
G9	using right hand to help tighten suture;
* G10	loosening more suture;
G11	dropping suture and moving to end points;
* G12	reaching for needle with left hand;
* G13	making C loop around right hand;
* G14	reaching for suture with right hand;
* G15	pulling suture with both hands.

Table 3.2: Notation

Var	Description
T	Number of feature samples
D	Size of each input feature vector
C	Number of categories (labels)
$\mathbf{u}$	Data input vector in $\mathbb{R}^{D\times T}$, $\mathbf{u}_t \in \mathbb{R}^D$
$\mathbf{y}$	Categorical ground truth, $\mathbf{y}_t \in \{0,1\}^C$
$\hat{\mathbf{y}}$	Categorical estimation, $\hat{\mathbf{y}}_t \in \{0,1\}^C$
ℓ	Operation block index, $\ell \in \{1,2,3\}$
$\mathbf{x}_{i,t}^{\ell}$	$\ell - th$ vector of outputs at layer i at time t
$\{c,a\}^{\ell}$	Functions applied in block ℓ
$S(t)$	Piecewise Linear Interpolating Function
$\mathbf{z}$	Output of the interpolator function
N^{ℓ}	Number of convolution filters in block ℓ
H	Window size
$\mathbf{W}^{\ell}$	Convolution kernel matrix for block ℓ
$\mathbf{b}^{\ell}$	Vector of biases for block ℓ
$\mathbf{M}$	Matrix of weights for the softmax layer
$\mathbf{o}$	Vector of bias for the softmax layer

by fitting Gaussian mixture models and, then, associated with the corresponding action labels. The work was later improved in [50] to include video features extracted from fine-tuned convolutional neural networks. Another approach using kinematic features is portrayed in [18], where a fuzzy membership matrix for PCA features is created. Segments starting from a fine segmentation prior are continuously aggregated whenever their distance falls below a threshold ϵ.

These techniques do not require training using ground truth since they rely on fitting probability distributions. As such, they are theoretically capable of operating in an unsupervised matter. On the other hand, they require many hyperparameters to be set and they usually produce lower quality results when compared to supervised techniques, since testing demonstrated how they can easily diverge towards either under or over-segmentation.

Assuming the start and end time instants of an action are identified, the use of *holistic features*, such as Bag of Words and Motion History Images, can be used to classify the

resulting segments into actions [14]. The approach presented in [81] represents kinematic signals and video streams, respectively, in terms of a linear dynamical system and of a holistic bag of words. Multiple kernel learning is applied to merge segmentation and classification results.

The use of feature spaces in video streams was dominant in the unsupervised method in [17], which has been later improved upon by [11]. It addresses the issue of data labeling thanks to incremental active learning, starting from pre-labeled data and invoking the user only when the clustering process for the features falls below a predefined quality threshold. Up to now, it has not been tested on fine-grained activity recognition. The solution presented in [39], instead, combines Hidden-Markov Models with Gaussian Mixture Models and Fisher Vectors to create an end-to-end generative pipeline for action segmentation. It produces state-of-the-art results at the expense of high computational complexity and parametrization.

Among the variety of machine learning algorithms, *neural networks* present the highest level of adaptability and generalization. Indeed, through the choice of the appropriate structure and loss function, it is possible to operate in both a supervised and unsupervised manner to solve both classification and regression problems.

The mathematical basis for this capability of neural networks to model non-linear regression and classification functions is found in the mathematical field of *approximation theory* starting from the *Weierstrass Approximation Theorem*, which states that any continuous function over a closed interval on the real axis can be expressed, in the specific interval, as an absolutely and uniformly convergent series of polynomials.

Theorem 3.1 (Weierstrass Approximation Theorem). *Let f be a continuous real-valued function defined on $[a, b] \in \mathbb{R}$, then $\forall \epsilon > 0 \in \mathbb{R}, \exists p \in P$, with P being the set of all polynomial functions, such that, $\forall x \in [a, b]$,*

$$|f(x) - p(x)| < \epsilon.$$

This result is fundamental in defining the basics of mathematical approximation theory, but, in the current formulation, it is limited on a compact subset of $\mathbb{R}$.

Amongst the extensions of the Weierstrass theorem theorized over different algebraic spaces, the most significant one that operates on non-linear input output mappings, is the *Universal Approximation Theorem*, presented here in the version available in [30].

Theorem 3.2 (Universal Approximation Theorem). *Let $\varphi(\cdot)$ be a non-constant, bounded, and monotone-increasing continuous function. Let I_{m_0} denote the m_0-dimensional unit hypercube $[0, 1]^{m_0}$. The space of continuous functions on I_{m_0} is denoted by $C(I_{m_0})$. Then, given any scalar $\varepsilon > 0$ and any function $f \in C(I_{m_0})$, there exist an integer m_1, real constants α_i, b_i and vectors $w_{ij} \in \mathbb{R}^{m_0 \times m_1}$, where $i = 1, \cdots, N$, such that we may define:*

$$F(x_1, \ldots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \varphi \left(\sum_{j=1}^{m_0} w_{ij} x_j + b_i \right)$$

as an approximation of the function $f(\cdot)$; that is,

$$|F(x_1, \ldots, x_{m_0}) - f(x_1, \ldots, x_{m_0})| < \varepsilon$$

for all $x_1, \ldots, x_{m_0} \in I_{m_0}$.

In summary, the theorem states that functions of the form $F(x_1, \ldots, x_{m_0})$ are *dense* in $C(I_{m_0})$. This fact still holds when replacing I_{m_0} with any compact subset of $\mathbb{R}^{m_0}$.

The Universal Approximation Theorem is at the basis of the *Multilayer Perceptron*, the first neural network model that, thanks to the increase in computational power, overcame the theoretical limitations of the *Rosenblatt Perceptron* to become one of the most effective *logistic regressors*.

Many different variations of neural network architectures have since been developed to tackle both action segmentation and classification. The advantage of neural networks lies in

their adaptability to heterogeneous tasks by simple variations of their topology, at the cost of being more dependent on extensive training data. The two most prominent network classes are *Convolutional Neural Networks* (CNNs) and *Recurrent Neural Networks* (RNNs). The former are usually applied to image analysis and segmentation; the latter are used specifically for time-series predictions, usually in the Long/Short-Term Memory network incarnation, as in [73, 69, 16], where both structures are combined to generate an end-to-end solution from frame to action segment. Ultimately, RNN-based methods involve an extensive topology that suffers from high computational complexity and short attention span due to the nature of its recurrent layers.

The approach found in [66] presents an end-to-end topology that uses 3D convolutions to train the spatial and temporal components concurrently. It follows, therefore, the same concept of temporal convolution instead of explicitly recurrent units to improve performance. Its advantage is found in the capability of retrieving both spatial and temporal features in each convolution operation, thus simplifying the overall structure. Its main drawback is visible at training time, since the increased amount of convolution operators requires a very high memory footprint less suitable for real time implementation in human-robot interaction scenarios.

The approach found in [43] employs a supervised encoder-decoder architecture to temporally filter the analyzed features. It represents the starting point for the improved temporal filter presented in this Chapter, Section 3.4. The complete processing stack developed in this work, which includes the featurization phase, employs a multi-modal neural network and a *Time-Interpolated Fully-Convolutional* temporal *autoencoder* that applies an interpolation function for the *decoder* phase.

The use of interpolation operators in deconvolution layers has been tested in [45] as an improvement over the object detection capabilities of *Single-Shot Detectors* (SSDs) while reducing the required number of floating point operations performed per second (FLOPs). Although the reasoning behind the choice of using an interpolating layer is similar to the rationale for the solution presented in this work, which is to boost both performance while reducing computational demands, interpolation is applied exclusively to the spatial domain. Furthermore, it is not employed in substitution of the decoder phase, but rather as a smoothing operation placed in between convolution layers. Nevertheless, the results obtained by the authors provide an additional support to the claim that the insertion of interpolating layers within encoder-decoder structures has a positive impact in temporal segmentation capabilities.

3.4 Efficient Time-Interpolated Convolutional Network for Fine-Grained Action Segmentation

This section presents the solution designed to filter temporal data, called the Time-Interpolated Fully-Convolutional (*TIFC*) neural network for discriminative segmentation of gestures to find a collection of functions mapping feature signals to action labels at the correct time. The focus is limited on the analysis of signals generated from video feeds recorded from a single camera operating in the RGB color space from which high-level features were extracted following the approach presented in [**Lea2016a**]. The proposed approach builds on the neural network presented in [43] by introducing variations in both the topology of the network and functions of the layers. In this application, the featurization phase is performed via convolutional neural network which takes as input

- a full RGB color frame;
- a Motion History Image computed over a temporal window of two seconds as an additional image layer.

For the evaluation of this component alone, the convolution operators act on a non-causal sliding window formulation for the problem. This means that the discrimination at each sample involves the use of information from past and future samples for the sample under analysis within the window. As aforementioned, a causal formulation would be required to

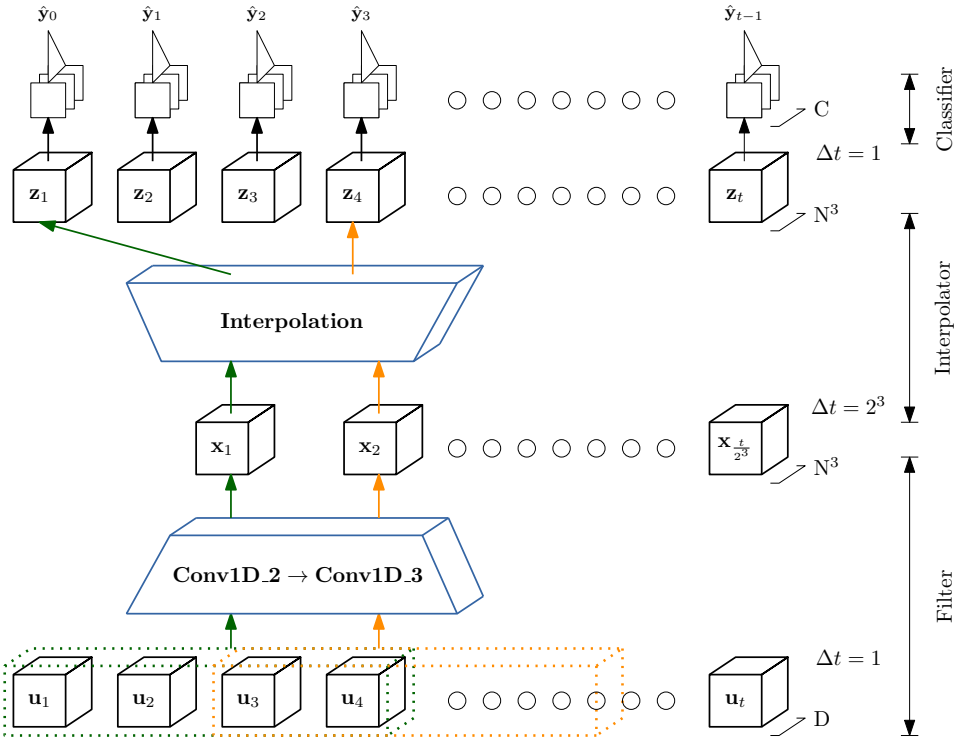

Fig. 3.2: Time-Interpolated Fully-Convolutional Network graph. The Input u_t is a set of D dimensional features that is processed by three sets of temporal convolutions with stride greater than 1 to produce an encoded feature set $x_{\frac{t}{2^3}}$ that is interpolated over the time dimension to create the output z_t.

use this solution as a real-time model to predict future actions. This modification will be presented in Chapter 5 on the complete network.

The training uses gradient descent on the error derivative between input $\mathbf{u}_t$ and output $\mathbf{y}_t$ to find the best approximation function $\hat{f} : \mathbf{u} \rightarrow \hat{\mathbf{y}}$.
The intent is to infer the function $\hat{f}$ using a minimal and more efficient neural network topology capable of improving on standard structures for fine-grained temporal action segmentation.

The proposed *TIFC* network is articulated into three subsystems:

1. a *filter*, operating on the input signals with multiple convolutions at different temporal sampling periods;
2. an *interpolator*, which reconstructs the original length of the input time series;
3. a *classifier*, which constructs a probability distribution $\mathbb{P}(\hat{\mathbf{y}})$ over the signals produced by the function at the second to last layer.

3.4.1 Filter

The TIFC *filter* stratum consists of three blocks ($\ell = \{1, 2, 3\}$) each composed by two layers. Within each block, each layer implements one of the interleaved functions defined below, amounting to the overall mapping:

$$\mathbf{x}_{2,t}^{\ell-1} \overset{c^\ell}{\mapsto} \mathbf{x}_{1,t'}^{\ell} \overset{a^\ell}{\mapsto} \mathbf{x}_{2,t'}^{\ell},$$

where:

1. the *convolution* layer c^ℓ operates over a fixed window size with stride $s = 2$ (i.e. the distance between consecutive locations of the kernel)

$$\mathbf{x}^{\ell}_{1,t'} = c^{\ell}(\mathbf{x}^{\ell-1}_{2,t}) = \mathbf{W}^{\ell} * \mathbf{x}^{\ell-1}_{2,t} + b^{\ell}, \tag{3.5}$$

where $\mathbf{W}^{\ell}$ and b^{ℓ} are the convolution kernel matrix and a bias vector for the $\ell - th$ block, respectively, and $\mathbf{x}^{0}_{2,t} \equiv \mathbf{u}_t$;

2. the *activation* function a^{ℓ} normalizes and redistributes the values

$$\mathbf{x}^{\ell}_{2,t'} = a^{\ell}(\mathbf{x}^{\ell}_{1,t'}) = NormPReLU(\mathbf{x}^{\ell}_{1,t'}). \tag{3.6}$$

The chosen activation function, $NormPReLU$, is a normalized rectified linear unit that does not simply set to zero all the negative values, but evaluates a parametric linear function such as:

$$a^{\ell} = \begin{cases} \alpha \frac{\mathbf{x}^{\ell}_{1,t'}}{\max \mathrm{abs}(\mathbf{x}^{\ell}_{1,t'})}, & \text{if } \mathbf{x}^{\ell}_{1,t'} < 0 \\ \frac{\mathbf{x}^{\ell}_{1,t'}}{\max \mathrm{abs}(\mathbf{x}^{\ell}_{1,t'})}, & \text{otherwise,} \end{cases} \tag{3.7}$$

where the parameter α is learned through training. This particular function has been shown in [31] to avoid issues with parameter initialization associated with randomly extracted negative values interrupting the gradient descent procedure, because of zeros introduced by standard ReLU activation.

We use the notation t' to index the samples produced by the convolution layer for, within each block ℓ, the output $\mathbf{x}^{\ell}_{1,t'}$ of c^{ℓ} has only a fraction of the samples as the output $\mathbf{x}^{\ell-1}_{2}$ of the top layer $i = 2$ of the previous block $\ell - 1$, depending on the value for convolution stride. Namely, assuming s as the stride size:

$$t' \in \{1, \ldots, |\mathbf{x}^{\ell-1}_{2}|/s\}$$

where the norm $|\cdot|$ indicates the number of time samples.

Down-sampling the data aims at extending the *receptive field* F of each network neuron, i.e., the length of the input sequence $\mathbf{u}$ involved in generating each filter output $\mathbf{x}^{3}_{2}$. Since the focus is on temporal convolutions, a wider receptive field allows the neurons to better relate information located at both close and distant time periods, thus capturing the inter-dependencies which characterize, respectively, short and long gestures.
Given stride s and convolution window size H, the overall receptive field for the filter can be computed recursively as:

$$\begin{cases} F^{0} & = 1; \\ F^{\ell} & = s(F^{\ell-1} - 1) + H, \end{cases} \tag{3.8}$$

in each block ℓ of the network, with an increasing number of convolution filters N^{ℓ}.

3.4.2 Interpolator

The main difference between the new network architecture presented here and the one in [42] is found in the decoder phase, i.e., in the way time series are reconstructed following the encoding.

The task of the *interpolator* stratum is to re-build the input's timeline from the down-sampled signal produced by the three-block filter in order to assign to each time sample the corresponding action label. The concept proposed in this work is to use simple linear upsampling to recreate the same number of samples as in the input $\mathbf{u}$.
The interpolating function $S(t), t = 1, \ldots, T$ is a vector-valued piecewise linear function of dimension D (the same as the input vectors), composed by $\bar{T} - 1$ linear segments, where $\bar{T} = |\mathbf{x}^{3}_{2}|$ is the number of samples outputted by the last block $\ell = 3$ of the filter:

$$\begin{aligned} S(t) &= \sum_{k=1}^{\bar{T}} s_k(t) \chi_{[k,k+1)}(t) \quad t = 1, \ldots, T; \\ s_k(t) &= \left(\mathbf{x}^{3}_{2,k+1} - \mathbf{x}^{3}_{2,k}\right)(t - k) + \mathbf{x}^{3}_{2,k}, \end{aligned} \tag{3.9}$$

where $k = 1, \ldots, (\bar{T} - 1)$ and the indicator function $\chi_{[k,k+1)}(t)$ assumes non-zero values only in the specified interval $[k, k + 1)$.

Using this formulation, we can compute the partial derivatives of $S(t)$ with respect to $\mathbf{x}_{2,k}$, namely:

$$\frac{\partial S(t)}{\partial \mathbf{x}_{2,k}} = \begin{cases} 2 - t & \text{if } k = 1, \\ t - k & \text{if } k = \bar{T}, \\ 2 & \text{otherwise.} \end{cases} \tag{3.10}$$

Since each output of the interpolator function is activated by the two samples defining the corresponding interpolating line (cfr. Figure 3.3), the receptive field for each neuron in the *interpolator* stratum is $F(S(t)) = 2F(\ell)$, for $\ell = 3$.

A graph illustrating the adopted interpolation strategy is shown in Figure 3.3. It illustrates upsampling with a non-integer ratio between the number of input ($\bar{T}$) and output (T) samples, which maintains the first and last samples. The output of the interpolator is denoted by $\mathbf{z}_t \doteq S(t)$ in Figure 3.2.

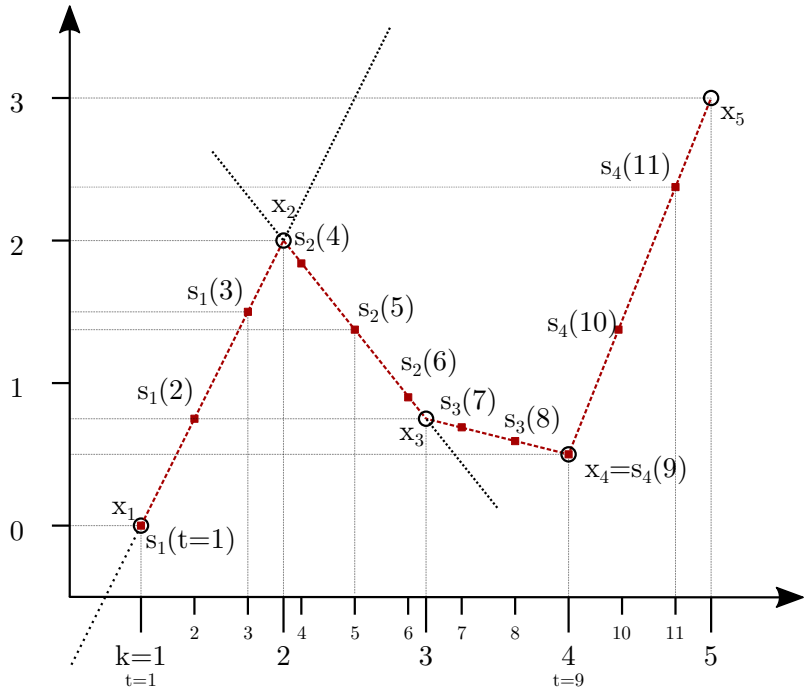

Fig. 3.3: Example of interpolation for a single feature from 5 data points (x_i) to 11 data points ($s_{j(n)}$) upsampling using 4 linear interpolating segments.

The function $S(t)$ can be considered a regularization function that is, by construction, C_0 continuous, but it does not respect the condition imposed for activation functions $\phi(\cdot)$ set by the *Universal Approximation Theorem*(3.2). Its theoretical viability for use in neural networks is still being studied and only empirical evaluations can be provided as of this writing.

Interpolation Issues and Network Design

The interpolation function assumes a constant sampling period between consecutive inputs. This represents a potential issue whenever used in conjunction with a max pooling layer, widely used in convolutional networks topologies when downsampling: the output of such operator introduces a non-uniform time sampling effect [56].

For this reason, in our *TIFC* network, max pooling layers are completely removed. The convolutions c_i^ℓ are, instead, directly computed with *stride* = 2, as indicated in [68]. The

filter becomes, effectively, a *fully convolutional* network. Using the stride option allows the network to preserve a constant sampling period in each layer. Fully convolutional networks are commonly used, for instance, in speech recognition systems [75].
The reason for using an interpolating function comes from observing that changes in actions performed by humans take place gradually over time rather than instantly, as indicated in [18]. In recreating the input as faithfully as possible from the output signal of the filter by means of an interpolation function, the network provides the classifier with information which better represents the original input, upon which a more accurate probability distribution can be built. As the operation is evaluated within the signal's domain, it does not introduce any spurious information that is not already present in its filtered version.

The validation tests empirically prove the validity of the *TiFC* network, which is shown to outperform standard specular encoder-decoder neural network architectures.

3.4.3 Classifier

As the last layer of the network, the classifier uses a classical $softmax$ operator:

$$\mathbb{P}(\hat{\mathbf{y}}_t) = \text{soft}\max\left(\mathbf{M}\mathbf{z}_t + \mathbf{o}\right), \tag{3.11}$$

where $\mathbf{M}$ is a matrix of weights, $\mathbf{z}_t$ is the output of the interpolator, and $\mathbf{o}$ is a bias vector. The function represents the Gibbs measure applied to a categorical distribution.

3.5 Experimental Validation

3.5.1 Evaluation Strategy

The dataset choice has been primarily driven by the potential application of this network to surgical gesture recognition. Not many dataset are available for such a peculiar scenario due to complex logistics and, possibly, legal obstacles in obtaining data. Additionally, in order to effectively compare results and provide a viable statistics, it was deemed necessary to select a well established dataset. For these reasons we selected JIGSAWS [27], whose suturing activity subset contains 39 sequences performed by 8 users about 5 times each, and provides videos with their corresponding action labels distributed over 10 classes. The *Leave One User Out* (LOUO) cross validation is being performed, i.e. out of 8 users the network is trained on sequences performed by 7 users and cross-validated on the 8th, to maintain coherence with the results available in literature; scores are averaged amongst all combinations of training and test users. This is ideal when the correlation between actions performed by the same user on different trials is high, whereas it diminishes when checked across users. It allows us to assess training reliability across the whole dataset. The input feature space is the same used in [43] and has been made available by the authors, allowing a direct comparison between the two solutions.

To provide a better comparative assessment of our network's results and test the generalization power of the presented solution, we also tested it on two additional datasets which are not related to surgical procedures, *50Salads* [70] and *GTEA* [19], that present similarities in terms of how the data is acquired and gesture granularity. Validation is again performed in following the LOUO rule.

For evaluation, we adopted three performance measures:

- 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 [80] between the longest of two strings $(s, \hat{s})$. 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

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

Given accuracy and edit score, it is possible to understand whether the algorithm is over- or under-segmenting sequences while fine-tuning the parameters. Obviously, if both accuracy and edit score increase, the network is producing a better segmentation. Conversely, if only accuracy increases, two alternatives may be occurring:

- if the edit score decreases, the network tends to over segment since it could be identifying multiple out-of-order actions within the same segment;
- if the edit score remains constant, the network is encountering a temporal shift in the prediction. This could mean that the test sequence is not completely temporally aligned with the related training ones.

Finally, under-segmentation occurs when both accuracy and edit score decrease.

3.5.2 Validation Parameters

We conducted tests over the *TIFC* topology to evaluate its effectiveness and efficiency. Across the various tests, we have kept constant parameters such as kernel size H and the number of convolution functions applied in the filter layer, so that $|\mathbf{W}^\ell| = H \times \{32, 64, 96\}$, for $\ell = \{1, 2, 3\}$.

After each convolution, when choosing a stride $s > 1$, the number of samples is divided by the stride, but the window size remains fixed. As mentioned before, this has the effect of increasing the receptive field of the network, effectively eliminating the need for additional convolution operations. The interpolation, then, effectively doubles the receptive field in the decoder phase prior to classification.

The gradient descent algorithm chosen for the backpropagation phase is the Adaptive Moment Estimator (ADAM) [36]. The learning rate has been set to 0.0001 to avoid abrupt variations during learning; consequently, the training steps have been increased, from 200 to 500, to increase effectiveness.

The TIFC network has, in total, 8 layers, of which 6 are convolutions and activation functions in the filter. The remaining 2 are the interpolator and the softmax operator, both of which are non parametric functions. The number of trainable parameters for the JIGSAWS suturing task is 738 627 when selecting $H = 20$. In comparison, the ED-TCN presents 1 782 848 parameters at training time with over 19 layers, showing how TIFC significantly reduces both training and testing complexity.

3.5.3 Results

Quantitative Comparison

The quantitative results of the tests can be found in Tables 3.33.43.5: they present a comparison between our proposed network architecture and those reported for some of the solutions available in literature, with the approaches of [43, 44] *in primis*. We report the average performance value of 50 training instances to verify the network's robustness to random parameter initialization.

Table 3.3: Results for the **JIGSAWS** dataset (%); † the result was taken from the most successful approach which uses video data only; ‡ the authors do not indicate the amount of downsampling on their input data.

Algorithm	Accuracy	Edit Score	F_1
MsM-CRF [**Gao2014**]	77.29†	n.a.	n.a.
ED-TCN [42]	81.4	83.1	87.1
TricorNet [16]	**82.9**	86.8	n.a.
TDRN [44]‡	*84.6*	*90.2*	*92.9*
TIFC [our]	81.97	**86.92**	**91.1**

Table 3.4: Results for the **50salads** dataset (%, mid granularity); † the result is not explicitly indicated as the average using a LOUO approach; ‡ as in Table 3.3.

Algorithm	Accuracy	Edit Score	F_1
SLM [63]	54.2	44.8	n.a.
ED-TCN [42]	64.7	59.8	n.a.
GMM [39]	83.8†	n.a.	n.a.
TDRN [44]	**68.1**	**66.0**	**72.9**
TIFC [our]	66.45	61.92	64.4

Table 3.5: Results for the **GTEA** dataset; ‡ as in Table 3.3 (%, mid granularity).

Algorithm	Accuracy	Edit Score	F_1
EgoNet [67]	68.5	n.a.	n.a.
ED-TCN [42]	62.5	58.8	72.2
TricorNet [16]	**64.8**	n.a.	76.0
TDRN [44]‡	*70.1*	*74.1*	*79.2*
TIFC [our]	63.24	**67.67**	**77.20**

The reported overall accuracy and edit score are approximately 81% and 86% on the JIGSAWS dataset respectively; when compared with *ED-TCN* [42], our *TIFC* exhibits an average 3% improvement in the edit score without repercussions on accuracy. The network is closer to the results of the current state of the art presented in [44] while using a much simpler architecture.
To ensure a fair performance comparison, we also need to stress that [44] appears to downsample the input video stream, whereas this work uses all available frames: this creates a discrepancy between the two methods that could lead to variations in the results.

These results validate the hypothesis that the proposed interpolator, as opposed to standard, parameter heavy decoders, is an effective solution to decrease the network's computational complexity as it allows us to reduce the number of required training parameters and also improve overall performance.

Example Segmentation

Figure 3.4 shows a sample segmentation result, where different actions have been coded using different colors (Table 3.1). For example, the action "*pushing needle through tissue*" is represented by the light green color and is executed exactly four times throughout the trial. The frame-by-frame differences between ground truth and predicted labels provide a visual representation of performance in terms of both accuracy and edit score. A black bar indicates a time-shift difference between correct predictions and ground truth, whereas a red line indicates a mismatched prediction. As discussed above, a time-shift does not influence the edit score since the sequence is correctly identified; it does, however, affect the overall accuracy and precision given that an incorrect temporal alignment could negatively influence the results of subsequent evaluations. Nevertheless, an evaluation of the shift conducted over

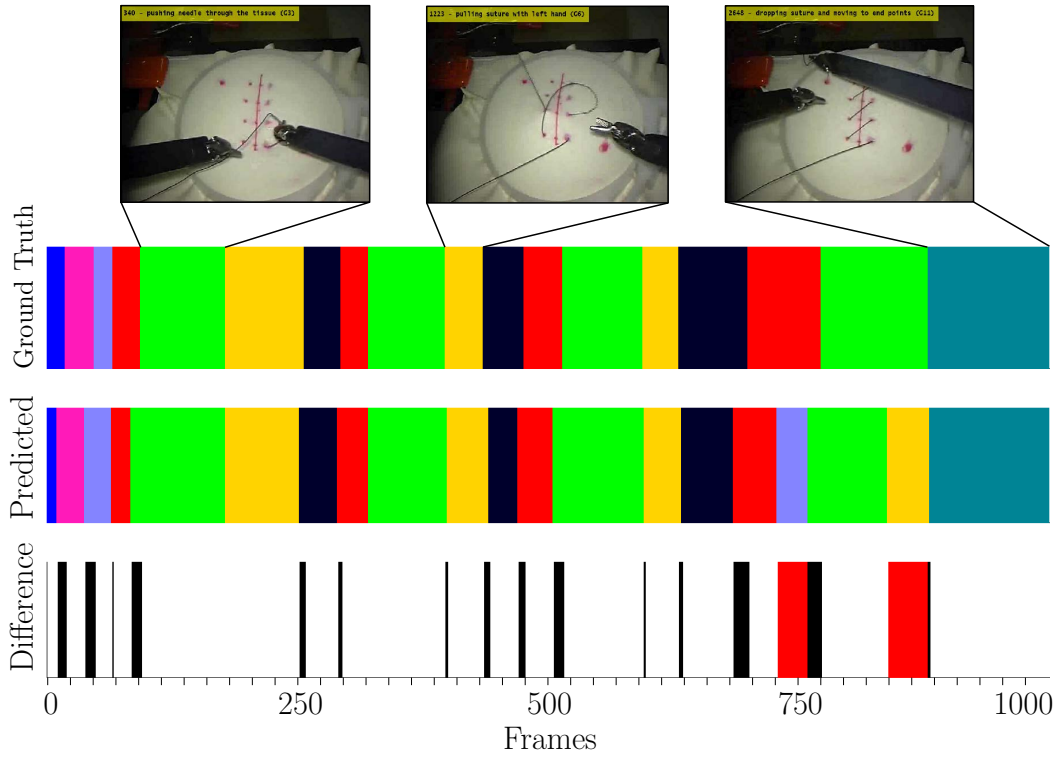

Fig. 3.4: Segmentation test results of the median user ("Suturing_F00x") in the JIGSAWS dataset: the black lines indicate temporal shifts affecting correct predictions, whereas the red lines indicate mispredicted actions.

the median user ("Suturing_F00x") indicates that the mean value is around 5 frames (or 0.2 seconds @ 25 fps) with a peak of 14 frames (0.56s).

Confusion matrix

Accuracy is better presented in the form of a *confusion matrix* (Figure 3.5). It can be observed how, on our JIGSAWS tests, the gesture "*using right hand to help tighten suture*" (G9) is confused with both (G4) and (G6). This is due to the diverse skills and styles characterizing each user: furthermore, this particular action appears in only 1.6% of the entire dataset. Such low occurrence inevitably influences both accuracy and edit score as it creates a bias in the sequences' training only for the specific users that adopt it.

Robustness and Performance Analysis

Choosing the most appropriate kernel is especially an issue when dealing with temporal convolutions since it affects the receptive field and represents the amount of "history" used by the network to predict the sequence of labels [3]. The second point has an impact on the overall memory requirement at testing time, since the signal needs to fill a sliding history window and be kept in memory for the sequencing process to complete.

TIFC presents a reduced sensitivity towards changes in this parameter, thanks to the comparative simplicity of its structure and the overall low number of convolution operations it requires, when compared with similar networks. Figure 3.6 proves this by showing how the F_1 score stabilizes despite an increasing kernel size value.

Additionally, overall computational efficiency is greatly improved thanks to the reduced number of parameters to be evaluated and adjusted in both the forward and the backward propagation phases. Table 3.6 compares the required parameters and floating point operations per second (FLOPs), evaluated using a profiler, for the network presented in this work

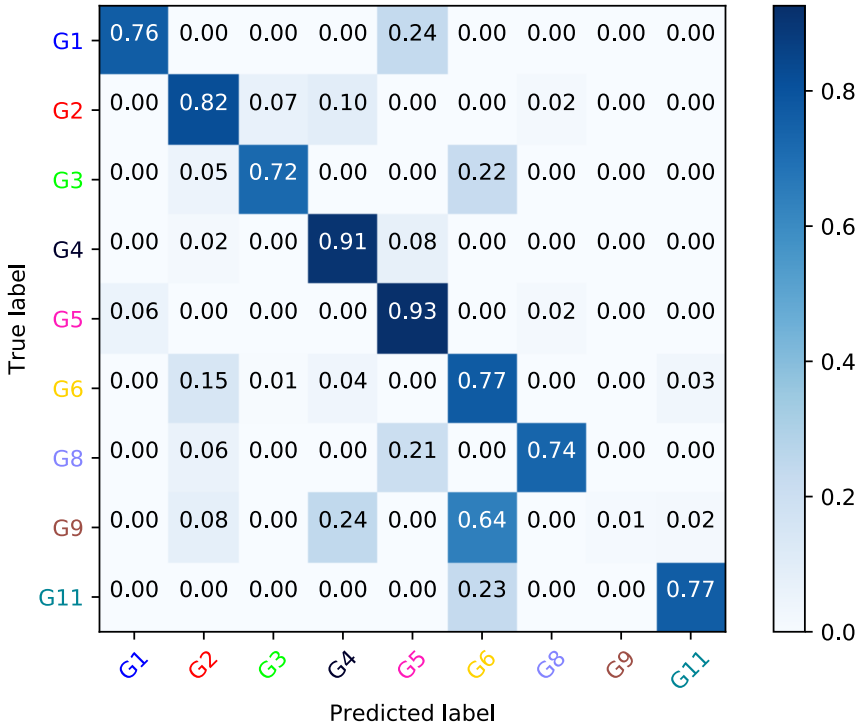

Fig. 3.5: Normalized confusion matrix for the combined 5 trials of the median user in the JIGSAWS dataset.

and the reference article presented in [43]. It shows that the number of parameters required drops by almost 200 thousand, and the required number of operations drops by a factor 3. Thanks to these improvements, the required time per training epoch (forward and backward) also drops by around a third, although any timing comparison is clearly dependent on the computational power of the hardware platform being used.

Model	Parameters (×1000)	MFLOPs	seconds per epoch
ED-TCN [43]	595	6.7	0.8
TiFC[ours]	410	1.9	0.3

Table 3.6: Performance comparison with the most similar competitor network structure.

3.6 Discussions

In our opinion, among all machine learning methods available for segmenting actions contained within data streams, only supervised neural network based approaches seem to provide the required accuracy and real-time computation for control applications

The foregoing proposed *Time-Interpolated Fully-Convolutional* network configuration has demonstrated to obtain better results over the state-of-the-art for this type of classifier using a simpler and more efficient configuration. Indeed, the experimental results show how replacing the decoder phase within the traditional encoder-decoder architecture with a linear interpolation can generate better segmentation results using roughly half as many parameters.

The piece-wise linear interpolation adopted in this work is just the simplest representative of a much larger class of interpolations which, if explored properly, could likely lead

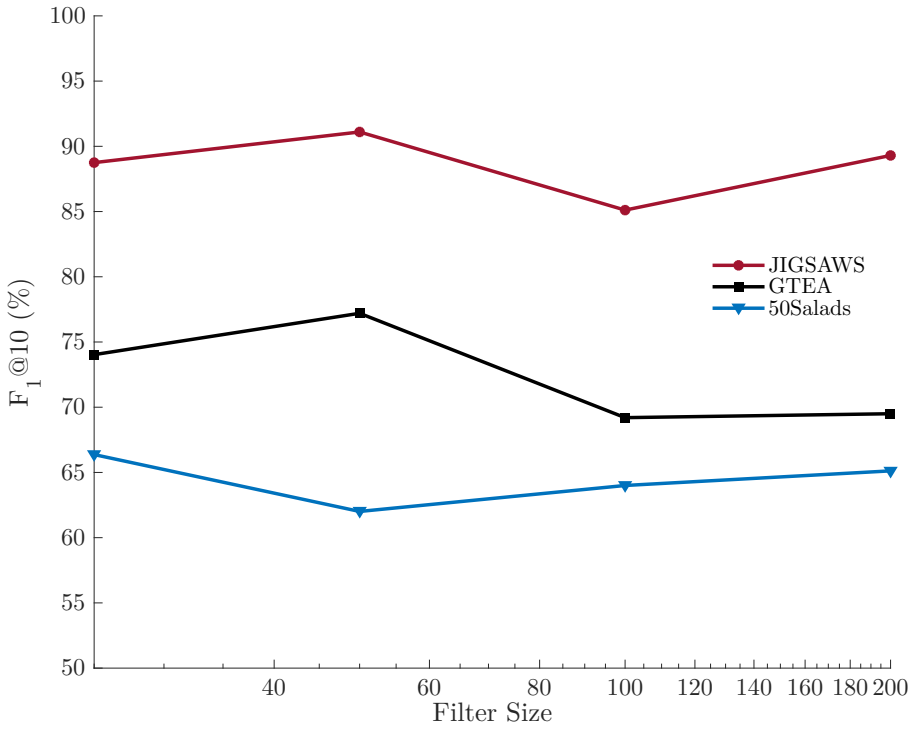

Fig. 3.6: Value of the F_1 score for each convolutional filter size.

to even better performance. As future work, the author intend to investigate, for instance, a basis spline interpolation to decrease the occurrence of both temporal shift and over-segmentation, pushing this architecture's results further. Theoretically, the relation between interpolation, convolution, and decoding is intriguing yet rather unexplored. Improved studies could provide a sound theoretical rationale for this approach. The presented methodology is intended to be used within a control system for semi-autonomous robotic surgery as a supervisory control component, therefore its computational efficiency in filtering temporal features is instrumental to allow real-time processing. An additional discussion is needed regarding the datasets used for validation. The JIGSAWS, GTEA, and 50Salads present annotations at a gesture (*surgéme*) level, which, as presented in the previous chapter, represent the first proper classification concerning semantic meaning. This fact inevitably introduces issues in transferring the acquired learning to different users, as each individual brings an unpredictable background noise in each performed motion that hinders the quality of action segmentation. This can be seen, for example, in the high variability the *Leave One User Out* cross-validation (Figure 3.5) and the mistaken action "G9", which was performed only by a single user in the entire dataset. On the other hand, these datasets have long been the most represented in action recognition and, therefore, the basis for comparison of new algorithms, especially regarding robotic minimally-invasive surgery. Regarding surgical robotic, the segmentation of *surgémes* other that higher levels of the semantic pyramid allow the usage of this technology as a *software sensor* for the reactive control of *cobots*.

In the next Chapters we will present how the confidence scores generated by the classification layer acquires additional importance as it is used to modulate the velocity at which the robots move: this helps avoiding, at the same time, situations in which the action recognition is in a "deadlock" state, due to the absence of motion of the robot in the scene, and the movement towards an erroneous goal due to a mispredicted action. To this end, an empirical evaluation of the favorable balance towards true positive classifications is performed by examining the classification performed on real-time control experiments.

4

Control of Surgical Robots

s

4.1 Introduction

Every R-MIS system has to comply with tight requirements to be allowed within an operating room and has to provide safe interaction for the tools with both soft tissues and hard surfaces, such as needles, clips, and the tools themselves. Soft-tissue surgery in non-rigid anatomical environments should take into account hardly predictable scene changes, complicated tasks requiring collision-free motion planning and physical interaction with the environment (i.e. contact with objects with unknown and even variable viscoelastic properties).

An important step in the development of cognitive surgical architectures was represented by the EU funded I-SUR project. It addressed the automation of needle insertion and suturing tasks [49] by means of a dual-arm robot with hybrid parallel/serial kinematics. The cognitive control architecture proposed by I-SUR [57] was able to operate in either teleoperated [22] or autonomous mode [58], guaranteeing a stable switch between the two and an adaptive interaction with the environment in both modes [20].

The inherent relationship between surgical and industrial collaborative robotics is demonstrated by the fact that the same control methods (i.e. admittance control with variable parameters) have also been applied to enforce stability in pHRI [71, 21]. Turning back to the specific case of the suturing task, the work presented in [58] proposes a motion planning solution based on a combination of previously specified motion primitives for the dual-arm system, designed to mimic the bimanual gestures of a human surgeon, and collision-free paths generated with a *plan-and-move* strategy. Similar approaches to surgical robotic suturing are described in [61] and [64], investigating advanced learning techniques, or [78] and [51], addressing the task using more classical robot motion planning techniques and analytic geometry. Even though surgical suturing tasks have also been automated by designing specific devices, not mimicking at all human gestures [4], the solutions based on general-purpose multi-arm robots and appropriate motion planners are more flexible and can be applied to different operations. Another notable example emphasizing the latter aspects can be found in [33].

The requirement of working in restricted environments, such as the abdomen of a patient undergoes a laparoscopy intervention, alongside human-operated tools bearing hard to predict motions leads to the implementation of reactive control methods to guarantee safety [35].

From this quick review it emerges how current solutions to actuate autonomous instruments are not meant to be used in a shared environment with a human-controlled tool primarily due to the lack of real-time generation of collision-free robot motions. A laparoscopy

This chapter is based on the papers "Dynamic Motion Planning for Autonomous Assistive Surgical Robots", published in *MDPI Electronics*, and "A motion planner integrating MPC and a dynamic waypoints generator for human-robot collaboration in a surgical scenario", submitted to *ICRA 2020* as of this writing.

robot assistant is required to reach the stated goal target but it is also especially required to avoid collisions with the instruments controlled by the main surgeon that operates with a non-predictable dynamics within the workspace. Additionally, current control methods do not dynamically react to erroneous readings coming from the sensors mostly due to the plan-and-move approach, which forces a full stop and recalculation. This Chapter introduces a control system for both the actuated SARAS arms based on *Model-Predictive Control* (MPC) [28, 8]. It allows the movement of the laparoscopy tools by solving an optimization process over a receding prediction horizon to be optimized for a defined set of constraints over a prediction horizon for the motion. Previous applications of MPCs within the confined environment of the human anatomy have been found to improve visual servoing control of under-actuated devices [6, 34]. The solution proposed hitherto specifically addresses the combined use of multiple laparoscopy instruments that need to operate without hinder each other movements and achieve the desired goal.

4.2 Model-Predictive Control: Requirements and Formulation

The MPC continuously guides the end-effector of the actuated laparoscopy tool towards a goal identified externally by a *supervisory controller* where the medical knowledge is encoded (an implementation of the latter will be presented in Chapter 5). Laparoscopic tools for minimally-invasive surgery present the physical constraint of the *remote center-of-motion* (abbreviated "RCM") positioned in the *trocar* at the entry point into the belly of the patient that makes embedding the kinematic constraints a peculiar problem with respect to standard formulation. Specifically, the issue of obstacle avoidance introduces the requirement for a *waypoint* computation strategy to guide the instruments towards the target. This has the favorable side effect of reducing the risk typical of optimization-based planning algorithms to be trapped into local minima.

4.2.1 Robot Model and Constraints

The robot model consists of four laparoscopy tools (two teleoperated and two autonomous) with four degrees-of-freedom (4-DoFs) each. Let $\tilde{x}_j \in \mathbb{R}^4$ be the state of each tool, with $\tilde{x} = [\bar{x}, \theta]$ where $\bar{x} \in \mathbb{R}^3$ are the (x, y, z) Cartesian position coordinates of the end-effector in the task space, $\theta \in \mathbb{R}$ is the rotation of the tool around its axis and $j \in \{r, l\}$, with the subscripts r and l for, respectively, the right hand and the left hand controlled tool. The control input for each autonomous arm $\tilde{u}_j \in \mathbb{R}^4, j \in \{r, l\}$ has also two components $\tilde{u}_\star = [\bar{u}, \omega]$: $\bar{u} \in \mathbb{R}^3$ are the input linear velocities of the end-effector, $\omega \in \mathbb{R}$ is the input angular velocity of the tool around its shaft. The kinematics of the overall system can be modeled as a single integrator in the discrete time domain:

$$x(k+1) = x(k) + Bu(k) \tag{4.1}$$

where $x = [\tilde{x}_r, \tilde{x}_l] \in \mathbb{R}^8$ is the state vector comprising the two tools, $B = \text{diag}\{\Delta t_c\} \in \mathbb{R}^{8\times 8}$ is the input matrix, Δt_c is the sampling time ($t = k\Delta t_c$, $k \in \mathbb{Z}$) and $u = [\tilde{u}_r, \tilde{u}_l] \in \mathbb{R}^8$ represents the control input.

Two types of constraints are considered in the MPC formulation: a *velocity limit*, and a *collision avoidance constraint*. To implement the velocity limit, the velocities of the robots are physically limited by bounding the control input:

$$|\bar{u}_j(k)| \leq \alpha(k)\, u_j^{max} \tag{4.2}$$

$$|\omega_j(k)| \leq \alpha(k)\, w_j^{max} \tag{4.3}$$

where $u_j^{max} \in \mathbb{R}^+$ is the linear velocity limit, $\omega_j^{max} \in \mathbb{R}^+$ is the angular velocity limit and $\alpha(k) \in [0, 1]$ is the confidence level. This is computed by the action segmentation algorithm explained in Chapter 3 and, thus, it is used to modulate the velocity of the robots depending on the uncertainty in the action recognition. To implement the *collision avoidance constraint*,

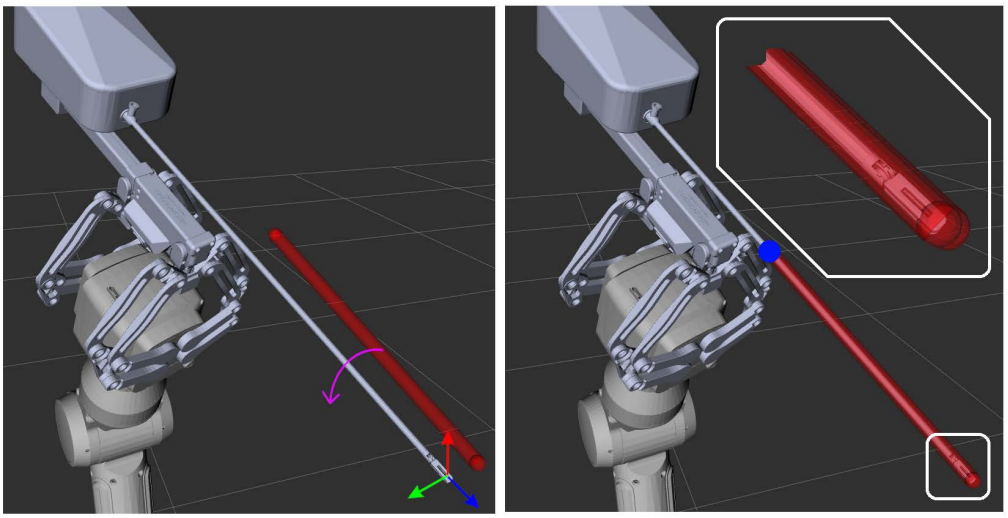

Fig. 4.1: Procedure used to enclose a tool into a capsule. On the left, the CAD model of the robotic laparoscopic tool and, next, a capsule. The x, y, z, θ axis are represented in the image on the left by the colors red, green, blue, and magenta respectively; on the right, the capsule wrapping the robotic laparoscopic tool with a blue dot indicating the RCM.

we use a simplified capsule-like geometrical representation of the laparoscopy tool's shaft and the distances among these capsules are exploited such that the collisions between the$i$-th tool and the j-th tool at time instant k are avoided by setting the following constraint:

$$d_i^j(k) \geq d_s \tag{4.4}$$

where $d_s \in \mathbb{R}$ is a user-defined positive parameter representing the safety distance. Figure 4.1 shows the procedure used to build a capsule around a laparoscopic tool.

Once the geometrical structure has been defined, it is possible to formulate an analytical solution to compute the distance in-between capsules and other geometrical shapes to achieve a sufficient spatial discretization.

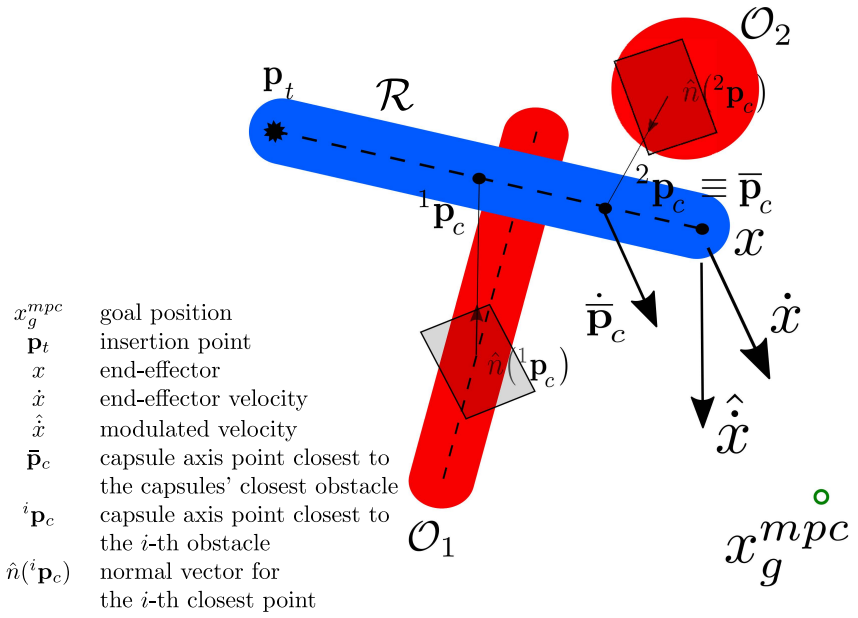

Fig. 4.2: In blue: the robot tool $\mathcal{R}$ modeled as a capsule. In red: a spherical $\mathcal{O}_1$ and a capsule shaped $\mathcal{O}_2$ obstacle.

The distance between two capsules $d_{1,2}$ can be computed easily by subtracting the radii r_1 and r_2 of the capsules from the distance between the two segments representing the axes of the capsules:

$$d_{1,2} = d_{a1,a2} - r_1 - r_2, \tag{4.5}$$

where $d_{a1,a2}$ is the distance between the axes of the capsules computed as explained in [46]. Using Figure 4.2 as reference, it is possible to compute the closest points on the capsules $\mathbf{p}_{c1}, \mathbf{p}_{c2}$ by finding the line that intercepts them and considering the radii r_1, r_2 starting from the closest points on the segments $\mathbf{p}_{a1}, \mathbf{p}_{a1}$ such as

$$\mathbf{p}_{c1} = \mathbf{p}_{a1} + r_1 \frac{\mathbf{p}_{a2} - \mathbf{p}_{a1}}{\|\mathbf{p}_{a2} - \mathbf{p}_{a1}\|} \tag{4.6}$$

$$\mathbf{p}_{c2} = \mathbf{p}_{a2} + r_2 \frac{\mathbf{p}_{a1} - \mathbf{p}_{a2}}{\|\mathbf{p}_{a1} - \mathbf{p}_{a2}\|}. \tag{4.7}$$

The procedure for calculating the distances and the closest points is efficiently computed via a geometric computation library tools such as the NASA Spice toolkit [52] which allows to compute the distance between a line and an capsule, the distance between a point and an capsule and the related closest points for both cases:

- if the line does not intersect the capsule (i.e. the line is at a distance $d_{l,e} > 0$ from the capsule), then the distance between the end-points of the axis of the capsule and the capsule has to be computed; if both of them are greater than $d_{l,e}$, then the distance between the line and the capsule is the minimum and the relative closest points are kept, otherwise the end point with the minimum distance is the closest point of the segment and the relative closest point on the capsule is the correct one;
- if the line intersects the capsule, then one of the end points is the closest point, so for both of them the distance is computed and the minimum one is kept together with the relative closest points.

On these premises, the point on the capsule closest to the obstacle $\mathbf{p}_c$ can be computed as follows:

$$\mathbf{p}_c = \mathbf{p}_a + r \, \frac{\mathbf{p}_e - \mathbf{p}_a}{\|\mathbf{p}_e - \mathbf{p}_a\|} \, , \tag{4.8}$$

where $\mathbf{p}_a$ is the point on the axis of the capsule, $\mathbf{p}_e$ is the point on the capsule and r is the radius of the capsule.

4.2.2 Model-Predictive Control Formulation

Since the task for the controller is to reduce the distance between the current position (and a single rotation angle θ) of the tool $\hat{x}(k+i)$ and the goal $x_g^{mpc}(k)$ at any given discrete time $k \in \mathbb{N}$, the cost function for the MPC $J(x_g^{mpc}, x)$ is, effectively, the Euclidean distance between the two points,

$$J(x_g^{mpc}, x) = \sum_{i=0}^{N_p-1} ||x_g^{mpc}(k) - \hat{x}(k+i)||^2 \tag{4.9}$$

where $N_p = t_c/\Delta T_c$ is the number of time steps in the prediction horizon t_c for the MPC sampling time ΔT_c and $\hat{x}(k+i)$ is the predicted state with initial condition $\hat{x}(k+0) = x(k)$. This cost function allows the system to reach the goal position along a straight trajectory in the absence of obstacles along the path.

The solution of the following constrained finite-horizon optimal control problem

$$\begin{aligned}
&\min_{\mathbf{u}} \textstyle\sum_{i=0}^{N_p-1} ||x_g^{mpc}(k) - \hat{x}(k+i)||^2 \\
&s.t.\ \hat{x}(k+i+1) = \hat{x}(k+i) + Bu(k+i) \\
&\quad |\bar{u_j}(k)| \leq \alpha(k)\, u_j^{max} \\
&\quad |\bar{\omega_j}(k)| \leq \alpha(k)\, \omega_j^{max} \\
&\quad d_{r_r}^{r_l}(k+i) \geq d_s \\
&\quad d_{r_j}^{o_h}(k+i) \geq d_s \\
&\quad i = 0, \ldots, N_p - 1 \\
&\quad h = 0, \ldots, N_o \\
&\quad j \in \{r, l\} \\
&\quad \hat{x}(k+0) = x(k)
\end{aligned} \tag{4.10}$$

returns the optimal control input sequence $\mathbf{u} = [u(k), ..., u(k+p-1)]$. In the optimization problem, $\hat{x}(k+i+1)$ represents the estimation of the state at time $k+i+1$ computed using the model (4.1), $u(k+i)$ is the control input at time $k+i$ and $d_{r_r}^{r_l}(k+i)$ is the distance between the virtual capsule built around the controlled tools at time $k+1$. $d_{r_j}^{o_h}(k+i)$ is the distance between the virtual capsules built around the controlled tool j and the obstacle h at time $k+1$ with N_o the number of obstacles in the workspace. The position of the obstacle in the prediction horizon is computed considering the velocity of the obstacle to be constant over the entire horizon. The same holds for the goal state $x_g^{mpc}(k)$ and the confidence level $\alpha(k)$. Finally, the first component $u(k)$ is used to compute the desired Cartesian position $x_d(k+1) \triangleq x(k) + Bu(k)$ that the robots need to reach.

4.2.3 Waypoint Generation

When the target configuration, that we can denote as x_g, selected by the *supervisory controller* is directly provided as the goal x_g^{mpc}, the MPC alone may not guarantee that the desired configuration is reached. As shown in Figure 4.3, it is possible for the optimization to be stuck in a local minima solution in which all of the constraints are satisfied, but the final target is not reached. This is mostly due to the limitation in DoFs available at the end effector of the tool imposed by the constraint introduced by the trocar point p_t.

A possible solution to this problem consists in properly planning a set of waypoints towards the final target configuration that are provided as intermediate goals to the MPC as depicted in Figure 4.4. By planning an alternative route to be followed, it is possible to

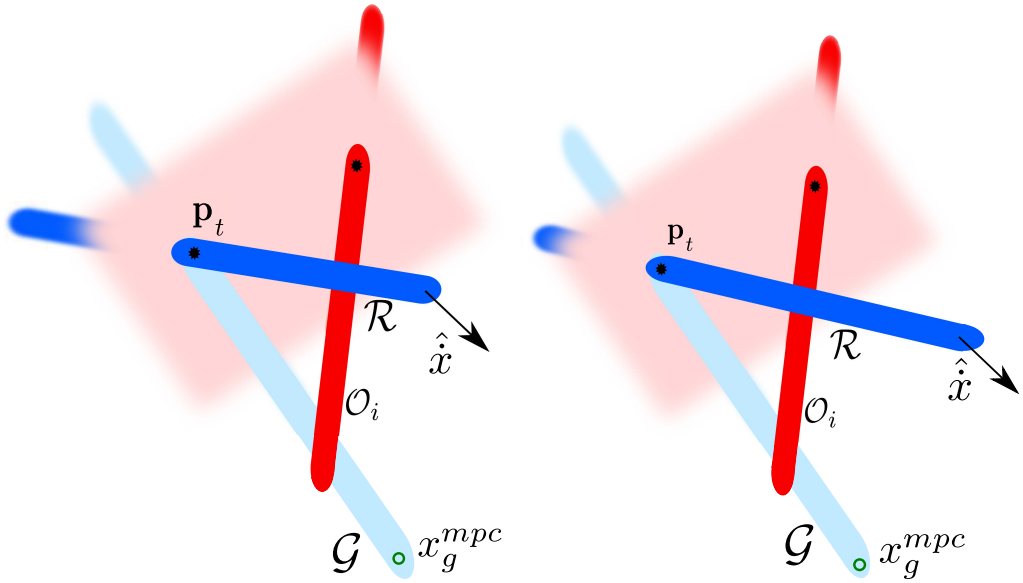

Fig. 4.3: Capsules of the tool position $\mathcal{R}$ (blue), desired tool position $\mathcal{G}$ (light-blue) and i-th obstacle $\mathcal{O}_i$ (red) before (a) and after (b) the insertion movement. The obstacle cannot be overtaken with a direct motion.

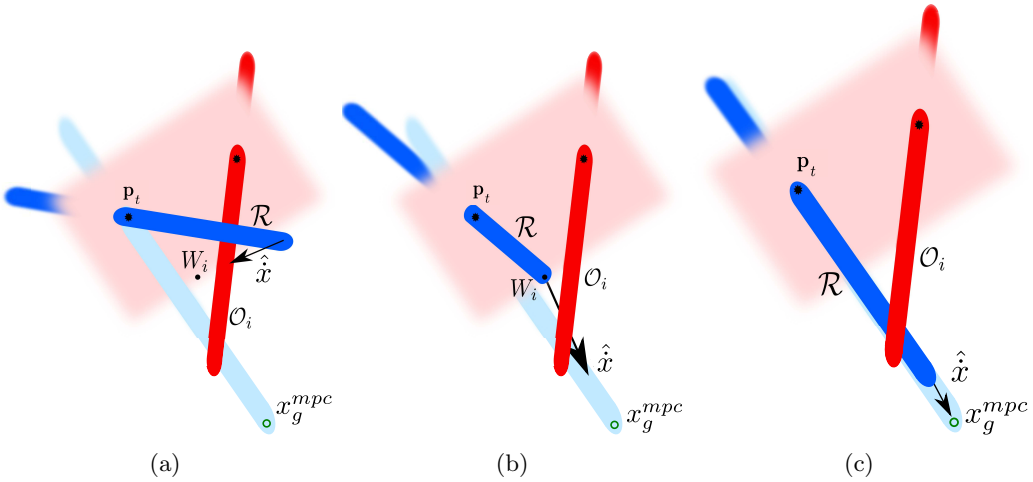

Fig. 4.4: Capsules for the tool position (blue), desired tool position (light-blue) and i-th obstacle (red). The waypoint W_i, (a) and (b), allows to direct the motion around the obstacle (c).

steer the robot along preferential directions to implement obstacle avoidance while taking into account the trocar constraint.

The proposed planning strategy is based on geometric considerations and it is efficiently computable in real time, which is crucial for a reactive behavior of the robotic system holding the laparoscopic tool. The motion of each arm is planned independently to each other through a *first-come, first served* approach: the solution to the motion of the first generated by the MPC is evaluated considering the remaining arms as obstacles. Since all tools and obstacles are wrapped in capsules, we can use $\mathcal{C}(C_1, C_2)$ to indicate a generic capsule, where $C_1 = (x_{C_1}(t), y_{C_1}(t), z_{C_1}(t))$ and $C_2 = (x_{C_2}(t), y_{C_2}(t), z_{C_2}(t))$ are the two end-points of the capsule that uniquely identify its pose in space. The notation $\overline{C_1 C_2}$ indicates the segment from C_1 to C_2.

Algorithm 1: The planning strategy

1 **Data:** $\mathcal{R}$, $\mathcal{G}$,$\mathcal{O}_i$
2 $\mathscr{Q} = \emptyset$
3 $\mathscr{M}$=computeMotionPlane($\overline{R_1R_2}$,$\overline{G_1G_2}$)
4 **for** $i \leftarrow 1$ **to** N_o **do**
5 S_i = Sample($O_{1,i}, O_{2,i}$)
6 Z_i=getClosest(S_i, T)
7 **if** *(isFree($O_{i,1}, O_{i,2}, R_1, \mathscr{M}$))* **then**
8 W_i=$\emptyset$)
else
9 W_i=project($Z_i, \mathscr{M}, \overline{O_{i,1}O_{i,2}}$) addWayPoint($W_i$, $\mathscr{Q}$)
11 W=computeFinalWayPoint($\mathscr{Q}$)

Let $\mathcal{R}(R_1, R_2)$, $\mathcal{G}(G_1, G_2)$ and $\mathcal{O}_i(O_{1,i}, O_{2,i})$, with $i = 0, \ldots, N_o$ be the capsules that identify the tool whose motion needs to be planned, the goal configuration for the robot $\mathcal{R}$, and the obstacles, respectively. The planning strategy is reported in Algorithm 1.

The necessary data are the capsules of the arm $\mathcal{R}$, of the goal $\mathcal{G}$ and of all the obstacles $\mathcal{O}_i$, $i = 1, \ldots, N_o$. For each obstacle, a local waypoint is generated according to the following procedure. The motion plane $\mathscr{M}$, i.e. the plane on which the tool can reach the goal configuration in case of no obstacles, is generated (Line 3). Formally, this is the plane orthogonal to the normal vector

$$n = \frac{R_1 - R_2}{\|R_1 - R_2\|} \times \frac{G_1 - G_2}{\|G_1 - G_2\|} \tag{4.11}$$

and containing $\overline{R_1R_2}$ and $\overline{G_1G_2}$. Then a set S_i of possible escape points from the obstacle is generated by uniformly sampling the space around the endpoint of the obstacle capsule $\mathcal{O}_i$ that is the closest to the tool (Line 5). Among these points, Z_i, the closest to the trocar, is chosen (Line 6) in order to give a preference to the retraction of the tool, which is always a feasible option. If the capsule of the obstacle is parallel to the motion plane, i.e. $(O_{i,1} - O_{i,2}) \cdot n = 0$, but not contained in it, or if the capsule is not intersecting the plane, i.e. $((O_{i,1} + \sigma(O_{i,2} - O_{i,1})) - R_1) \cdot n = 0$ for all $\sigma \in [0, 1]$, then the motion plane is free and the robot can reach the desired configuration. Thus, no local waypoints are generated (Line 8). If the obstacle intersects the motion plane, a local waypoint W_i is generated by projecting Z_i on the motion plane along the obstacle axis (Line 9). In this way, W_i is reachable by the robot and it does not intersect the obstacle by construction. The set containing the local waypoints is updated (Line 10).

The global waypoint is computed as the centroid of the waypoints associated to each obstacle, using as weight of each waypoint β_i the inverse distance d_i between the tool and the related obstacle, $\beta_i = \frac{1}{d_i}$, (Line 11):

$$W = \frac{\sum_{i=1}^{N} \beta_i W_i}{\sum_{i=1}^{N} \beta_i} \tag{4.12}$$

where N represents the cardinality of $\mathscr{Q}$ and $\beta_i = 1/d_i$. Despite the fact that each local waypoint is external to the related obstacle, it is possible (though unlikely) that their centroid, computed as (4.12), could lie inside one of the obstacles: in this case, the waypoint W is set equal to the trocar, thus the MPC will compute an extraction movement, which is always collision free.

4.3 Validation

To verify the feasibility of the proposed control architecture to provide effectiveness of operations while maintaining a safe distance from defined obstacles, its validation has been performed on both a virtual and physical SARAS platforms. The former employs the *computer-*

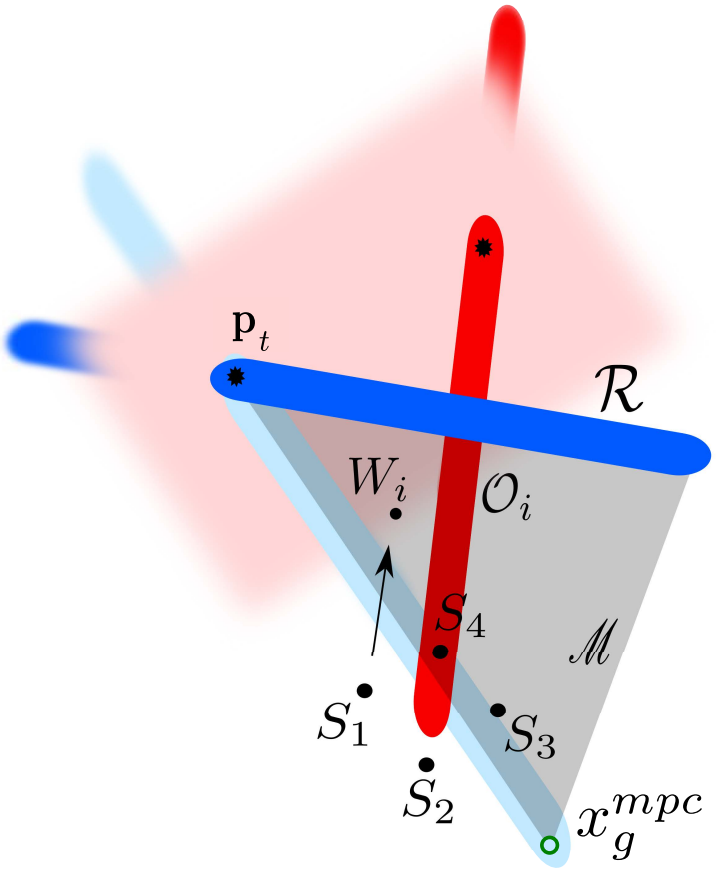

Fig. 4.5: Tool position (blue), desired tool position (light-blue) and capsule-shaped obstacle (red). The point S_1 is chosen to be projected on the lane $\mathscr{M}$ (grey) to find the waypoint W_i

aided designs (CADs) and the low-level characteristics of all physical robots involved to provide an accurate dynamical feedback of the induced motions. The scene sees, specifically, four laparoscopic tools operate in a shared workspace: two of them are mounted on the SARAS robots and controlled by the autonomous system, while the other two, the daVinci® tools, are teleoperated by the main surgeon (in the simulated environment, the surgeon's tools are not teleoperated, but controlled by pre-determined trajectories). The daVinci® tools are considered obstacles to be avoided by the planning algorithm not to disrupt in any way the surgeon's primary activities. The control system moves the controlled tools in order to reach the target point while avoiding collisions between all the tools inside the environment. The obstacles are both static and in motion, moved by a trajectory designed to intercept the SARAS tools and disrupt their position-keeping goal. Even though the experimental setup does not include a realistic environment to perform the experiments, the validity of the approach can still be evaluated in terms of goal achievement and reduced workflow impact.

4.3.1 Validation in simulation

To simulate the real movement of the robot and faithfully reproduce the real setup, a visual model and a kinematic simulator of the SARAS arms were created using the Robot Operative System (ROS) [24] and the design models of the robot as shown in Figure 4.6. The daVinci® arms were simulated with by inserting in the virtual environment the position of the RCM and of the end-effector for both arms with their respective virtual capsules.

Simulations are performed by providing to the system only the goal configuration x_g, the initial configuration of the two arms $\tilde{x}_r, \tilde{x}_l$ and the initial configuration of the two obstacles.

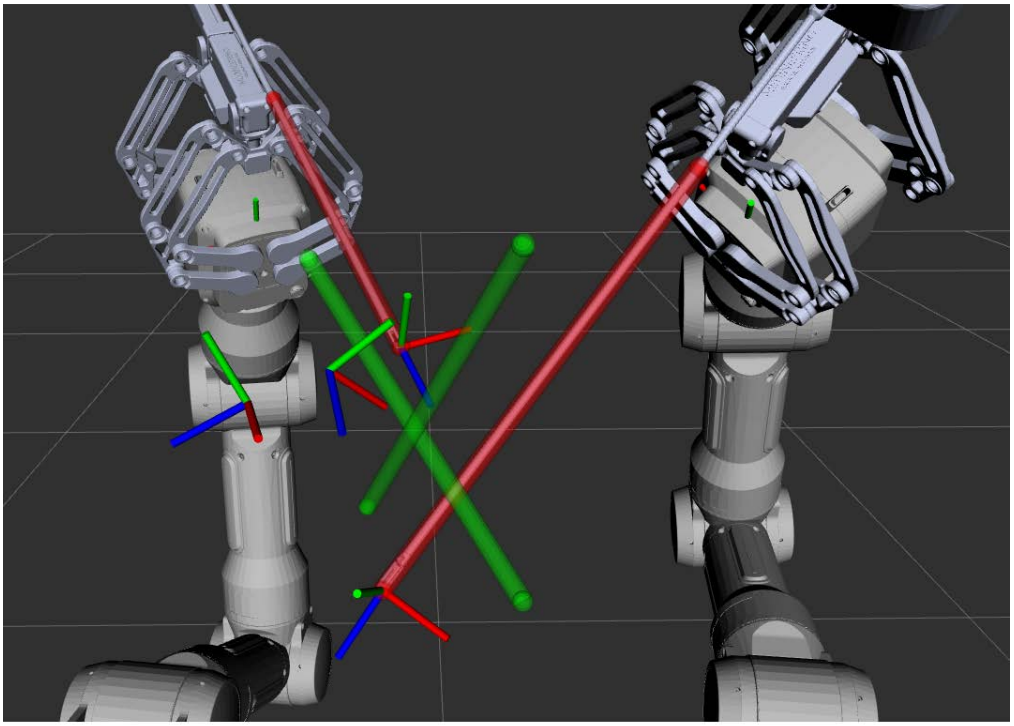

Fig. 4.6: Simulation environment. A SARAS arm model is placed on the right side and on the left side. In red, the virtual capsules wrapping each arm's tool. In green, the virtual capsules wrapping the obstacles. The frames at the end of each arm represent the pose of the end-effector while the other two frames represents the goal positions.

In the following graphs, to improve readability, only the movement along one of the Cartesian axis is reported. The real Cartesian position of the arms is not reported in the plots since the simulator is purely kinematic, i.e. there is no difference between the commanded position and the real position. Figure 4.7 shows the results achieved using the simulation environment. The magnitude of the Cartesian velocity vector (Figure 4.7-a) of the two controlled arms shows the effect of the modulation introduced by the confidence level α, reported in Figure 4.7-b, which forces the controller to scale down the velocities if an uncertain situation is detected, which has been set to to happen at time $t = 13s$. Figure 4.7-c shows the corresponding evolution of the Cartesian positions. The position of the waypoint switches during the simulation in order to allow the SARAS arms to avoid the obstacles (interval $t \in [0; 25]s$). The waypoint is automatically set to the target position if no obstacle is present along the path (i.e. no intersection between capsules is detected on the plane (M)), as for the right robot, where the waypoint and the target position overlap for the entire simulation. Since the waypoint position is used as reference for the MPC controller and the waypoint position converges to the target position, the overall controller allows the system to reach the objective. Thanks to the MPC controller and the waypoint motion strategy, the controller is able to perform all the movements avoiding collision between tools, as clearly visible in Figure 4.7-d where the distances between the tools are reported. A peculiar behavior of the controller can be observed from Figure 4.7. Indeed, in the first few seconds of the simulation, the right robot reaches the target position and starts to track it, as visible in Figure 4.7-c. At that time, the right robot moves in order to allow the left robot to reach its goal position. Indeed, the distance between the two arms goes to the minimum allowed distance, as visible in Figure 4.7-d. Then, a new configuration is computed for both the robots in order to minimize the distance from the target position.

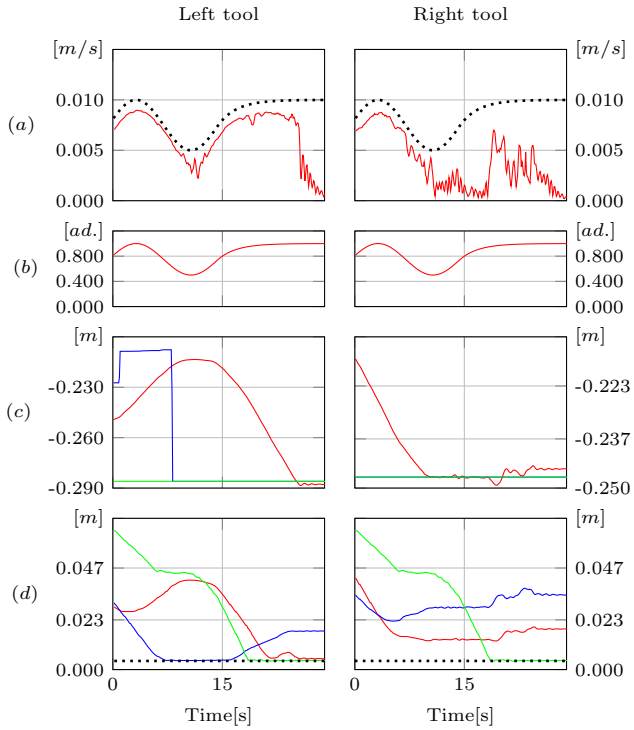

Fig. 4.7: Simulation results. (a) The norm of the controller output velocity (red line) and the confidence modulated maximum velocity norm (black dotted line). (b) The confidence level. (c) The commanded Cartesian position of the robot (red line), the Cartesian position of the waypoint (blue line) and the Cartesian position of the target position (green line). (d) The distance between the robot tool capsule and the first obstacle capsule (red line), the distance between the robot tool capsule and the second obstacle capsule (blue line), the distance between the robots tool capsules (green line) and the minimum allowed distance between tools (black dotted line). Plots are reported for both left and right robots.

Simulation tests have also been performed with moving obstacles (Figure 4.8) to validate the local waypoint algorithm and the response of the MPC optimization. The obstacles pivot at a constant tool-tip linear velocity of 2 $\frac{\text{mm}}{\text{s}}$ to interfere with the initial planned waypoint; the different geometrical alignment of the tools forces the re-evaluation of a new waypoint. A direct comparison of Figure 4.7 and 4.8, subfigures (c) and (d), illustrates the adapted control strategy to the moving obstacles as the obstacle closes in to the moving tools, thus forcing a different trajectory.

4.3.2 Validation on the SARAS setup

The experiments on the robotic platform were performed by providing the system with the goal configuration x_g. The configuration of the two arms $\tilde{x}_r, \tilde{x}_l$ and the configuration of the two obstacles are continuously updated using the robots' readings. The two controlled arms and the two obstacles arms are intentionally positioned in such a way that each controlled arm needs to overcome an obstacle. Figure 4.9 shows the relative position of the robots for the experiment while Figure 4.10 shows the results achieved using the setup and confirms the results obtained in simulation. Figure 4.10-a reports the Cartesian velocities of the two controlled arms while Figure 4.10-c reports their Cartesian positions. It is worth highlighting that the noise in the velocities in Figure 4.10-a is due both to the numerical derivation of positions measured by potentiometers (and not encoders) and by the fact that the RCMs on the SARAS platform are imposed virtually via software and no trocar has been placed to provide a physical constraint. When the robots are correctly positioned in the trocar, the

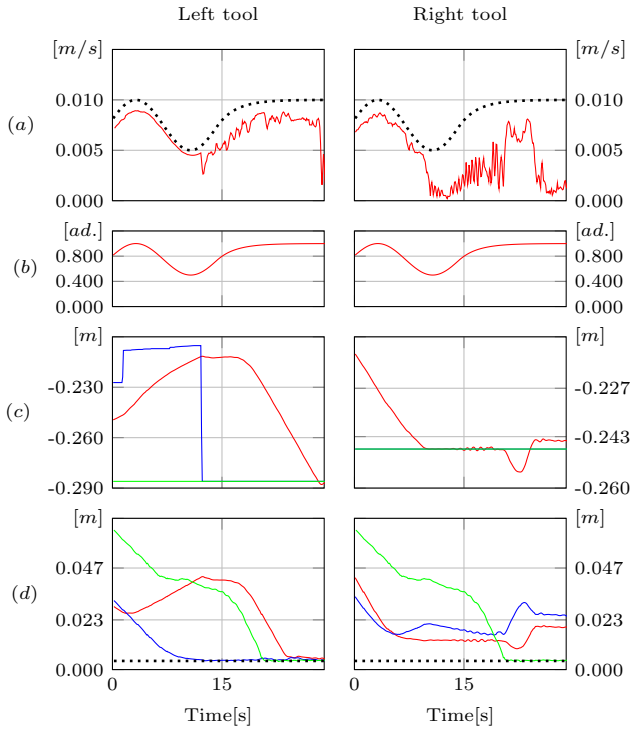

Fig. 4.8: Simulation results with moving obstacles (refer to the caption for Figure 4.7).

Fig. 4.9: SARAS experimental setup: the two white arms in the foreground are the novel laparoscopy arms being developed for the SARAS project; the endoscope (in the middle) and the two robotics arms in the background are the daVinci® surgical system arms connected to the DVRK [32] platform

shaking of the slender laparoscopy tools would be drastically reduced. Good tracking performance can be appreciated by looking at the small difference between the the commanded Cartesian position and the real Cartesian position (represented by red and orange lines in Figure 4.10-c). This shows that the robots implementing the MPC commands reach their target positions while avoiding the obstacles, thanks to the waypoint motion strategy. All of the movements are performed avoiding collisions, as clearly appreciable in Figure 4.10-d,

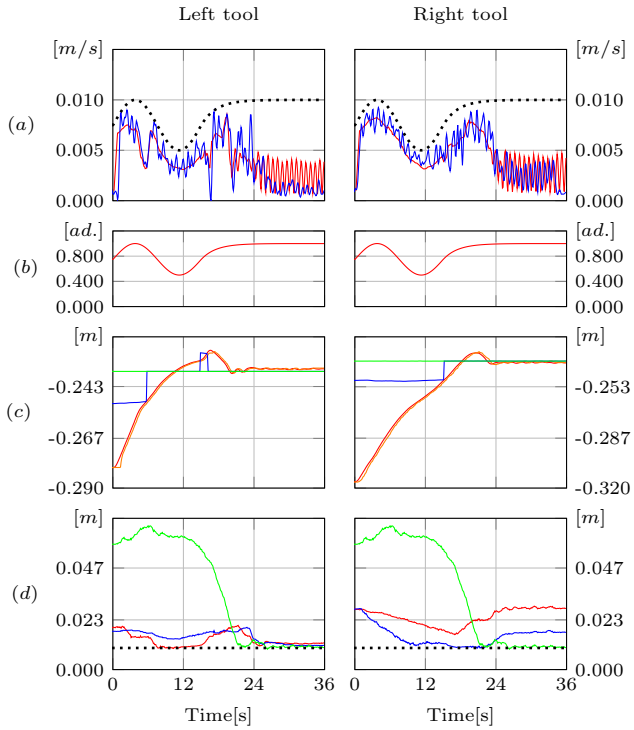

Fig. 4.10: Experimental results. (a) The norm of the controller output velocity (red line), the norm of the real velocity of the robot (blue line) and the confidence modulated maximum velocity norm (black dotted line). (b) The confidence level. (c) The commanded Cartesian position of the robot (red line), the Cartesian position of the waypoint (blue line), the real Cartesian position of the robot (orange line) and the Cartesian position of the target position (green line). (d) The distance between the robot tool capsule and the first obstacle capsule (red line), the distance between the robot tool capsule and the second obstacle capsule (blue line), the distance between the robots tool capsules (green line) and the minimum allowed distance between tools (black dotted line). Plots are reported for both left and right robots

and modulating the velocities with respect to the confidence level provided to the MPC. The same position was commanded as goal configuration for the two controlled tools. In Figure 4.10, it is possible to see how the position of the waypoints is computed as an intermediate point for both the controlled arms. When the tool has reached the waypoint, the latter switches to the target position, thus driving the robot towards the goal configuration since obstacles are assumed to have already been overcome. Since the goal configuration is the same for both the tools, neither of them reaches the target position since a collision would occur. The system converges to a configuration where the distance between the actual position and the desired one is minimized but collisions are avoided, as observed also in simulation.

4.4 Discussions

The model-predictive controller implemented to move the autonomous arm represents a clear improvement over state-of-the-art controllers that do not consider the unpredictable motions of laparoscopy tools controlled by human operators inside highly restricted environments. The control of the robot holding the tool is indeed challenging given the constraints and the performance requirements. Indeed, the use of system adopting MPC formulations with multiple constraints has been possible only with the advancement of computational performance and the development of the fast analytical solutions adopted in this Chapter.

The results obtained in both simulation and on the experimental platform demonstrate that the control system is capable of delivering a responsive control for the SARAS robotic platform. This claim is additionally tested in the next Chapter in which both the *action segmentation* and the *model-predictive controller* will be integrated in the cognitive control system to execute a surgical cooperation task between a human and a robot.

5

A Semi-Autonomous Surgical Robot

5.1 Introduction

This chapter integrates the technologies analyzed in Chapters 3 and 4 in a system designed to execute a laparoscopic cooperative task in a semi-autonomous manner using one tool of the SARAS platform in cooperation with one of the daVinci® minimally-invasive tools controlled by a human operator. A proposal for the classification of autonomy grade in a surgical system [**yang2017autonomy**], identifies five progressive levels:

- Level 0: no autonomy. The robot is fully teleoperated.
- Level 1: robot assistance. The robot provides support during teleoperation, such as virtual fixture or assisted guidance.
- Level 2: task autonomy. The robot can perform autonomously specific task initiated by the user, i.e. the user determines which task has to be performed and where.
- Level 3: conditional autonomy. The robot can generate autonomously different strategies to perform a task an the user decides which one should the robot apply.
- Level 4: high autonomy. The robot can take decision on the task to be performed in the surgery but under the supervision of the user.
- Level 5: full autonomy. The robot can perform autonomously the entire surgery.

Within this scale (reported in Figure 5.1), 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 cognitive architecture of the cognitive control has been formalized in our previous work [**DeRossi2019IROS**]. This paper presents a system designed to fulfill the requirements for completing a laparoscopic pick-and-place cooperative task in a semi-autonomous manner using the novel SARAS robotic minimally-invasive tools.

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 [37, 9].

Figure 5.2 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 $\left(\hat{\mathcal{I}}, \hat{x}, \hat{\xi}\right)$. 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.

This chapter is based on the paper "Cognitive Robotic Architecture for Semi-Autonomous Execution of Manipulation Tasks in a Surgical Environment", presented at *IROS 2019*; the chapter has been prepared for submission to the *Robotics and Automation Magazine* (RAM) under the title "A Multi-Modal Learning System for Action Segmentation and Control of Surgical Robots".

Fig. 5.1: Autonomy levels mapped to robotic surgery [79].

The entire system represents, therefore, a seamless integration of perception, decision, planning and action in a simplified but still realistic and challenging surgical scenario.

The foremost attention has been dedicated to the design of an Action Segmentation neural network: it uses multi-modal learning capabilities over image data and kinematic trajectories of the robots to provide high level 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 providing the robot with the required goal points and grasp directives.

5.1.1 A Semi-Autonomous Cooperative Task

When it comes to R-MIS applications, there is a shortage of readily available datasets with full labelling to construct a reliable benchmark for gesture segmentation. Currently, only the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [2] dataset has been extensively tested with a variety of approaches. For the specific problem of recognition of the assistant surgeon's actions, no dataset has been proposed yet in the relevant literature. Validation is, therefore, performed by completing a in-house 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 (Figure 5.3). 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. In view of the considerations and definitions discussed in Chapter 2, the actions performed in this custom dataset can be classified as *surgémes* for the pick-and-place task, even though the task itself generalizes over proper surgical gestures primitives.

Each data acquisition session was prepared with the intent of avoiding overfitting by excessive duplicates: both the orientation of the target square shown in Figure 5.3 and

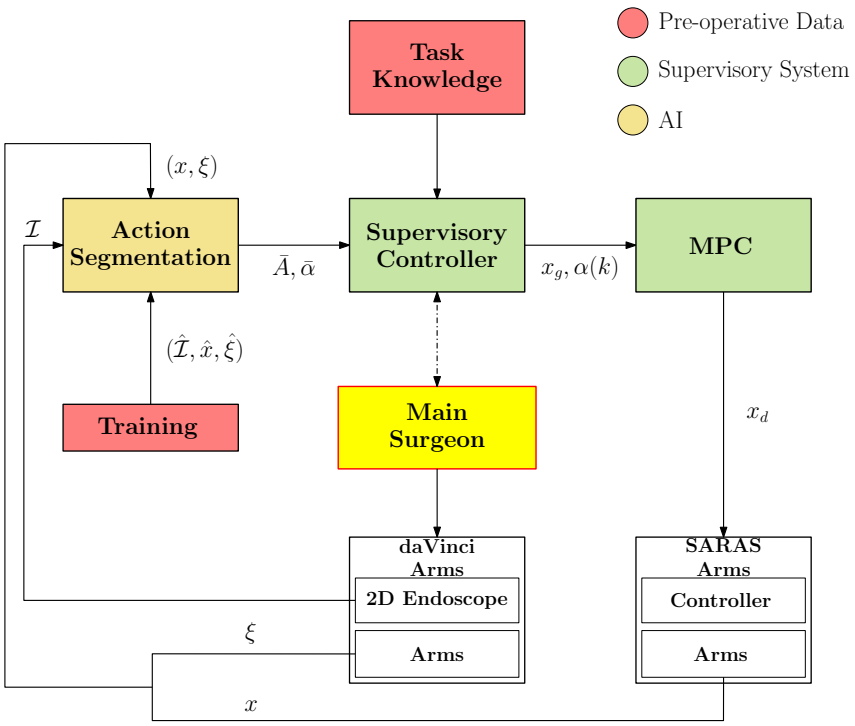

Fig. 5.2: Control architecture schematics. The dashed line indicates an event-based information stream between the Supervisory Controller and the Main Surgeon, i.e. an user input request after displaying an error condition.

(a) (b)

Fig. 5.3: Experimental setup. (a) daVinci® arm (right) and SARAS arm (left). (b) same scene seen through the left endoscope camera.

the initial position of the ring were randomized. Moreover, the final dataset contains five recordings per ring color to provide a sufficient and differentiated amount of data to the learning process. The labelling, shown in Table 5.1, has been divided into 8 different fine-grained actions for the main surgeon (MS) and the assistant surgeon (AS)

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 the SARAS, autonomously, brings the ring to the target area and moves away (actions A06 A07 A08).

The neural network for action segmentation has been trained on a customized dataset of videos acquired using the setup shown in Figure 5.3. In the data acquisition phase, both the

Table 5.1: Action labels for the customized semi-autonomous pick-and-place experiment.

Label	Action
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

daVinci and SARAS arms were teleoperated by two technicians. The 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 [24]. 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. To facilitate the training phase, the parameters have been initialized with weights from the *ImageNet* competition [12].

5.2 Neural Network Specifications

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.

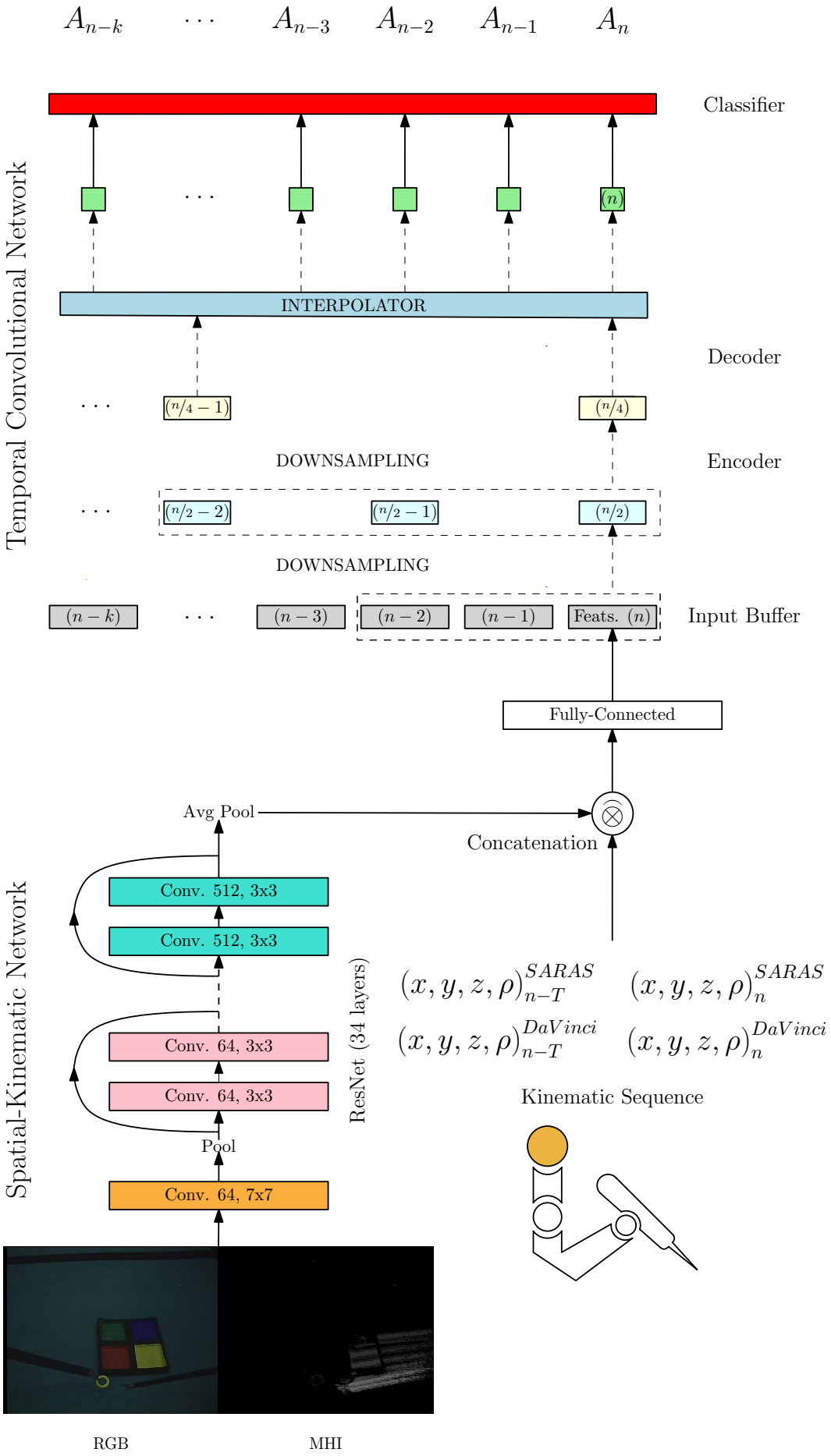

Fig. 5.4: Neural network schema for action segmentation: the RGB and MHI images are processed simultaneously as a 4-channel enhanced frame.

The resulting neural network architecture, called *EdSkResNet* (Figure 5.4), is composed of two sub-networks: the *Spatial-Kinematic Network*, which produces high-level features by processing image and kinematic data, and the *Time-interpolated Fully Convolutional* network presented in Chapter 3.4, 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* [31]) 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.

Multi-modal learning has been introduced to further enhance the capabilities of *ResNet* for the specific problem of action segmentation. In parallel with the RGB frames, the network processes

- an additional image channel called the *Motion History Image* (MHI), a solution also adopted in [1, 43], implemented as a decay factor that weights more recent and older grayscale frames over a temporal window T;
- a sequence of kinematic positions, 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. Combining 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 features computed over time are then pushed into a circular buffer to be processed within the Temporal Convolutional Network; the buffer is designed not to interrupt the training of the neural network from the input images and kinematics. The benefits of the temporal network are twofold:

- it stabilizes the output relative to input changes, which has a considerable impact for online use;
- it allows to obtain a prediction horizon by simply shifting the temporal output sequence during training.

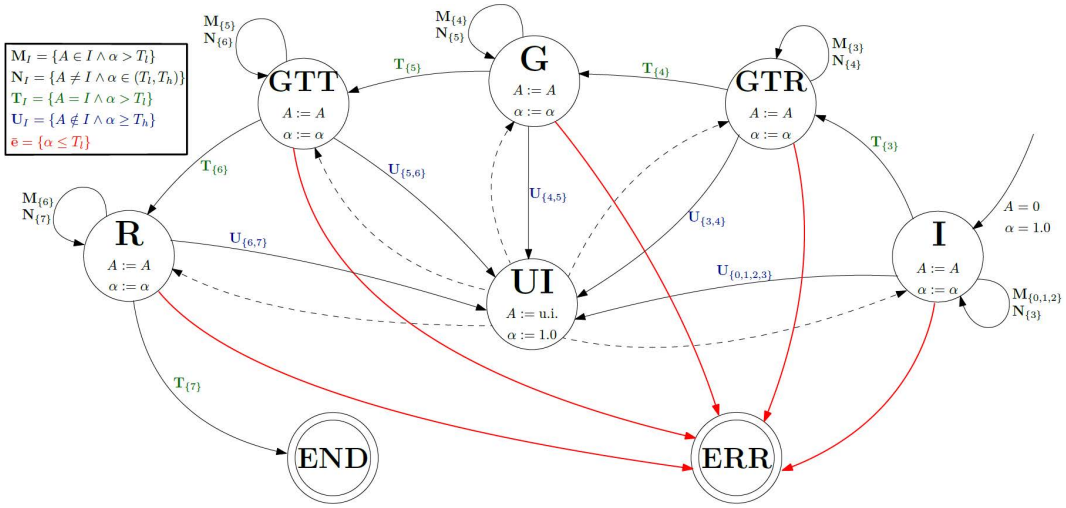

Fig. 5.5: Scheme of the hybrid automaton supervisor

Even though the prediction results were not used explicitly in this work, they demonstrate the *shift-invariance* [77] capability of this network architecture and represent another proof of how the overall system is optimized for processing streaming data. 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) (Figure 5.2), is determined by a *supervisory controller*.

For the proposed experimental scenario, the supervisory controller can be implemented as the *hybrid automaton* shown in Figure 5.5. It differs to a classical *Finite-State Machine* from its dependency on the time-varying variables $\bar{A}$ and $\bar{\alpha}$. Three distinct thresholds to control the next-state function of the automaton have been defined: firstly, the lower threshold T_l representing the minimum value for trustworthiness below which the network is producing defective results (i.e. below the random extraction probability); secondly, the higher threshold T_h, as the minimum level of confidence for which it is assumed that the network has identified the action with sufficient accuracy; finally, the M_{tol} which discriminates over the amount of time the segmentation output remains within the two confidence thresholds ($T_l \leq \alpha(k) \leq T_h$). 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 fulfil 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 for too long below the threshold T_h.

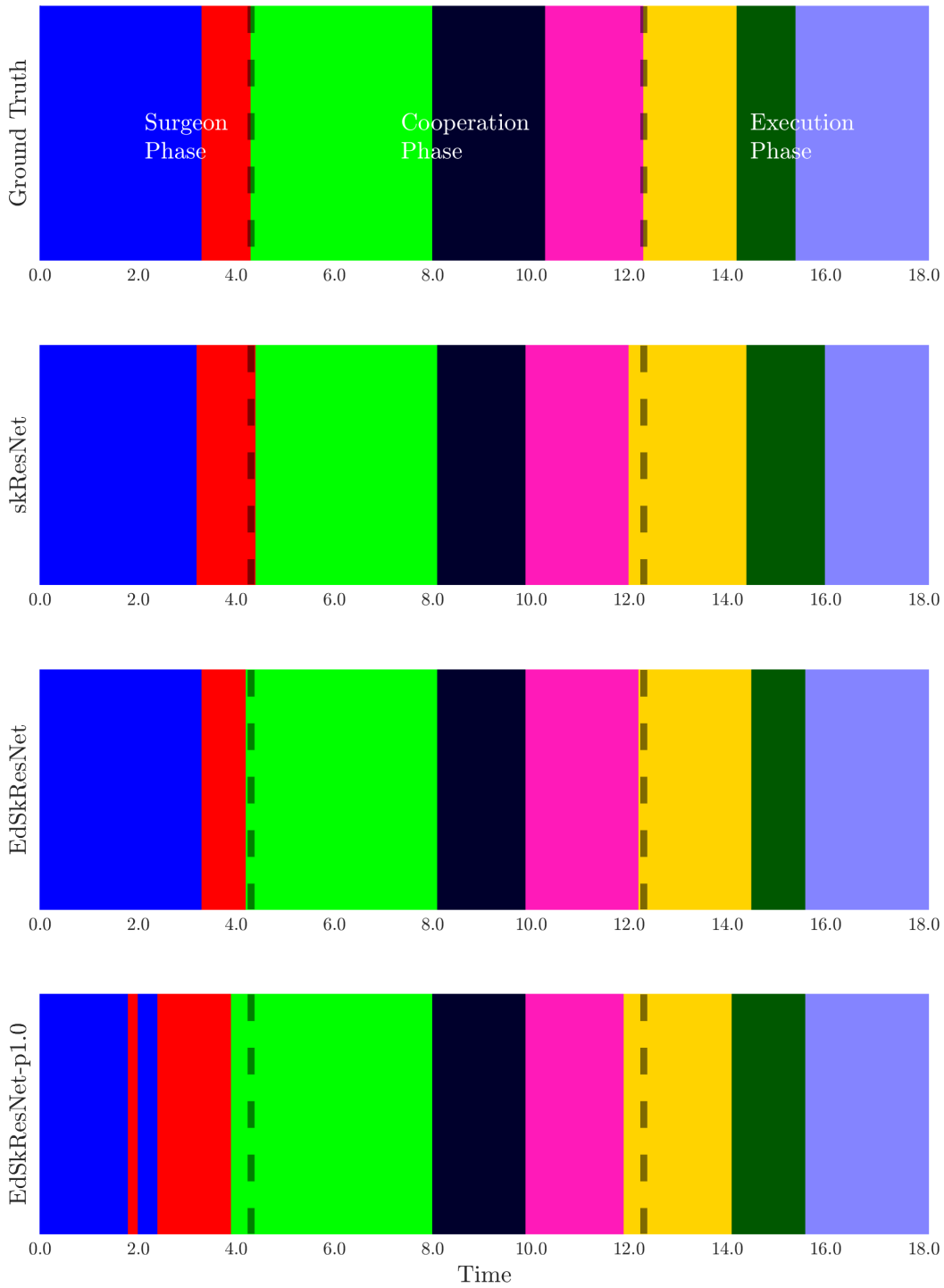

Fig. 5.6: Segmentation graphs for kernel size $k_t = 60$ performed on RGB + MHI enhanced images and kinematic data. From the top: the ground truth labelling; the results respectively without (*skResNet*) and with (*EdSkResNet*) temporal filtering. The bottom plot is the estimate via *EdSkResNet* with a look-ahead horizon of 1.0 seconds.

5.3 Ablation Study

The full neural network (*EdSkResNet*, Figure 5.4), has been split into its various sub-networks which have been tested separately to identify the single contribution of each one to the overall result, thus verifying the validity of using the complete structure over singular components. These sub-networks are:

- *kClass*, a simple kinematic classifier composed of two fully-connected layers (a similar structure has been tested also in [26] for the JIGSAWS dataset);
- *ResNet*, the standard RGB-only ResNet34 image classification network [31];
- *sResNet*, the ResNet34 computed over the RGB + MHI enhanced frames (which is similar to [42]);
- *skResNet*, the sResNet with the addition of kinematic sequences.

The results of each network have been evaluated using the same three statistical indices presented in Chapter 3, Section 3.4:

- the *Accuracy Score*;
- the *Edit Score*;
- the F_1 *Score*.

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 results have been maintained to improve generalization in online usage where conditions can differ w.r.t. the data acquisition ones. The best results have been obtained using a 2.0 seconds history time window for both the kinematic position trace and the MHI. Table 5.2 reports the median results for each score and network topology when segmenting at the latest timestamp (without a prediction horizon). 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 spatialkinematic network alone;
2. improved continuity and stability in recognition when used online for controlling the robot

Table 5.2: Ablation studies results (%).

Network	**Accuracy**	**Edit Score**	$\mathbf{F_1}$
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

Table 5.3: Look-ahead labelling on EdSkResNet (%).

Horizon	**Accuracy**	**Edit Score**	$\mathbf{F_1}$
0.5 *s*	87.29	88.89	87.86
1.0 *s*	85.63	88.89	86.22

The scores presented in Table 5.2 are better presented in Figure 5.6. It shows the sequence of actions as color boxes encoded following the convention in Section 5.1.1: most notably, the segmentation around the critical phase changes, indicated in the figure by the black dashed 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 apt to be used as an on-line soft-sensor in critical applications.

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, within 1.0 s the overall segmentation quality remains acceptable according to both metrics (as shown in Table 5.3 and Figure 5.6). 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; the MPC would still evaluate the prediction horizon at each computation cycle to properly update all command velocities. It is possible, when needed, to quickly evaluate a look-ahead prediction by simply changing the temporal shift for the labels and re-train only the Temporal Convolutional Network.

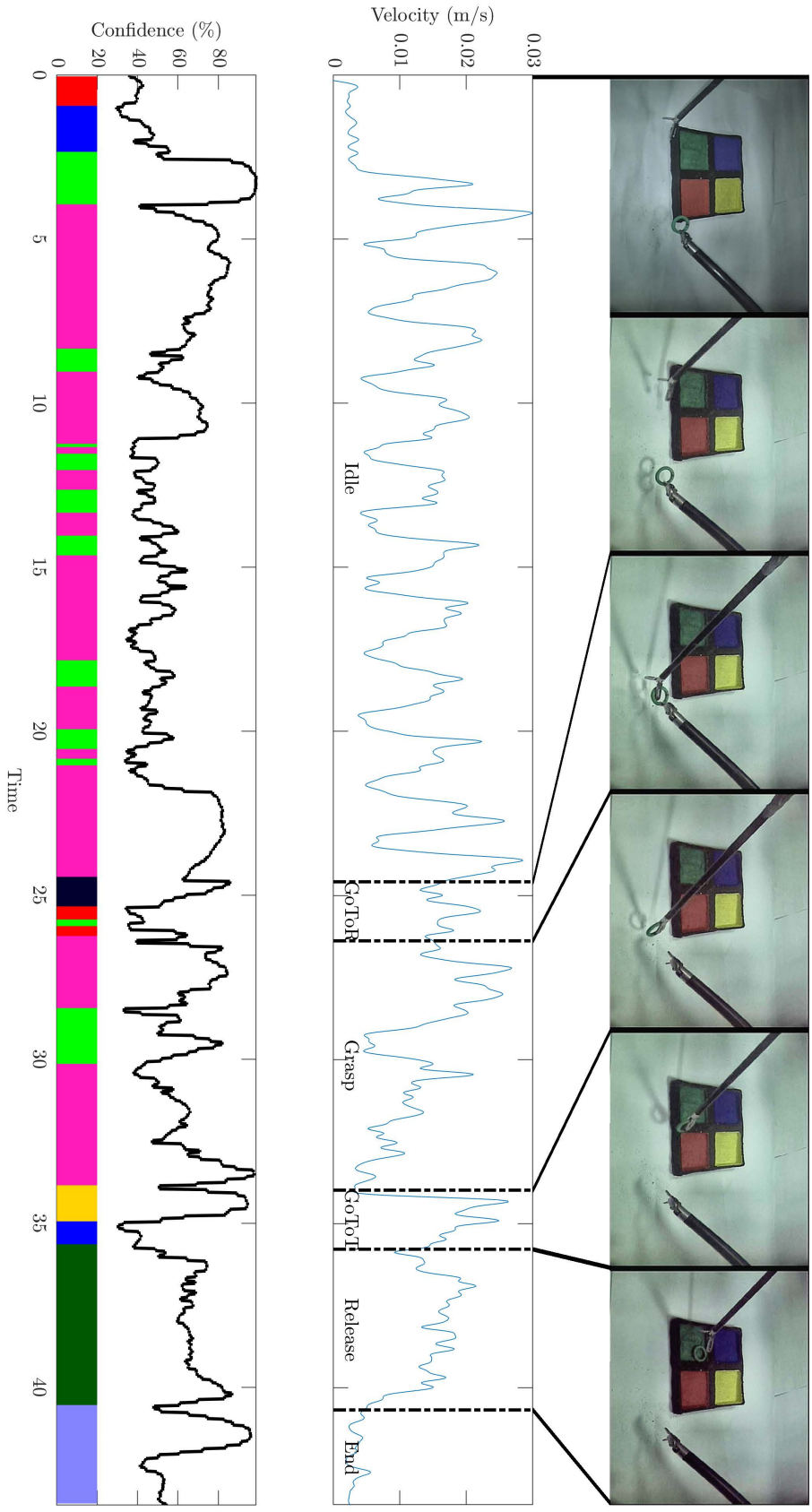

Fig. 5.7: Plot of an experimental task instance performed autonomously by the SARAS arm: the middle plot shows the norm of the Cartesian velocity vector with superimposed automaton states; the bottom plot shows the confidence level with the corresponding identified actions.

The combined contributions of action segmentation, supervisory controller, and model-predictive controller allow the cooperation to be completed successfully. The autonomous arm is capable of understanding 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), in which it is of greater concern to always maintain high performance, the action segmentation delivers the required performance, in both recognition and confidence, to complete the task. Figure 5.7 shows the full on-line execution of the task. At the top plot, the view from the endoscope camera velocity profile of the SARAS arm response to the optimal input velocities produced by the MPC. In the middle plot, the states of the automaton are superimposed over the velocity profile, computed as the magnitude of the Cartesian velocity vector; 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 color convention of Table 3.1. The relationship between the automata states and the recognized actions is evident since the autonomous robot reacts to the correct perceived user action. It is worth noticing how the confidence modulation affects the maximum velocity during the robots' 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 Main Surgeon, the system recognizes action A04 (at approximately second 25) and the Assistant Surgeon 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.

Tests has been conducted 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.

5.4 Discussions

Given the simple task to be executed for the validation, the timing precision of the real-time action segmentation is not critical for the correct execution. Nevertheless, since the intended goal for this algorithm is to perform complex tasks in high-risk scenarios, the uncertainty 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 5.2, once an empirical choice has 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 is poised to introduce segmentation noise. The output stabilizes with the introduction of the *Temporal Convolution Network*, especially for online evaluations as presented in Figure 5.7.

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. This introduced uncertainty especially in-between actions A04 and A05, which lead to the oversegmentation among their respective time periods. The uncertainty, however, necessarily reduces the confidence level, which allows a careful thresholding in the FSM to avoid spurious activations. Therefore, we can assert that, under all conditions, the constrained 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 operating as a safe reactive cognitive system. Surely, one of the main issues

with classification-based methods based on neural networks is the separation between true and false positives (as shown also in the confusion matrix presented in Chapter 3.4: due to the processing within the hidden layers, it is difficult to effectively direct the results via combinations of regularization techniques and validation strategies.

Table 5.4 shows the average confidence for true positive and false positive predictions over seven online experiments. It empirically validates the claim that the gesture segmentation tends to produce higher confidence values whenever the gesture classification identifies the correct instance. A higher number of experiments is of course necessary to obtain statistical validity, yet the trend appears to indicate how the gesture segmentation module correctly separates true and false positive identifications thus making it viable to supervise the results through *a posteriori* confidence evaluation. An increased dataset for training, which would contain more diverse representations of the task, will result in a more pronounced separation between positives and negatives and, therefore, an increased classification robustness.

Table 5.4: Average confidence for true positive (correct) and false positive (incorrect) online action identification.

Experiment	Avg. TP Confidence (%)	Avg. FP Confidence (%)
1	0.74	0.56
2	0.68	0.57
3	0.62	0.54
4	0.67	0.58
5	0.71	0.54
6	0.63	0.56
7	0.67	0.59
Avg.	**0.67**	**0.56**

The dataset used for the pick-and-place task is, clearly, a simplified representation of an actual surgical task. Therefore, the complete network structure has been tested also with the JIGSAWS dataset. The results are only preliminary as they contain a test over only six users out of eight. However, the results for the *EdSkResNet* have been obtained with a strictly causal formulation opposite to the approaches that employ non-causality in the temporal filtering. Table 5.5 reports the results of the *TiFC* network and includes results that use kinematic or multi-modal data to segment gestures on the dataset.

Table 5.5: Results for the *EdSkResNet* over the JIGSAWS dataset compared with the state of the art. † indicates that the result has been obtained with non-causality formulations; (*) indicates usupervised, non-causal, kinematic features only; (**) indicates unsupervised, non-causal, RGB and kinematic features.

Algorithm	**Accuracy**	**Edit Score**	**F_1**
CRF-DPM(a) [41]†*	65.25	n.a.	n.a.
Soft-UGS [18]†*	73.5	75.8	67.4
CRF-DPM(b) [41]†**	67.38	n.a.	n.a.
MsM-CRF [27]†	77.29	n.a.	n.a.
ED-TCN [42]†	81.4	83.1	87.1
TricorNet [16]†	82.9	86.8	n.a.
TDRN [44]†	84.6	90.2	92.9
TIFC†	81.97	86.92	91.1
EdSkResNet	81.71	91.74	80.08

Many works available in literature present liabilities regarding their use in control applications. For instance, the algorithms in [41, 18, 27] cannot operate in an incremental matter

and, thus, do not respect the causality requirement; the remaining works, which are based on neural networks, are theoretically computable in a causal manner, but the available results are obtained by relaxing such requirement during operations. A test performed on the code provided by [42] by adopting the required convolution operations revealed a significant drop in performance of around 10% less across all scores. Therefore, the comparable *Accuracy* score and the higher *Edit Score* obtained in this preliminary comparison show how the improved features that employ multi-modal learning on motion-enhanced RGB and kinematics is beneficial to the temporal segmentation of surgical gestures. The anomalous F_1 score can be attributed to a temporal shift of the overall segmentation which is in turn usually driven by an averaging of all action lengths as performed by the operator.

6

Conclusions

Artificial Intelligence applied to surgical procedures has undoubtable immense potential to revolutionize patients' treatments when combined with robotics platforms to physically perform the tasks. On the other hand, the use of AI and machine learning techniques has to be limited as much as possible to solve issues for which either there exist no other solution or the solution available do not provide the necessary performance. This work has been conducted with the idea of adhering to what is known in autonomous driving applications as the *engineering stack*, a combination of well-known and well-proven control strategies that encompass autonomous reasoning and learning. This solution is in clear opposition to *full stack* (or *end-to-end*) AI applications that intend to, as the name suggests, replace the entire system architecture, from the sensing to the actuation, with machine learning.

The development of the engineering stack revolves around the *gesture segmentation* algorithm designed to provide the system with a reading on the gestures being performed by the surgeon to try to provide a fine-grained assistance. At first, the studies conducted on the *Time-interpolated Fully Convolutional* neural network provided the architecture with an efficient temporal filter for generic feature data. Secondly, the development of the *Model-Predictive Controller* that modulates the robots' control velocities on both the segmentation confidence and the safety distance between laparoscopic tools assured the safety of control through soft-constraints. Finally, the integration of both technologies using a *hybrid automaton* as *Supervisory Controller* assured a strict correlation between lower-level gestures and phases of the operation during the experimental trials.

Throughout this thesis, the test environments have been mostly simplified scenarios in which existing robotic platform, which are, *de facto*, advanced manipulators completely in control of surgeons located in the operating room via teleoperation, have been provided with the capability of processing higher-level data towards their future deployment as *cobots*. The transfer to more complex environments of the gesture recognition modules that try to naturally interface the surgeon and the robot clearly requires additional adjustments to the hyperparameters and the acquisition of training data. Nevertheless, an increased focus in the identification of proper semantic classes within the surgical procedure and the combination of multiple, specific AI technologies to tackle the peculiarities of each individual class, will surely result in a more robust system for human-robot interaction in hazardous environments.

In consequence, all of these systems will necessarily perform, at best, to the best knowledge of the operator performing (and annotating) the task, as this remains the greatest source of bias in the data for classification-based machine learning techniques. A possible solution could be found in advanced prototype works for cooperative reinforced learning, where the data acquisition phase is itself a coordinated process between a human operator and a reinforcement learning agent that hints and corrects proactively possible bias in annotations.

However, such systems will remove full control from the hands of the surgeon, the ethical consequences of which transcend the scope of this work: this thesis improved and combined over the most effective and state-of-the-art technologies already available into a novel cooperative robotic platform to act as an assistant to surgeons that are kept in full control of

laparoscopic manipulation tasks. Semi-autonomous robotic assistants is considered by the author as a necessary intermediate step towards the definition and implementation of fully automatized surgical procedures.

References

[1] M. A. R. Ahad. *Motion History Images for Action Recognition and Understanding*. SpringerBriefs in Computer Science. London: Springer London, 2013. ISBN: 978-1-4471-4729-9.

[2] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager. "A Dataset and Benchmarks for Segmentation and Recognition of Gestures in Robotic Surgery". In: *IEEE Transactions on Biomedical Engineering* 64.9 (2017), pp. 2025–2041. ISSN: 1558-2531.

[3] S. Bai, J. Z. Kolter, and V. Koltun. "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling". In: *arXiv preprint arXiv:1803.01271* (Mar. 2018). arXiv: 1803.01271.

[4] Z. Baili, I. Tazi, and Y. Alj. "StapBot: An autonomous surgical suturing robot using staples". In: *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*. Apr. 2014, pp. 485–489.

[5] F. Bovo, G. De Rossi, and F. Visentin. "Surgical robot simulation with BBZ console". In: *Journal of Viualized Surgery* 3.4 (2017).

[6] B. Calli and A. M. Dollar. "Vision-based model predictive control for within-hand precision manipulation with underactuated grippers". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 2839–2845.

[7] M. Capiluppi, L. Schreiter, P. Fiorini, J. Raczkowsky, and H. Woern. "Modeling and verification of a robotic surgical system using hybrid input/output automata". In: *2013 European Control Conference (ECC)*. IEEE. 2013, pp. 4238–4243.

[8] M. Cefalo, E. Magrini, and G. Oriolo. "Sensor-based Task-Constrained Motion Planning using Model Predictive Control". In: *Proc. of 12th IFAC Symposium on Robot Control (SYROCO)*. 2018.

[9] L. Cheng, J. Fong, and M. Tavakoli. "Semi-Autonomous Surgical Robot Control for Beating-Heart Surgery". In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. 2019, pp. 1774–1781.

[10] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (Apr. 1979), pp. 224–227.

[11] R. De Rosa, I. Gori, F. Cuzzolin, and N. Cesa-Bianchi. "Active Incremental Recognition of Human Activities in a Streaming Context". In: *Pattern Recognition Letters* 99 (2017), pp. 48–56. ISSN: 0167-8655.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[13] O. Dergachyova, X. Morandi, and P. Jannin. "Knowledge transfer for surgical activity prediction". In: *International journal of computer assisted radiology and surgery* 13 (2018), pp. 1409–1417.

[14] F. Despinoy, D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet, and P. Jannin. "Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training". In: *IEEE Transactions on Biomedical Engineering* 63.6 (2016), pp. 1280–1291. ISSN: 15582531.

[15] S. Dieterich and I. Gibbs. "The CyberKnife in Clinical Use: Current Roles, Future Expectations". In: *IMRT, IGRT, SBRT – Advances in the Treatment Planning and Delivery of Radiotherapy*. Ed. by J. Meyer. Karger AG, 2011, pp. 181–194.

[16] L. Ding and C. Xu. "TricorNet: A Hybrid Temporal Convolutional and Recurrent Network for Video Action Segmentation". In: (2017), pp. 1–10. arXiv: 1705.07818.

[17] S. R. Fanello, I. Gori, G. Metta, and F. Odone. "One-shot learning for real-time action recognition". In: *Iberian Conference on Pattern Recognition and Image Analysis*. Vol. 14. 1. Springer. JMLR. org, 2013, pp. 31–40.

[18] M. J. Fard, S. Ameri, R. B. Chinnam, and R. D. Ellis. "Soft Boundary Approach for Unsupervised Gesture Segmentation in Robotic-Assisted Surgery". In: *IEEE Robotics and Automation Letters* 2.1 (2017), pp. 171–178. ISSN: 2377-3766.

[19] A. Fathi, X. Ren, and J. M. Rehg. "Learning to recognize objects in egocentric activities". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2011. ISBN: 9781457703942.

[20] F. Ferraguti, N. Preda, A. O. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi. "An Energy Tank-Based Interactive Control Architecture for Autonomous and Teleoperated Robotic Surgery". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1073–1088.

[21] F. Ferraguti, C. Talignani Landi, L. Sabattini, M. Bonfé, C. Fantuzzi, and C. Secchi. "A variable admittance control strategy for stable physical human-robot interaction". In: *The International Journal of Robotics Research (SAGE)* (Jan. 1, 2019). published.

[22] F. Ferraguti, N. Preda, G. De Rossi, M. Bonfe, R. Muradore, P. Fiorini, and C. Secchi. "A two-layer approach for shared control in semi-autonomous robotic surgery". In: *2015 European Control Conference, ECC 2015*. 2015, pp. 747–752. ISBN: 9783952426937.

[23] F. Ficuciello, G. Tamburrini, A. Arezzo, L. Villani, and B. Siciliano. "Autonomy in surgical robots and its meaningful human control". In: *Paladyn, Journal of Behavioral Robotics* 10.1 (2019), pp. 30–43.

[24] M. Fleder. *ROS : Robot " Operating " System*. http://www.ros.org. 2012.

[25] G. Forestier, F. Lalys, L. Riffaud, B. Trelhu, and P. Jannin. "Classification of surgical processes using dynamic time warping". In: *Journal of biomedical informatics* 45.2 (2012), pp. 255–264.

[26] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, et al. "JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling". In: 2014.

[27] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, et al. "JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling". In: *MICCAI Workshop: M2CAI*. Vol. 3. 2014, p. 3.

[28] C. E. Garcia, D. M. Prett, and M. Morari. "Model predictive control: theory and practice—a survey". In: *Automatica* 25.3 (1989), pp. 335–348.

[29] B. Gibaud, C. Penet, and P. Jannin. "OntoSPM: a core ontology of surgical procedure models". In: *Computer-assisted medical interventions: scientific problems, tools and clinical applications, Surgetica, Chambery, France* (2014).

[30] S. Haykin. *Neural Networks and Learning Machines*. Vol. 3. 2008, p. 906. ISBN: 9780131471399. arXiv: `arXiv:1312.6199v4`.

[31] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[32] P. Kazanzides, Z. Chen, A. Deguet, G. Fischer, R. Taylor, and S. DiMaio. "An Open-Source Research Kit for the da Vinci Surgical System". In: *IEEE Intl. Conf. on Robotics and Auto. (ICRA)*. Hong Kong, China, June 1, 2014, pp. 6434–6439.

[33] B. Kehoe, G. Kahn, J. Mahler, J. Kim, A. Lee, A. Lee, K. Nakagawa, S. Patil, W. Boyd, P. Abbeel, and K. Goldberg. "Autonomous multilateral debridement with the Raven surgical robot". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. May 2014, pp. 1432–1439.

[34] M. Khadem, C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli. "Ultrasound-Guided Model Predictive Control of Needle Steering in Biological Tissue". In: *Journal of Medical Robotics Research* 01.01 (2016), p. 1640007.

[35] S. M. Khansari-Zadeh and A. Billard. "A Dynamical System Approach to Realtime Obstacle Avoidance". In: *Autonomous Robots* 32.4 (2012), pp. 433–454.

[36] D. Kinga and J. B. Adam. "A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*. 2015.

[37] Y. Kouskoulas, D. Renshaw, A. Platzer, and P. Kazanzides. "Certifying the Safe Design of a Virtual Fixture Control Algorithm for a Surgical Robot". In: *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*. HSCC '13. New York, NY, USA: ACM, 2013, pp. 263–272. ISBN: 978-1-4503-1567-8.

[38] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg. "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning". In: vol. 36. 13-14. SAGE Publications Sage UK: London, England, 2017, pp. 1595–1618.

[39] H. Kuehne, J. Gall, and T. Serre. "An end-to-end generative framework for video segmentation and recognition". In: *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*. Sept. 2016. ISBN: 9781509006410. arXiv: 1509.01947.

[40] F. Lalys and P. Jannin. "Surgical process modelling: A review". In: *International Journal of Computer Assisted Radiology and Surgery* 9.3 (2014), pp. 495–511. ISSN: 1861-6429.

[41] C. Lea, G. D. Hager, and R. Vidal. "An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks". In: *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015* (2015), pp. 1123–1129. ISSN: 1550-5790.

[42] C. Lea, R. Vidal, and G. D. Hager. "Learning convolutional action primitives for fine-grained action recognition". In: *Proceedings - IEEE International Conference on Robotics and Automation*. Vol. 2016-June. 2016, pp. 1642–1649. ISBN: 9781467380263.

[43] C. Lea, R. Vidal, A. Reiter, and G. D. Hager. "Temporal convolutional networks: A unified approach to action segmentation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9915 LNCS (2016), pp. 47–54. ISSN: 16113349. arXiv: 1608.08242.

[44] P. Lei and S. Todorovic. "Temporal Deformable Residual Networks for Action Segmentation in Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6742–6751.

[45] Y. Li, J. Li, W. Lin, and J. Li. "Tiny-DSOD: Lightweight Object Detection for Resource-Restricted Usages". In: (July 2018). arXiv: 1807.11013.

[46] V. J. Lumelsky. "On fast computation of distance between line segments". In: *Information Processing Letters* 21 (1985), pp. 55–61.

[47] E. Mavroudi, D. Bhaskara, S. Sefati, H. Ali, and R. Vidal. "End-to-End Fine-Grained Action Segmentation and Recognition Using Conditional Random Field Models and Discriminative Sparse Coding". In: (Jan. 2018). arXiv: 1801.09571.

[48] F. Meng, L. D'Avolio, A. Chen, R. Taira, and H. Kangarloo. "Generating models of surgical procedures using UMLS concepts and multiple sequence alignment". In: *AMIA Annual Symposium Proceedings*. Vol. 2005. American Medical Informatics Association. 2005, p. 520.

[49] 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. P. Nguyen, N. Preda, G. Riolfo, A. Ristolainen, A. Sanna, C. Secchi, M. Torsello, and A. E. Yantac. "Development of a cognitive robotic system for simple surgical tasks". In: *International Journal of Advanced Robotic Systems* 12 (2015), p. 37. ISSN: 17298814.

[50] A. Murali, A. Garg, S. Krishnan, F. T. Pokorny, P. Abbeel, T. Darrell, and K. Goldberg. "TSC-DL: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with Deep Learning". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2016-June (2016), pp. 4150–4157. ISSN: 10504729.

[51] F. Nageotte, P. Zanne, C. Doignon, and M. deMathelin. "Stitching Planning in Laparoscopic Surgery: Towards Robot-assisted Suturing". In: *International Journal of Robotics Research* 28.10 (Oct. 2009), pp. 1303–1321.

[52] NASA. *SPICE toolkit*.

[53] N. A. Netravali, M. Börner, and W. L. Bargar. "The Use of ROBODOC in Total Hip and Knee Arthroplasty". In: *Computer-Assisted Musculoskeletal Surgery: Thinking and Executing in 3D*. Ed. by L. E. Ritacco, F. E. Milano, and E. Chao. Springer International Publishing, 2016, pp. 219–234.

[54] T. Neumuth, F. Loebe, and P. Jannin. "Similarity metrics for surgical process models". In: *Artificial intelligence in medicine* 54.1 (2012), pp. 15–27.

[55] E. Oleari, A. Leporini, D. Trojaniello, A. Sanna, U. Capitanio, F. Dehó, A. Larcher, F. Montorsi, A. Salonia, F. Setti, and R. Muradore. "Enhancing Surgical Process Modeling for Artificial Intelligence development in robotics: the SARAS case study for Minimally Invasive Procedures". In: *12th International Symposium on Medical Information and Communication Technology (ISMICT)*. IEEE. 2019, pp. 1–6.

[56] L. Pigou, A. A. A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre. "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video". In: *International Journal of Computer Vision* 126.2 (2016), pp. 1–10. ISSN: 15731405. arXiv: 1506.01911.

[57] N. Preda, F. Ferraguti, G. De Rossi, C. Secchi, R. Muradore, P. Fiorini, and M. Bonfè. "A Cognitive Robot Control Architecture for Autonomous Execution of Surgical Tasks". In: *Journal of Medical Robotics Research* 1.04 (Jan. 1, 2016). published.

[58] N. Preda, A. Manurung, O. Lambercy, R. Gassert, and M. Bonfè. "Motion planning for a multi-arm surgical robot using both sampling-based algorithms and motion primitives". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 1422–1427.

[59] J. Pujara, H. Miao, L. Getoor, and W. Cohen. "Knowledge graph identification". In: *International Semantic Web Conference*. Springer. 2013, pp. 542–557.

[60] J. Qi, Z. Jiang, G. Zhang, R. Miao, and Q. Su. "A surgical management information system driven by workflow". In: *2006 IEEE International Conference on Service Operations and Logistics, and Informatics*. IEEE. 2006, pp. 1014–1018.

[61] C. Reiley, E. Plaku, and G. Hager. "Motion generation of robotic surgical tasks: Learning from expert demonstrations". In: *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. Aug. 2010, pp. 967–970.

[62] C. E. Reiley and G. D. Hager. "Task versus subtask surgical skill evaluation of robotic minimally invasive surgery". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5761 LNCS.PART 1 (2009), pp. 435–442. ISSN: 03029743.

[63] A. Richard and J. Gall. "Temporal Action Detection Using a Statistical Language Model". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[64] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel. "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. Nov. 2013, pp. 4111–4117.

[65] F. Setti, E. Oleari, A. Leporini, D. Trojaniello, A. Sanna, U. Capitanio, F. Montorsi, A. Salonia, and R. Muradore. "A Multirobots Teleoperated Platform for Artificial Intelligence Training Data Collection in Minimally Invasive Surgery". In: *International Symposium on Medical Robotics (ISMR)*. IEEE. 2019, pp. 1–7.

[66] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S. F. Chang. "CDC: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Vol. 2017-Janua. 2017, pp. 1417–1426. ISBN: 9781538604571. arXiv: 1703.01515.

[67] S. Singh, C. Arora, and C. V. Jawahar. "First person action recognition using deep learned descriptors". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2620–2628.

[68] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. "Striving for Simplicity: The All Convolutional Net". In: *arXiv preprint arXiv:1412.6806* (2014), pp. 1–14. ISSN: 0254-8704. arXiv: 1412.6806.

[69] N. Srivastava, E. Mansimov, and R. Salakhutdinov. "Unsupervised Learning of Video Representations using LSTMs". In: (Feb. 2015). ISSN: 1938-7228. arXiv: 1502.04681.

[70] S. Stein and S. J. McKenna. "Combining embedded accelerometers with computer vision for recognizing food preparation activities". In: *Proceedings of the 2013 ACM*

international joint conference on Pervasive and ubiquitous computing - UbiComp '13. ACM. New York, New York, USA: ACM Press, 2013, p. 729. ISBN: 9781450317702.

[71] C. Talignani Landi, F. Ferraguti, L. Sabattini, C. Secchi, M. Bonfè, and C. Fantuzzi. "Variable Admittance Control Preventing Undesired Oscillating Behaviors in Physical Human-Robot Interaction". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, Canada, 2017.

[72] L. Tao, E. Elhamifar, S. Khudanpur, G. D. Hager, and R. Vidal. "Sparse Hidden Markov Models for Surgical Gesture Classification and Skill Evaluation". In: *Information Processing in Computer-Assisted Interventions* (2012), pp. 167–177.

[73] E. Tsironi, P. Barros, C. Weber, and S. Wermter. "An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition". In: *Neurocomputing* 268 (2017), pp. 76–86. ISSN: 18728286.

[74] 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". In: *IEEE Transactions on Medical Imaging* 36.1 (2017), pp. 86–97. ISSN: 1558254X. arXiv: 1602.03012.

[75] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. "WaveNet: A Generative Model for Raw Audio". In: (2016), pp. 1–15. ISSN: 0899-7667. arXiv: 1609.03499.

[76] T. R. K. Varma and P. Eldridge. "Use of the NeuroMate stereotactic robot in a frameless mode for functional neurosurgery". In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 2.2 (2006), pp. 107–113.

[77] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang. "Phoneme recognition using time-delay neural networks". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.3 (Mar. 1989), pp. 328–339. ISSN: 0096-3518.

[78] H. Wang, S. Wang, J. Ding, and H. Luo. "Suturing and tying knots assisted by a surgical robot system in laryngeal MIS". In: *Robotica* 28.Special Issue 02 (Mar. 2010), pp. 241–252.

[79] 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". In: *Science Robotics* 2.4 (2017), p. 8638.

[80] L. Yujian and L. Bo. "A Normalized Levenshtein Distance Metric". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1091–1095. ISSN: 0162-8828.

[81] L. Zappella, B. Béjar, G. Hager, and R. Vidal. "Surgical gesture classification from video and kinematic data". In: *Medical Image Analysis* 17.7 (2013), pp. 732–745. ISSN: 13618415.

A

APPENDIX

Author's Publications

A.1 N. Preda, F. Ferraguti, G. De Rossi, C. Secchi, R. Muradore, P. Fiorini, and M. Bonfè. "A Cognitive Robot Control Architecture for Autonomous Execution of Surgical Tasks". In: *Journal of Medical Robotics Research* 1.04 (Jan. 1, 2016). published

This work represents the culmination of the I-SUR European Project in which the authors developed a cognitive robotic task planning architecture to autonomously execute a series of simple surgical tasks, specifically a cryoablation needle insertion and a superficial wound suturing, based on Markovian temporal sequence formulations. The resulting architecture has been tested on a customized robotic platform developed by ETH-Zurich. The final tests proved the feasibility of the platform to execute the assigned tasks and revealed the potential of adopting probabilistic temporal constraints for cognitive architectures, paving the way for the following SARAS and ARS European Projects.

A.2 D. Dall'Alba, D. Zerbato, F. Bovo, G. De Rossi, and P. Fiorini. "Mixed training in surgical robotic combining simulation and test on real robot: preliminary results". In: *29th international conference of the international Society for Medical Innovation and Technology*. 2017

This study analyzes the quality of training achieved by novice trainee surgeons while adopting combinations of tasks executed on a real robotic minimally-invasive surgical platform (in this case the daVinci® Surgical System) and a simulated version developed by BBZ srl. Thanks to the use of the DaVinci Research Kit platform, a set of custom training platforms (that were replicated in the simulated environment), and the insights provided by the simulator, multiple indicators have been recorded on the subjects to conclude that, like training in other specialties such as airline pilot, the adoption of a simulated environment does improve dexterity, augment supervision capabilities over the task, and reduces the overall cognitive load on the subjects.

A.3 P. Fiorini, D. Dall'Alba, G. De Rossi, D. Naftalovich, and J. Burdick. "Mining robotic surgery data: training and modeling using the DVRK". In: *Hamlyn Symposium on surgical robotics*. London, U.K., 2017

This paper explores the potential offered by the data-driven approach to robotic surgery when the capability of recording low-level data from surgical actions being performed by both novice and expert surgeons can be accessed to model the next generation of surgical robots. The training data has been acquired by the aforementioned study and augmented by recordings of tasks performed by six expert urological and gynecological surgeons.

A.4 D. Naftalovich, D. Dall'Alba, G. De Rossi, P. Fiorini, and J. Burdick. "Foot pedal interface supplement for intra-operative camera control during microsurgery using the da Vinci Research Kit". In: *CRAS workshop*. Montpellier, France, 2017

This workshop contribution for the Conference on Robotic-Assisted Surgery (CRAS) saw the development of a complementary device for controlling the endoscope camera during microsurgery, a specific instance of minimally-invasive surgery that operates specialized sub-millimetric actuated devices to operate in highly constrained spaces such as the capillary system, the neuro system or even fetuses inside the mother's womb. In these conditions, the standard control of the camera through the master arm manipulators is insufficient and, thus, an additional footpedal interface has been successfully tested.

A.5 D. Naftalovich, D. Dall'Alba, G. De Rossi, P. Fiorini, and J. Burdick. "Robotic-assisted microsurgical anastomosis training with motion capture using the DVRK: The Caltech-Verona Dataset". In: *CARS 2017 Computer Assisted Radiology and Surgery*. Barcelona, Spain, 2017

A lack of data was noticed in the specific application of microsurgery, i.e. operations where the target is substantially smaller than the classic anatomical regions of interest for laparoscopy surgery and a specific application where robotic platforms are essential to the task. This work is intended to provide for the data requirement by assembling the freely-available Verona-Caltech microsurgery dataset with the required video and kinematic data to generate data-driven models.

A.6 F. Bovo, G. De Rossi, and F. Visentin. "Surgical robot simulation with BBZ console". In: *Journal of Viualized Surgery* 3.4 (2017)

This work is a demonstration of the digital twin capabilities provided by the BBZ simulated surgery console to robotic minimally-invasive surgery platforms. It demonstrates how the simulated environment is capable of reproducing faithfully not only the gestures (surgémes) that can be achieved in a commercial platform currently in use, but also the overall environment perception both anatomical and for training.

A.7 G. De Rossi and R. Muradore. "A bilateral teleoperation architecture using Smith predictor and adaptive network buffering". In: *IFAC 2017 World Congress*. 2017

A contribution to the 2017 IFAC World Congress, this paper focuses on overcoming the limitations of bilateral teleoperation systems (i.e. any teleoperation system that provides haptic feedback to the user) to operate within reasonable distances to avoid the insurgence of control instabilities dictated by the reduction in phase margin for the overall system. The solution provided included both buffering operations to maintain causality and predictive actions to reduce delays in haptic rendering to the user.

A.8 D. Naftalovich, D. Dall'Alba, G. De Rossi, P. Fiorini, Y. Fong, and J. Burdick. "Data Driven Analysis of Robotic Surgical Performance and Training". In: *American Physician-Scientist Association (APSA) Annual meeting 2017*. Chicago, U.S.A., 2017

This research paper encompasses and summarizes the data-driven research conducted on both standard laparoscopy and microsurgery composing the simulation and real robot training dataset and the Verona-Caltech dataset respectively.

A.9 A. Diodato, M. Brancadoro, G. De Rossi, H. Abidi, D. Dall'Alba, R. Muradore, G. Ciuti, P. Fiorini, A. Menciassi, and M. Cianchetti. "Soft Robotic Manipulator for Improving Dexterity in Minimally Invasive Surgery". In: *Surgical Innovation* (2018)

For this paper, the Scuola Superiore Sant'Anna in Pisa and the University of Verona cooperated in integrating the soft robot "StiffFlop" into the daVinci® surgical system as an improved fully-actuated endoscopic camera that allows the surgeon to maintain visual contact with the most intricate parts of the anatomy during laparoscopy. A miniaturized camera has been mounted at the end effector of the StiffFlop manipulator and a control strategy has been devised to map the infinite degrees-of-freedom of the latter on the master manipulator arms of the DaVinci that maintains the operator's movements coherent to the point-of-view. The tests performed by both trained and novice users confirmed that having access to increased camera movements was still sufficiently intuitive to the user while incrementing exponentially the exploratory capabilities in constrained environments.

A.10 G. De Rossi, N. Nicola Piccinelli, F. Cuzzolin, F. Setti, and R. Muradore. "Efficient Time-Interpolated Convolutional Network for Fine-Grained Action Segmentation". In: *Pattern Recognition* Submitted for review (2019)

In this paper submitted to *Pattern Recognition Letters*, the authors propose an encoder-decoder convolutional neural network for segmenting temporal features that replaces the standard "hourglass" structure with and encoder plus interpolator. This choice empirically proved to be beneficial when tested against fine-grained datasets such as the JIGSAWS as it both improved scores and reduced sensitivity to changes in hyperparameter (the latter being the result of the replacement of additional "deconvolution" operations dependent on kernel size). This work is supported by the related use of interpolation for image segmentation through neural networks.

A.11 G. De Rossi, M. Minelli, A. Sozzi, N. Piccinelli, F. Ferraguti, F. Setti, M. Bonfè, C. Secchi, and R. Muradore. "Cognitive Robotic Architecture for Semi-Autonomous Execution of Manipulation Tasks in a Surgical Environment". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2019)

This work laid the foundation for using gesture segmentation as the primary input for a cognitive control system for an assistant surgeon that integrates both reactive and supervision capabilities on the actions performed by the operator and the knowledge on the task available *a priori*. The system integrates seamlessly the predicted action being performed by the analysis of endoscope video data, the confidence of the prediction itself, the safety of execution enabled by a hybrid automata and a model predictive controller to provide collision-free velocity commands modulated by the prediction confidence. The entire system has been tested on a robotic minimally-invasive surgery pick-and-place training task in which an operator performed the "pick" act and the autonomous assistant completed the "place".

A.12 De Rossi, M. Minelli, S. Roin, F. Falezza, A. Sozzi, F. Ferraguti, F. Setti, M. Bonfè, C. Secchi, and R. Muradore. "A Multi-Modal Learning System for Action Segmentation and Control of Surgical Robots". In: *Robotics and Automation Magazine* Submitted for review (2019)

An improvement over the previous work, the setup is maintained but the neural network integrates multi-modal learning on videos and kinematic data to achieve greater precision in segmenting the pick and place task. The featurization of video is obtained via a motion history image enhanced ResNet while the temporal segmentation of the combined video and kinematic features is computed by a temporal convolutional network.

A.13 A. Sozzi, M. Bonfé, S. Farsoni, G. De Rossi, and R. Muradore. "Dynamic Motion Planning for Autonomous Assistive Surgical Robots". In: *MDPI Electronics* 8.9 (2019). ISSN: 2079-9292

This paper presents an approach to trajectory generation for laparoscopy tools based on dynamical systems that allows for obstacles to be modeled as moving solid geometric shapes instead of points in space. The adoption of a fast distance computation allows the entire system to evolve in real-time with the robot, thus allowing an exact and safe path planning solution. The approach was validated in both a simulated environment and on the SARAS assistant robotic platform and has been integrated in the controller for the autonomous assistant surgeon.

A.14 M. Minelli, A. Sozzi, G. De Rossi, F. Ferraguti, F. Setti, R. Muradore, M. Bonfè, and C. Secchi. "A motion planner integrating MPC and a dynamic waypoints generator for human-robot collaboration in a surgical scenario". In: *ICRA 2020*. Vol. Submitted for review. 2020

This paper submitted to ICRA 2020 presents a multi-robot model-predictive controller that optimizes the motions on the confidence level computed by the action segmentation soft-sensor and the obstacle avoidance path planner to control both arms of the SARAS assistant robot at the same time. The control strategy guarantees safety-of-motion by optimizing the entire predicted trajectory towards the desired goal using a constrained formulation. The optimization process has been improved in its computations to be performed at every control cycle of the robots in real-time. The validation on the SARAS platform proved the capability of the controller to provide both the desired minimum safety distance among the tools and the required velocity modulation dictated by the confidence level.