IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 October 2024, accepted 1 December 2024, date of publication 13 December 2024,
date of current version 23 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3516949

RESEARCH ARTICLE

Laparoscopic Suture Gestures Recognition via Machine Learning: A Method for Validation of Kinematic Features Selection

JUAN M. HERRERA-LÓPEZ, ÁLVARO GALÁN-CUENCA, ANTONIO J. REINA, ISABEL GARCÍA-MORALES, AND VÍCTOR F. MUÑOZ, (Member, IEEE)
Department of Systems Engineering and Automation, University of Málaga, 29071 Málaga, Spain

Corresponding author: Víctor F. Muñoz (vfmm@uma.es)

This work was supported in part by the Spanish Ministry of Science and Innovation under Grant PID2019-111023RB-C31 and Grant PID2022-138206OB-C31; and in part by the Department for Economic Transformation, Industry, Knowledge, and Universities of the Junta de Andalucía under Grant PY20-00738.

ABSTRACT In minimally invasive surgery, robotics integration has been crucial, with a current focus on developing collaborative algorithms to reduce surgeons' workload. Effective human-robot collaboration requires robots to perceive surgeons' gestures during interventions for appropriate assistance. Research in this task has utilized both image data, mainly using Deep Learning and Convolutional Neural Networks, and kinematic data extracted from the surgeons' instruments, processing kinematic sequences with Markov models, Recurrent Neural Networks and even unsupervised learning techniques. However, most studies that develop recognition models with kinematic data do not take into account any study of the significance that each kinematic variable plays in the recognition task, allowing for informed decisions at the time of training simpler models and choosing the sensor systems in deployment platforms. For that purpose, this work models the laparoscopic suturing manoeuvre as a set of simpler gestures to be recognized and, using the ReliefF algorithm on the JIGSAWS dataset's kinematic data, presents a study of significance of the different kinematic variables. To validate this study, three classification models based on the multilayer perceptron and on Hidden Markov Models have been trained using both the complete set of variables and a reduced selection including only the most significant. The results show that the aperture angle and orientation of the surgical tools retain enough information about the chosen gestures that the accuracy does not vary between equivalent models by more than 5.84% in any case.

INDEX TERMS Feature selection, hidden Markov models, laparoscopic suturing, neural networks, surgical gestures recognition, surgical robotics.

I. INTRODUCTION

Laparoscopic surgery is a type of minimally invasive surgery that involves interventions in the abdominal area. This type of surgery arises from the need to develop procedures that would minimise injuries caused by conventional open surgery. In laparoscopic surgery, interventions are performed by introducing special elongated tubular instruments and an endoscopic camera through small incisions in the patient's abdomen, previously insufflated with carbon dioxide in order to create a working space. This type of surgery has a number of advantages, including a reduction in mortality, length of hospital stay, and intervention costs [1]. However, it also has some disadvantages such as a steep learning curve, loss of 3D vision, or lack of ergonomics, which lead to increased surgeon fatigue, reducing surgeon performance [2].

To address the problems presented by laparoscopy, robotic systems have long been developed to perform minimally invasive procedures, in which typically the surgeon operates a series of robotic instruments by means of teleoperation. This increases the precision of the intervention, improving the accuracy, efficiency and results of surgical interventions, and allows surgeons to carry out laparoscopic procedures in a more precise and controlled manner, providing ergonomics

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

and reducing training times for doctors [3]. These type of robotic systems, however, are teleoperated systems, in which robots simply reproduce the movements that a surgeon performs through a controller. These kind of systems are generally not capable of performing tasks autonomously. Research on how to make robotic camera-handling systems more autonomous in laparoscopic surgery had already begun in 2014 [4], [5].

One of the main challenges of autonomous surgical robotics research is to reach level 4 autonomy, as proposed in [6]. These robotic platforms must be able to plan and execute a sequence of surgical manoeuvres autonomously. However, the need for the surgeon to supervise the entire process, and to have the capacity to intervene and take control of the robotic platform if necessary is also highlighted, which introduces a necessity for human-robot collaboration in autonomous surgical robotics. Some works such as [7] or [8] study the automation of the reasoning part of robotic platforms from the perspective of learning by demonstration and learning by reinforcement respectively, although they do not take into account aspects like the inclusion of systems for identifying the surgeon's gestures [9] or supervision systems for comparing the expected results with those obtained [10].

In the last years, there has been a growing interest in the development of perception algorithms capable of recognising the the gestures made by the surgeon during the intervention, in terms that are useful for human-robot collaborative systems. These algorithms use data extracted from the surgical environment mainly in the form of either video or kinematic data extracted from the surgeons' tools, generally used to build Machine Learning perception systems [11]. In the case of the models that use kinematic data, the choice of the variables to be used in the recognition models determines the amount of sensors that the platform needs to have, and the complexity of the algorithms that need to be implemented. Because of this, it is necessary to address the problem of kinematic features selection, verifying that the most relevant variables are selected, and that they contain enough information for the recognition algorithms to work.

This work proposes a method to validate a kinematic features selection algorithm that can be used to select the most relevant variables for the recognition of surgical gestures. The method consists in the training and testing of a set of classifiers, using the same dataset and different architecture configurations for each one of them, while varying the set of kinematic variables fed to the recognizers with each of the different classifier configurations. In this work, two of the classifiers are based on Hidden Markov Models (HMMs), as they allow for the modeling of the sequential dynamics of time series like the kinematic data extracted from the surgeons' tools. The other one is based on Multilayer Perceptron (MLP) neural networks, which are suitable for the classification of non-linearly inseparable problems like the recognition of surgical gestures, although they do not capture the sequentiality of the data. Also, in this work, the manoeuvre to be studied is the laparoscopic suturing manoeuvre, because of the clear differentiation of roles of each of the surgeon's tools during the procedure, which allows for the potential study and implementation of human-robot collaboration paradigms in future works. However, the work can be extended to other types of laparoscopic procedures that can be formulated as a sequence of simpler gestures.

The rest of the paper is structured as follows. Section II provides a description of previous works in the field of perception of surgical gestures. Section III describes the problem statement, and the state machine model of the suture manoeuvre used in this work. Section IV analyses the kinematic variables used for the recognition of the surgeon's gestures, and the significance of the different variables using a feature selection algorithm. Section V describes the HMM-based classifiers used for the recognition of the gestures and Section VI describes the MLP based classifier. Section VII presents the results obtained from the training of the models, using all the different classifier configurations, and Section VIII presents a discussion of the results, validating the significance of the kinematic variables. Finally, Section IX presents the conclusions of the work.

II. RELATED WORK

To implement human-robot collaboration paradigms, it is necessary for the robots to perceive the surgical field and the surgeon's gestures, so that it can perform the actions that best assist the surgeon. Multiple works study perception systems that analyse surgical images, mostly through Machine Learning techniques [12]. Some works like [13] use Deep Neural Network (DNN) architectures to identify the surgeon's tools in endoscopic images, providing a segmented version of the image from which the localization of the tool can be inferred. There are other works that can complement the latter, providing a 3D point cloud of the elements of the surgical field seen by the endoscope, using Visual SLAM techniques [14].

Some of the works of the last years center around the use of endoscopic images to recognize the surgical phase, like [15], in which uses a DNN together with an attention mechanism and the Online Hard Frame Mapper (OHFM) to extract the image features and recognize the phase. Other works like [9] use Convolutional Neural Networks together with Long Short-Term Memory (LSTM) blocks, a type of recurrent neural network, to classify surgical gestures. In spite of that, there are also works that do not solely rely on the use of images to recognise the phase, but also implement ontology-based models of the procedure and reasoners to infer that information with different levels of abstraction [16].

In addition to image based gesture classifiers, there is also a big group of works that study the recognition of the surgeons' gestures using the kinematic data extracted from the tools they use. In [17], the authors study the use of Hidden Markov Models (HMMs) together with reinforcement learning to recognize the gestures made by a surgeon that wears a

special sensorized glove during hand-assisted laparoscopy, in order to command certain actions of a surgical robot. The authors of [18] proposed a Multi-Scale Recurrent Neural Network (MS-RNN), based around the use of wavelet scattering operations and LSTMs, to recognize surgical gestures performed during various surgical procedures. Other works propose the use of unsupervised learning techniques to train their models, like in the case of [19], that develops an unsupervised trajectory segmentation method so that the segments can be classified as gestures with support vector machines (SVM) and k-nearest neighbors (KNN) algorithms. In [20], an unsupervised hierarchical Bayesian model called PRISM is proposed to identify common gestures across time series, also introducing a new metric called Temporal Structure Score (TSS) that evaluates the temporal structure of the recognized gestures.

The main drawback of the cited articles based on the use of kinematic data is the lack of a study of significance of the different kinematic variables used to differentiate the surgeons' gestures. Providing such a study would have helped reduce computational costs and the complexity of the data acquisition frameworks used by those works.

Table 1 summarizes the techniques used in the works discussed in this section that deal with the surgical gestures recognition task, and adds information about the datasets used in each work, and the performance that each model achieved. Most performance metrics are measured with the accuracy of the model, and vary in a range between 76.30% and 90.85%, although they are not directly comparable, as they were measured using different datasets. Reference [20] does not report an accuracy metric, but their own defined metric TSS of 46.88.

III. PROBLEM STATEMENT

The main task addressed in this work is the recognition of the state in which a laparoscopic suturing manoeuvre is found, based on the kinematic data of the suturing tools, that can later be used in the automation of certain actions. As most Machine Learning approaches require for classification tasks, a dataset is needed. In the case of this work, the "JHU-ISI Gesture and Skill Assessment Working Set" (JIGSAWS) dataset [21] was chosen. This set collects data on three laparoscopic surgery manoeuvres: suturing (the manoeuvre studied in this paper), needle passing, and knot tying, using two robotic arms with mechanised needle holders teleoperated by surgeons, from which the kinematic data was extracted. Although this work focuses on the suturing manoeuvre, the same methodology for validating a kinematic features selection developed in this work can be applied to the other two manoeuvres.

In this dataset, there is data from 8 different surgeons, named B to I, each of whom performed the manoeuvre 5 times. All the kinematic data from the needle holders controlled by the surgeons was captured at a frequency of 30 Hz. In addition, the kinematic samples in the dataset were labelled as one of 15 surgical gestures, which are the atomic actions that make up the manoeuvre. In the case of the suturing manoeuvre, only 10 of those gestures, from gesture 1 to 6 and from gesture 8 to 11, are taken into account, which are:

- G_1: *Reaching for needle with right hand.*
- G_2: *Positioning the tip of the needle.*
- G_3: *Pushing needle through the tissue.*
- G_4: *Transferring needle from left to right.*
- G_5: *Moving to center of workspace with needle in grip.*
- G_6: *Pulling suture with left hand.*
- G_8: *Orienting needle.*
- G_9: *Using right hand to help tighten suture.*
- G_{10}: *Loosening more suture.*
- G_{11}: *Dropping suture and moving to end points.*

As some of the samples in the JIGSAWS dataset at the beginning and at the end of some trials are not labelled, we have added a gesture G_0, *No gesture associated*, to the set of gestures, to account for those samples.

Having described the different surgical gestures proposed by JIGSAWS, [22] proposed a workflow model in which atomic gestures are composed in a way that they can be sequentially executed to perform the suturing manoeuvre. In this work, we chose to represent this model as a state machine, where the states correspond to these actions or gestures, and have associated with them the actions that the surgeon would perform during each gesture. In addition, they also have associated transitions between gestures, which model the direction of the workflow, and the order in which the gestures must be performed to achieve the suture, as well as the completion conditions of each phase. That way, the state machine model can be formalized as:

$$M = \{Q, A, T\}, \tag{1}$$

where Q is the set of states, $\{Q_i\}$, A is the set of actions $\{A_i\}$, and T is the set of transitions that, applied to a state, lead to another state from set Q:

$$T = \{t_i : Q_i \rightarrow Q_j\}. \tag{2}$$

The model proposed in [22] takes into account atomic actions that allow for a detailed decomposition of the manoeuvre in which the authors considered some things like fault recovery. However, a simpler model has been chosen in this work, with a more linear structure, which emphasises the main actions of the manoeuvre. This results in a state machine model for the suture manoeuvre based on 6 gestures, one for each of the states, that encompass JIGSAWS gestures. The simpler state machine can be seen in Fig. 1. In the figure, the states are represented by boxes containing a graphic representation of the elements of the surgical field during each state. The states are labelled as Q_i, and there is a list of the JIGSAWS gestures G_i encompassed in each state. The transitions between states are represented by one-way arrows, labelled t_i, and happen when the surgeon decides that the current state has been successfully executed. Also, points $P_{rest,L}$, $P_{rest,R}$, P_{in}, P_{out}, P_{pull} and $P_{rest,i}$ are represented by circles, and they show the points in the surgical

TABLE 1. Summarized description of the works related to gesture recognition.

Ref.	Authors	Year	Technique	Dataset	Performance
[9]	Huynhnguyen and Buy	2021	CNN and LSTM	JIGSAWS (Video)	Accuracy: 76.30%
[15]	Li et al.	2021	DNN with self attention and OHFM	M2CAI16-workflow	Accuracy: 85.80%
[16]	Neumann et al.	2022	Ontology based inference engine	Lumbar disc herniation removal dataset	Accuracy: 87.23%
[17]	López-Casado et al.	2019	HMM and Reinforcement learning	Custom glove dataset	Accuracy: 90.85%
[18]	Gurcan and Nguyen	2019	LSTM-based MS-RNN	JIGSAWS (Kinematics)	Accuracy: 90.20%
[19]	Despinoy et al.	2016	Trajectory segmentation, KNN and SVM	Custom kinematic trajectories dataset	Accuracy: 81.90%
[20]	Goel and Brunskill	2019	PRISM hierarchical Bayesian model	JIGSAWS (Kinematics)	TSS: 46.68

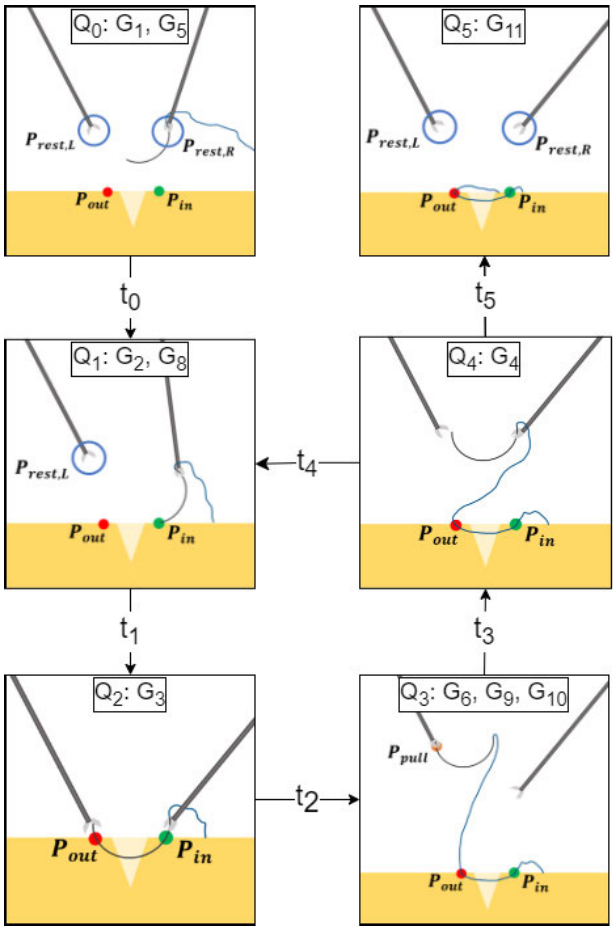


FIGURE 1. State machine model of the suture manoeuvre.

setup where the surgeon must place the needle holders to perform the manoeuvre. They are not fixed positions for the tools, as they are dynamically chosen by the surgeon and in some instances they represent a range of positions. The intermediate positions are not represented in the figure, as they are not relevant for the manoeuvre.

Also, in Fig. 1, it can be seen that in the procedure, the surgeon uses two laparoscopic needle holders, and follows a series of steps to perform the suture. The first state is a preliminary state in which both needle holders are prepared to start de manoeuvre. The next four states form a loop in which first, the right needle holder is used to perform a stitch, and the left needle holder is used to stretch the suturing thread and give the needle back to the right tool so that the surgeon can perform the next stitch loop iteration. The detailed description of the actions taken by the surgeon in each state of the state machine is as follows:

1) $\boldsymbol{Q_0}$: This is the initial state of the suture, in which the surgeon holds the needle with the right tool, and in which both tools are around resting points $P_{rest,i}$. In this state, it is considered that the surgeon is ready to start the manoeuvre. Formed by JIGSAWS gestures G_1 and G_5.
2) $\boldsymbol{Q_1}$: This is the first state of the state machine loop, where the objective is to move the needle closer to the stitch by moving the right needle holder, bringing it to the position from which it will pass through the tissue, P_{in}. Formed by JIGSAWS gestures G_2 and G_8.
3) $\boldsymbol{Q_2}$: In this state, the associated action consists of passing the needle through the tissue, so that the needle enters and exits the tissue through the appropriate positions and with the correct orientation. In addition, the left needle holder is brought close to the tip of the needle protruding through the tissue, in order to pick up the needle as it comes out through point P_{out}. Formed by JIGSAWS gesture G_3.
4) $\boldsymbol{Q_3}$: The purpose of this state is to stretch the suturing thread, by moving the needle away from the stitch point with the left tool, to point P_{pull}, to prepare the thread for a new iteration of the stitch loop. Formed by JIGSAWS gestures G_6, G_9 and G_{10}.
5) $\boldsymbol{Q_4}$: In this state the needle is transferred from the left holder to the right holder, preparing it for the next stitch. Formed by JIGSAWS gesture G_4.
6) $\boldsymbol{Q_5}$: This is the last state of the manoeuvre, in which the suture is dropped so that the manoeuvre can finish, and both tools return to points P_{rest}. Formed by JIGSAWS gesture G_{11}.

Each one of the states that compose the suturing manoeuvre are characterised by the values of the kinematic variables extracted from each needle holder controlled by the surgeon. In general those combination of kinematic variables are characteristic enough to be able to differentiate between the different states of the manoeuvre. However, not all the variables supply the same amount of information for the state recognition, thus appearing the necessity of performing a study of significance of the different variables.

IV. ANALYSIS OF KINEMATIC VARIABLES

This section provides a study of the kinematic variables that can be extracted from the surgeon's tools, and that can

TABLE 2. Summarized description of the kinematic variables that appear in this section.

Variable	Description
${}^{O}T_L$, ${}^{O}T_R$	Pose, with the cartesian position and orientation of the surgical left and right tool tips, respectively.
$\overrightarrow{P_L}$, $\overrightarrow{P_R}$	Position vector of the left and right surgical tools, respectively.
$\overrightarrow{\phi_L}$, $\overrightarrow{\phi_R}$	Orientation vector of the left and right surgical tools, respectively.
$\overrightarrow{Z_L}$, $\overrightarrow{Z_R}$	Unitary orientation vectors, extracted from the Z axis of the pose, of the left and right surgical tools, respectively.
$\overrightarrow{v_L}$, $\overrightarrow{v_R}$	Linear velocity vector of the left and right surgical tools, respectively.
v_L, v_R	Modulus of the linear velocity of the left and right surgical tools, respectively.
$\overrightarrow{\omega_L}$, $\overrightarrow{\omega_R}$	Angular velocity vector of the left and right surgical tools, respectively.
θ_L, θ_R	Opening angle of the left and right surgical tools, respectively.
D	Distance between the left and right tool tips.
α	Angle between the Z axes of the left and right tool tips.

be used to extract information about the surgical gestures being performed. For clarity, Table 2 introduces the different kinematic variables considered in this section.

In order to propose a set of variables to be used in the gesture recognition systems, it is necessary to define the original kinematic information from which the variables are extracted. Although in the case of a teleoperated surgical robotic system the kinematic information of the surgeon controllers could be used, in this work the kinematic information is extracted directly from the reference frames associated to the surgical tool tips. This is because the kinematic information of the surgeon controllers may not be always available, and the kinematic information of the tool tips is independent of the robotic platform used and therefore it is more reproducible.

For this reason, it was considered that each one of the surgical tools has a pose ${}^{O}T_i$ associated to it, where $\{O\}$ is the base reference system of the surgical setup, and $i = \{L, R\}$ (L and R stand for left and right respectively) is the tool. This transform encompasses both position and orientation information for a given time instant. This cartesian kinematic information for each tool is completed by adding its velocity and, finally, the tool's opening angle, which represents how open or closed the needle holder is, thus adding an articular kinematic variable.

That way, from each needle holder a set of kinematic variables V_i can be extracted, that is:

$$V_i = \{\overrightarrow{P_i}, \overrightarrow{\phi_i}, \overrightarrow{v_i}, \overrightarrow{\omega_i}, \theta_i\}, i \in \{L, R\}, \tag{3}$$

where $\overrightarrow{P_i}$ is the position vector; $\overrightarrow{\phi_i}$ is the orientation vector, in Euler X, Y, Z angles; $\overrightarrow{v_i}$ is the linear velocity vector; $\overrightarrow{\omega_i}$ is the angular velocity vector; and θ_i is the opening angle, all from tool i. All these variables capture, for each tool, its pose, velocity, and opening angle, that are considered sufficient to describe the state of the maneuver at a given time. So, the complete set of variables V that can be extracted from the surgical setup is:

$$V = V_L \cup V_R. \tag{4}$$

Most of the previous works use this complete set of kinematic variables V to train their machine learning models. However, in this work, the position variables $\overrightarrow{P_L}$ and $\overrightarrow{P_R}$ have not been considered, as they are not relative but absolute variables, meaning that they depend on the position of the base reference system used in the surgical setup. This dependency makes it difficult to train a machine learning model on a given data set and fails when deployed in a different surgical configuration. However, the remaining variables are relative, meaning that they only depend on the orientation of the base reference system, and not on its position, which is much more reproducible. In this way, a reduced set of variables $\overline{V}$ that excludes the position variables can be defined as:

$$\overline{V} = V - \{\overrightarrow{P_L}, \overrightarrow{P_R}\}. \tag{5}$$

Together with the previous variables, two additional relative kinematic variables have been considered, as in the work developed by [23]. These variables are D, the relative distance between the end effectors of each needle holder, and α, the angle between the two end effectors, both of them extractable from V. These variables are introduced, as they provide a kinematic description of the interaction between both tools as a distance and an angle. To obtain them, two different functions are applied, defined as:

$$D = f_P\left(\overrightarrow{P_L}, \overrightarrow{P_R}\right) = ||\overrightarrow{P_L} - \overrightarrow{P_R}|| \tag{6}$$

and

$$\alpha = f_\phi\left(\overrightarrow{Z_L}, \overrightarrow{Z_R}\right) = \arccos\left(\overrightarrow{Z_L} \cdot \overrightarrow{Z_R}\right), \tag{7}$$

where $\overrightarrow{Z_L}$ and $\overrightarrow{Z_R}$ are the unitary orientation vectors of the left and right needle holders, respectively, extracted from ${}^{O}T_L$ and ${}^{O}T_R$.

In this work, it was also considered to add the modulus of the linear velocity vector of each needle holder end effector, v_L and v_R, as they provide information about the dynamic state of the manoeuvre [23]. The combination of all kinematic variables from $\overline{V}$, defined as (5), with the interaction variables, defined as (6) and (7), and the velocity modulus variables, form the set of kinematic variables V' to be used in the recognition of the surgeon's gestures, that is:

$$V' = \overline{V} \cup \{D, \alpha, v_L, v_R\}. \tag{8}$$

In this way, the set of kinematic variables V' is composed of 24 components, between individual variables and vector coordinates, and they provide a description of the state of the manoeuvre for each time instant, in terms of the kinematic variables from set $\overline{V}$, the distance and angle between tools, D and α, and the modulus of the velocities v_L and v_R.

Eventually, the complete set of kinematic variables used for the recognition of the surgeon gestures, V', is:

$$V' = \{\overrightarrow{\phi_L}, \overrightarrow{\phi_R}, \overrightarrow{v_L}, \overrightarrow{v_R}, \overrightarrow{\omega_L}, \overrightarrow{\omega_R}, \theta_L, \theta_R, D, \alpha, v_I, v_D\}. \quad (9)$$

This set contains all the kinematic variables to be studied, to determine which of them are the most useful for recognition tasks, and what is their relationship with the surgical gestures defined in section III.

To determine this information, a feature selection analysis has been performed over the set of kinematic variables V'. This analysis aims to identify the most informative and discriminative variables from the relatively high-dimensional dataset of 24 kinematic components in V'. By reducing the complexity of algorithms that utilize this data and improving the performance and interpretability of models, this analysis provides more insight into the decision-making process of the models, allowing surgeons and medical professionals to validate and understand the factors influencing the recognition of surgical gestures. Also, a reduction of the dimensionality of the data set can speed up training processes, reduce data collection time and the computational cost of gesture recognition systems, and even reduce the number of physical sensors needed for the surgical setup.

In particular, the ReliefF algorithm [24] is used, based on a supervised learning process that allows to compute the importance of each variable of a dataset according to its ability to discriminate between the classes of that dataset. This algorithm assigns a higher score to the variables that give different values to samples of different classes, and a lower score to the variables that give different values to samples of the same class. ReliefF has been executed on the JIGSAWS dataset, from which the set of kinematic variables V' has been extracted, as it provides the kinematic data from the reference frames associated to each needle holder controlled by the surgeon, and the opening angle of the needle holders, which form the set of variables V, and from which set V' has been derived through (5) and (8).

Fig. 2 shows a bar chart resulting from running the ReliefF algorithm on the dataset, from which the variables that compose set V' were extracted. Each bar represents the score obtained by a variable according to the ReliefF algorithm, labelled on the vertical with its symbol, and the variables are ordered from highest to lowest score. The scale of the horizontal axis has been normalised so that the maximum score is 1.

When analysing the figure, it can be seen that ReliefF assigns a higher score to the needle holder opening angle variables, θ_L and θ_R. This can be understood intuitively, since in almost all proposed gestures one could assign by hand if each needle holder should be open or closed, giving the opening angle variables a very discriminative nature. In general, the next most important variables are the orientation variables, while those associated with the linear and angular velocities of the tools are the least discriminative.

In the following sections, a series of Machine Learning algorithms will be used to classify the kinematic samples

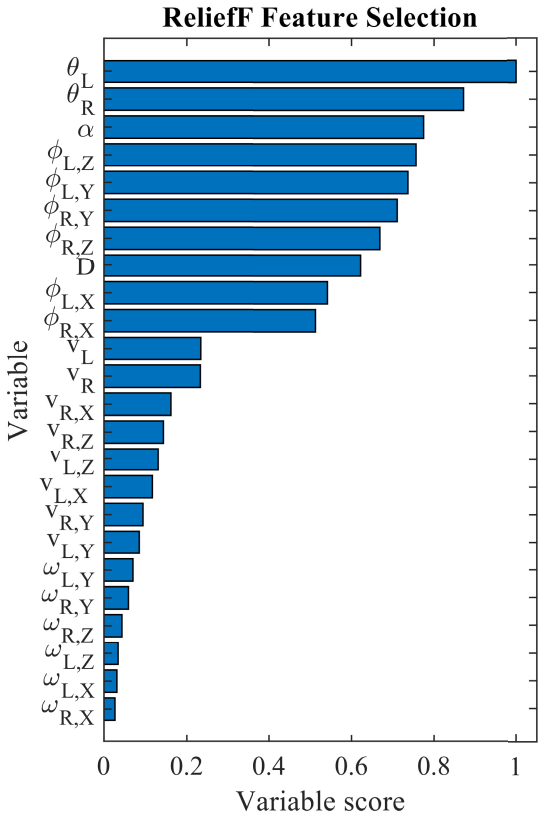


FIGURE 2. Results of the feature selection study using the ReliefF algorithm.

from JIGSAWS according to the surgical gestures they belong to. The training of all the models will be carried out using two different sets of kinematic variables. The first one will be the complete set of variables V', and the second one will be a reduced set, V'', which will only take into account the 10 most significant variables according to the ReliefF algorithm:

$$V'' = \{\overrightarrow{\phi_L}, \overrightarrow{\phi_R}, \theta_L, \theta_R, D, \alpha\}. \quad (10)$$

The 10 most significant variables have been chosen because, analysing the results of the ReliefF algorithm in Fig. 2, the sharpest drop in score can be observed between variables 10 and 11 ($\phi_{x,R}$ and v_L). In order to simplify the notation, the set of variables used for training will be denoted as $\widehat{V}$, and it will signify either V' or V''.

To signify each of the kinematic samples obtained from JIGSAWS that belong to the variables space $\widehat{V}$, the notation $\widehat{V}(k)$ is used, representing a sample taken at time $k \cdot \Delta t$, where Δt represents the sampling rate. Also, assuming a dataset where each kinematic sample is labeled to a gesture Q_j, $Q(k)$ represents the label for the sample taken at time $k \cdot \Delta t$. So, the complete data set of N kinematic samples and labels from JIGSAWS that belong to $\widehat{V}$ can be denoted as a set of tuples:

$$DS = \left\{\left(\widehat{V}(k), Q(k)\right)\right\}, \quad k = 1 \ldots N. \quad (11)$$

The training of all models with both V' and V'' sets of variables will allow to validate the results of the feature

selection analysis, and to determine if the reduction of the dimensionality of the data set has any impact on the performance of the models. The ideal result would be that the models trained with the reduced set of variables would have a similar performance to the models trained with the complete set of variables, since this would allow to reduce the complexity of the algorithms, and to improve the performance and interpretability of the models.

V. PROPOSED MACHINE LEARNING MODELS FOR GESTURE RECOGNITION BASED ON HIDDEN MARKOV MODELS

Three machine learning models have been considered to classify kinematic samples from a suturing manoeuvre. This section describes the first two models, based on the use of HMMs, which are a classical tool in the field of machine learning based on the modelling of processes using graphs, and therefore are especially suitable for the analysis of sequential data. In addition, these models are stochastic, which means that they incorporate randomness into the representation of events and transitions between states. Their main strength lies in their ability to model real processes that, despite having a more or less predefined sequence, may present variability, errors, or deviations from the expected sequence, such as processes performed by humans who may choose not to follow the expected sequence. The flexibility of these models allows them to capture complex patterns and non-linear behaviour in the data.

Since HMMs are based on the representation of processes using graphs, they can be used as a tool that provides useful information about the suturing process. Specifically, one could have a graphical model of the entire suture, representing the manoeuvre as a graph composed of gestures as states. One could also have a graph model for each of the gestures, being these graphs composed of atomic actions.

Hidden Markov Models can be defined as [25]:

$$\lambda = (S, E, A, B, \pi), \tag{12}$$

where S is the set of states, E is the set of emissions or observable characteristics, A is the transition matrix, B is the emission matrix, and π is the initial state distribution. According to this definition, HMMs have two main parameters that can be varied:

- The number of hidden states, which can be varied to capture more or less complex processes, as a lower number of states will force the models to be simpler.
- The number of possible emissions, which can be varied to capture more or less complex patterns in the data, as discretizing a set of continuous variables into a low number of qualitative emissions would result in a simpler interpretation of the original continuous variable space $\widehat{V}$.

Both variations of parameters would result in a variation of the sizes of matrices A and B, respectively.

HMM-based models used in this article work with discrete emissions, which means that the continuous kinematic variables from $\widehat{V}$ must be discretised. For this, although we could have chosen a more refined algorithm like DBSCAN, which can be used in problems involving complex elements like cross-modality [26], the k-Means clustering algorithm has been chosen, due to its simplicity and ease of use, which also presents good results in similar problems [27]. In this way, each kinematic sample $\widehat{V}(i)$ is assigned to a cluster, and therefore to a discrete emission. The centroids of the clusters are found by first assigning the number of clusters to be found, M_e, and then applying the k-Means algorithm to the complete dataset DS:

$$C = \{C_1, \ldots, C_{M_e}\} = kMeans(DS, M_e), \quad C_j \in \widehat{V}. \tag{13}$$

This way, set C is formed by M_e centroids, one for each cluster, each one belonging to the variables space $\widehat{V}$. Also, each centroid can be identified by a natural numeric label. With this, the set of discrete emissions E can be directly considered as the numeric labels of the centroids, so it is defined as $E = \{1, \ldots, M_e\}$. This way, each element of E, E_j, represents a discrete emission, and it is associated with a centroid C_j.

To obtain the emission that encodes a given kinematic sample $\widehat{V}(k)$, the following function is defined:

$$E(k) = f_{encoding}\left(\widehat{V}(k)\right), \quad E(k) \in E. \tag{14}$$

This function finds which of the centroids in C is closest to the kinematic sample $\widehat{V}(k)$, and assigns it the label of that centroid as the discrete emission $E(k)$, which is an element of set E.

The two HMM-based classifiers considered in this work use algorithms that need as input a sequence of emissions that encode the kinematic variables of the manoeuvre being executed that starts some time before the evaluation instant, and ends right at the evaluation instant. For this reason, a sequence with a window size of the last N_s kinematic samples is considered:

$$\left\{\widehat{V}(k - N_s + 1), \ldots, \widehat{V}(k)\right\}, \tag{15}$$

where k is the time instant at which the evaluation is performed. By applying the encoding function (14) to each of the kinematic samples in the window, a sequence of emissions belonging to set E is obtained for the same samples interval:

$$\{E(k - N_s + 1), \ldots, E(k)\}. \tag{16}$$

The HMM-based classifiers need a sequence of at least N_s samples to be able to classify a gesture, as they need information about the state of the manoeuvre in the previous time instants to work. As a consequence, the first N_s samples of the manoeuvre will not be classified, because there will not be enough samples to form a sequence of emissions.

A. HIDDEN MARKOV MODELS FOR EACH GESTURE

The first of the HMM-based classification strategies uses a procedure similar to [23]. In this process, each Hidden

Markov Model λ_i corresponds directly to a certain gesture Q_i. All HMMs are stored in a Gesture Library, set G:

$$G = \{\lambda_i\}, \quad i = 1 \ldots |Q|, \tag{17}$$

being $|Q|$ the number of gestures. As this classification model is based on HMMs, both the number of hidden states, identified as M_q and the number of possible emissions M_e can be varied as parameters that can directly affect the final performance of the implemented models.

As it is shown in Fig. 3, during the suturing manoeuvre, kinematic data is captured and encoded as a sequence of emissions $E(k)$, and it is fed to the Forward-Backward algorithm [28], [29], $FB(E(i), G)$. This algorithm is executed for each HMM in set G,computing the probability of observing that sequence of emissions for each of the HMMs. Finally, the HMM with the highest probability λ_i is selected, and the sequence of observable emissions is classified as the associated gesture Q.

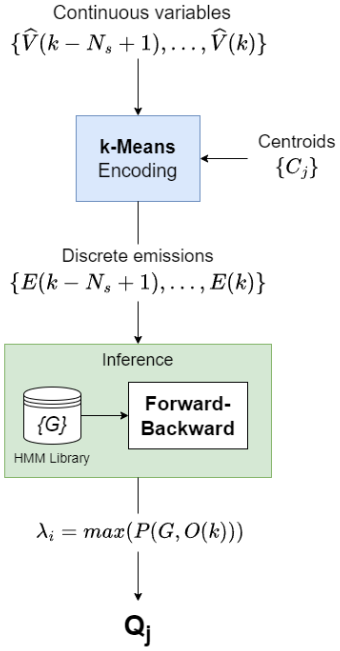


FIGURE 3. Inference process for the HMM-based gesture recognition model, with one HMM for each gesture.

Prior to the inference process described above, the gesture library must be constructed in the form of HMMs. For this, the same discretisation process of the training sequences is used, maintaining the same centroids for the k-means clusters and working with the same set of emissions E, and the Baum-Welch algorithm [30], [31], BW is executed, which automatically finds HMMs that model the input emission sequences:

$$\lambda_j = BW\left(E(i)_Q, |S|\right), \tag{18}$$

where $E(i)_Q$ is the sequence of emissions associated only with each gesture in set Q, and $|S|$ is the number of states of the HMM. This process is repeated for the training set of each of the gestures to be recognised.

To execute the algorithm described in this subsection, the models must be trained first with all the considered configurations. For this, it is assumed that there is a dataset with sequences of all the kinematic variables considered in set $\widehat{V}$ with the corresponding labels. To avoid the introduction of a bias in the dataset partition process, a k-fold cross-validation scheme will be applied, with $k = 10$. The training process will be the following:

1) The characteristics of the model are selected: number of hidden states, number of possible emissions, and set of kinematic variables, either V' or V''.
2) All the variables of the dataset are discretised with the centroids found by the k-means algorithm, obtaining sequences of observable emissions, and a downsampling of the dataset to 10 Hz is performed. This downsampling procedure consists of reducing the sampling frequency of the dataset from its original frequency, in case it is higher than 10 Hz.
3) The emissions are separated into sub-sequences, one for each gesture executed by the surgeon.
4) The sub-sequences of emissions are grouped into sets according to their labelled gesture, and each set is divided into 10 folds with the same proportion.
5) The k-fold training of the different models will be carried out. For each of the 10 folds, a Hidden Markov Model is trained for each gesture, using the associated training sub-sequences of emissions, an initialisation of the model, and the Baum-Welch algorithm, and leaving out the corresponding fold for testing, while training with the remaining folds.
6) The process is repeated for all the considered model configurations and sets of kinematic variables.

Once this process is finished for all the considered configurations, a set of trained models $G = \{\lambda_j\}$ for each different configuration will be obtained.

B. HIDDEN MARKOV MODEL FOR THE SUTURING MANOEUVRE

The second gesture recognition strategy using HMMs is based on the use of a single HMM, λ, that models the complete suturing manoeuvre. In this classifier, each hidden state of the HMM, S_i, corresponds directly to one of the gestures considered in section III, so $S_i = Q_i$. Unlike in the recognizer proposed in subsection V-A, each HMM hidden state does have a physical meaning directly related to the manoeuvre, and it has a direct correspondence to each of the gestures that are to be inferred. Therefore, the Markov model for the manoeuvre as a whole must have as many hidden states as gestures to be distinguished, $|Q|$:

$$S = \left\{S_1, \ldots, S_{M_q}\right\}, \quad M_q = |Q|, \tag{19}$$

where M_q is the number of hidden states of the HMM. For the emissions, the same procedure k-Means algorithm will be carried out to obtain set E and centroids C from the dataset. Also, the same sample encoding process will

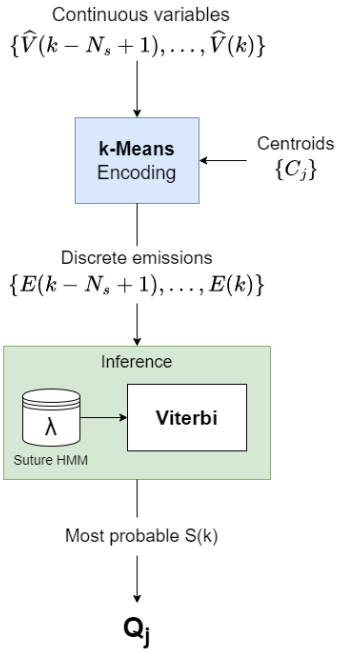


FIGURE 4. Inference process for the HMM-based gesture recognition model, with one HMM for the suturing manoeuvre.

be performed by following (14), obtaining the sequence of emissions belonging to E for any input sequence of kinematic samples.

The HMM used in this classification model maintains the number of possible emissions M_e as a variable parameter that impacts on the final performance of the model, the same as in the classifier from subsection V-A. However, as each state of the HMM corresponds to a surgical gesture, the number of hidden states is fixed to the number of gestures to be classified. This will also result on the matrices A and B from the HMM having a fixed size of $M_q \times M_q$ and $M_q \times M_e$, respectively.

The operation of the gesture recognizer can be seen in Fig. 4. As in the case of the recognizer using one HMM for each gesture, the input to the recognizer is a sequence of continuous variables belonging to set $\widehat{V}$, composed of the last N samples captured by the sensing platform for each of the kinematic variables, extracted from the surgeon's needle holders. Once inside the recognizer, the first step is the already discussed discretisation of the continuous kinematic variables into a discrete index, forming the sequence of discrete observable emissions belonging to E. Next, and using the HMM of the suturing manoeuvre, the Viterbi algorithm [32], V, is executed on the sequence of observable emissions, obtaining as a result the most probable HMM state for the last element of the sequence of emissions, $S(N)$, that directly relates to the classified gesture Q.

With this recognizer, the hidden states S_i of the HMM have been considered the same as the gestures of the suturing manoeuvre, so $S_i = Q_i$, and the emissions E correspond to the encoded kinematic variables. It is also assumed that there is a set of sequential data labelled by gestures, sample by sample, as noted in the dataset DS. Applying the encoding function (14) to the sequence of kinematic variables, and using the direct correspondence between gestures Q and hidden states S, the dataset would be transformed as:

$$\begin{aligned}\overline{DS} &= \left\{\left(f_{encoding}\left(\widehat{V}\left(k\right)\right), S\left(k\right)\right)\right\} \\ &= \left\{\left(E\left(k\right), S\left(k\right)\right)\right\}, \; S = Q, \; k = 1, \ldots, N. \end{aligned} \quad (20)$$

From this set, two sequences can be extracted, a sequence of emissions $E(k)$, formed by elements belonging to set E, and a sequence of hidden states $S(k)$, formed by elements belonging to set S, whose length is N and whose elements are related by the index k. Contrary to what happens to classifier from section V-A, no iterative training algorithm would be necessary to compose the HMM of the suturing manoeuvre, λ. Instead, it would simply be necessary to perform a statistical calculation of the number of transitions from one state to another, and the emissions associated with a particular state. This way, each element of the transition matrix A would be calculated as the following probability:

$$A_{M_q \times M_q} : A_{ij} = P\left(S_i(k) | S_j(k+1)\right), \quad k = 1, \ldots, N, \quad (21)$$

where A_{ij} is the element of S in row i and column j, and $P\left(S_i(k)|S_j(k+1)\right)$ is the probability of transitioning from state S_i at any given time instant k to state S_j at the following time instant $k+1$. This is computed for all the N elements of the dataset $\overline{DS}$, and for each of the M_q rows and columns of A. The elements of the emission matrix B would be calculated as the following probability:

$$A_{M_q \times M_e} : B_{ij} = P\left(E_j(k) | S_i(k)\right), , \quad k = 1, \ldots, N, \quad (22)$$

where B_{ij} is the element of B in row i and column j, and $P\left(E_j(k)|S_i(k)\right)$ is the probability of observing emission E_j at any given time instant k when the HMM is in state S_i, for all the N elements of the dataset $\overline{DS}$ and all the M_q rows and M_e columns of B.

Once the different configurations of the model have been defined, the description of the model building process can be described. This process is not an iterative training process such as the Baum-Welch algorithm, but simply a statistical calculation of the transitions between states and the emissions associated with each state, resulting in matrices A and B of the HMM. Assuming that there is a labelled dataset with sequences of all the kinematic variables considered in set $\widehat{V}$, the training process, with a k-fold cross-validation scheme with $k = 10$, will be carried out:

1) The characteristics of the model are selected: sampling frequency, number of possible emissions, and set of kinematic variables, either V' or V''.
2) The sequences of continuous kinematic variables are discretised using the centroids obtained by the k-means algorithm, obtaining sequences of emissions.
3) A downsampling of the emissions and gestures sequences is performed, to the selected sampling frequency.

4) Both the sequence of emissions and the sequence of gestures are composed into a single vector each, regardless of the organisation of the dataset finally used.
5) The vector is divided into 20 sub-vectors that maintain the sequentiality, and they are randomly permuted into a new vector which is then divided into 10 folds.
6) For each fold, the matrices A and B of the HMM of the suturing manoeuvre are calculated using the sequences of emissions and gestures, leaving the corresponding fold for testing, while being computed with the remaining folds.
7) The process is repeated for all the other configurations considered.

Once this process is finished, the A and B matrices of each HMM for the models based on the algorithm described in this section will be available, each one with a different configuration.

VI. PROPOSED MACHINE LEARNING MODELS FOR GESTURE RECOGNITION BASED ON THE MULTILAYER PERCEPTRON

Section V describes two different classifiers based on the use of HMMs. In addition to those, a third surgical gestures classifier has been considered in this section. This classifier is based on the use of a Multilayer Perceptron (MLP) neural network, due to its ability to learn non-linear relationships, which allows it to capture complex patterns in the data. Although this type of model does not capture the sequentiality in the data as well as models based on graphs like HMMs, and it does not allow such a clear interpretation as graph models, it has a great capacity for generalisation, a great ease of training and inference, and a great capacity to work with high dimensional data.

The MLP-based model takes as an input a vector of kinematic variables, and its output is the surgical gesture classification for that input. In this model, the input is not a sequence of kinematic samples from the variables space $\widehat{V}$ like in the case of the HMM based models, but just a single sample of kinematic variables $\widehat{V}(k)$ taken at a time instant k. This is because the MLP is not a sequential model, and it does not use the temporal context of an input sample to classify it. Inside the classifier, unlike the previous models, data is not encoded or discretized, and the kinematic variables are fed directly to the input layer of the MLP. Then, the data is propagated through the hidden layers of the MLP with the Rectified Linear Unit (ReLU) activation functions, and it is processed by the output layer. As the output layer implements a *Softmax* function, it returns a vector of probabilities of classification for each of the considered gestures. Finally, the gesture with the highest probability is selected as the inferred label Q_j for the input. The application of the MLP model can be formalized as a function f_{MLP}:

$$Q_j = f_{MLP}\left(\widehat{V}\,(k)\right), \quad Q_j \in Q, \quad \widehat{V}\,(k) \in \widehat{V}. \tag{23}$$

With this architecture, the two main parameters chosen to study their impact on the performance of the model are the number of hidden layers, M_{HL}, and the number of neurons per hidden layer, M_N. Usually, as either of these parameters increases, the capacity of the model to capture complex patterns in the data increases, but the model also becomes more prone to overfitting. In the case of the number on hidden layers, that increase is directly translated to its capacity of classifying non-linearly separable classes, or even classes that form disjoint sets in the variables space $\widehat{V}$, when the number of layers increases from 1 to 2 and 3 respectively.

In order to implement the MLP-based classifier, the neural network must be trained first, varying its architecture parameters for each considered configuration. Assuming that there is a dataset DS, with a sequence of all the kinematic variables considered in this article, $\widehat{V}(k)$, and a matching sequence of gesture labels for each of the kinematic samples, $Q(k)$, the k-fold training process with $k = 10$ can be described as follows:

1) The characteristics of the model's architecture are selected: architecture and set of kinematic variables, either variables set V' or V''.
2) The model architecture of the MLP is built with the design parameters previously chosen.
3) A sample matrix $\widehat{V}(k)$ and a label vector $Q(k)$ are built from the dataset DS. The sample matrix data does not need to be discretised, since the MLP model is capable of working with continuous variables.
4) Both the sample matrix and the labels vector are divided intro 10 folds.
5) For each of the 10 folds, the training process of the model is carried out leaving the corresponding fold for testing, while being computed with the remaining folds. While training, a validation partition is used to monitor the process and avoid overfitting of the model by means of an early stop if the process requires it. A random selection of 30% of the training data is used.
6) Also, for each of the 10 folds, the trained model is evaluated using the corresponding fold as the test set.
7) The process is repeated for all the other configurations considered.

This training process results in a series of MLP classifiers, one for each architecture configuration considered, that can be used to classify the gestures of the suturing manoeuvre.

VII. IMPLEMENTATION AND TRAINING RESULTS

With the description of the considered classifier models, the implementation and evaluation of those models can be carried out on the previously described JIGSAWS dataset, and with the considered architecture parameters for the models. As it was mentioned in previous sections, the dataset provides all the necessary kinematic data for all the recorded suturing experiments, expressed in the original variables space V, and a set of labels for each of the kinematic data samples, which correspond to the gestures executed by the surgeon. As the

JIGSAWS dataset provides kinematic variables belonging to the variables space V, all the samples are first converted to the variables space V', for the complete set of kinematic variables, and to the space V'', for the reduced set of variables according to the results of section IV. Then, all the models are built according to the procedures described in sections V and VI, taking into account all the different classifier configurations.

After all models have been built, the evaluation of the results is carried out using the corresponding partitions of the dataset, taking into account the k-fold cross-validation scheme with k = 10, and the performance of the models is measured in the form of a percentage of frame-by-frame accuracy averaged across all the folds. This accuracy is defined as the percentage of correct answers of the models on the test partition of the corresponding fold.

The building and validation of all configurations with either variables set V' or V'' is specially important as it will allow to verify the results of the feature selection process, and to determine the impact of the feature selection on the performance of the models. That is why an extensive study of the impact of the rest of the configuration parameters on the performance of the models was not carried out, as it was not the main focus of this work.

Also, all of the considered models have been built and evaluated using MATLAB 2022b with the Deep learning and the Statistics and Machine Learning Toolboxes, for the MLP and the HMM models respectively. The computer used was running Ubuntu 20.04 LTS, with an AMD Ryzen 5 5600X CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 3070 Ti GPU, which was used for the training of the MLP models.

1) HIDDEN MARKOV MODEL FOR EACH GESTURE

For the HMM-based recognizer with one HMM for each gesture, the considered models have been trained according to the methodology described in section V-A. The different model configurations considered in this classifier take into account the following parameters:

- **Number of hidden states, M_q.** It has been considered a number of 5, 10 and 30 hidden states. This captures a small number of states for a complex process like a surgical gesture, a medium number of states, and a sufficiently large number of states to capture the complexity of the process.
- **Number of possible emissions, M_e.** It has been considered a number of 30, 60 and 120 possible emissions. Taking into account the number of continuous variables, as much as 24, this captures a small number of clusters, a medium number of clusters, and a sufficiently large number of clusters to capture the complexity of the process.
- **Space of kinematic variables.** It has been considered the complete set of 24 kinematic variables, V', and the reduced set of 10 kinematic variables, V'', according to the criteria described in section IV.

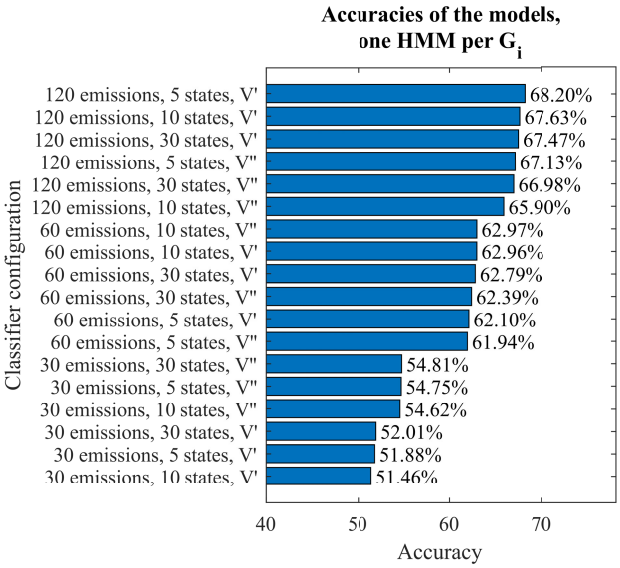


FIGURE 5. Comparative graph of performance of HMM-based models for each gesture, in the form of percentage of accuracy on the training set, for different numbers of hidden states (N_q), different numbers of emissions (M_e), and different sets of variables (V' or V''), ordered from best to worst.

Once all possible configurations of the classifier were built using the procedure from section V-A and the corresponding training partitions, its evaluation was carried out using the test partitions according to the k-fold scheme. For this evaluation, the performance was measured as the frame-by-frame accuracy of the models on the test dataset, using the classifier architecture described in Fig. 3, and performing the average over all 10 folds. With this architecture, the full test dataset sequence was processed by applying the classifier to the sequence with a moving window of length N_s of 10 samples.

Fig. 5 shows the results obtained for each of the configurations considered, ordered from best to worst according to their accuracy, from top to bottom of the figure. On the vertical axis, the characteristics of the models are labelled, and on the horizontal axis, the accuracy of the models is labelled.

Fig 5 shows that performance increases with the number of possible emissions, achieving accuracies of up to 68.20% for the model with 120 emissions, 5 hidden states, and the complete set of variables V'. On the lower side of the graph, the models with the lowest number of possible emissions have a poorer performance, with accuracies getting as low as 51.46% for the case of the model with 30 possible emissions, 10 hidden states, and the complete set of variables V'.

Although there seems to be a correlation between performance of the models and the number of possible emissions, that is not the case for the other two parameters. For the best performing models, the performance seems to increase as the number of hidden states decreases, but that trend is not maintained for the rest of the models. Also, there is no consistent boost in performance from the increase or reduction of the number of variables contained in the reduced

set V'', or the complete set V', although this will be further analyzed in the next section of this work.

2) HIDDEN MARKOV MODEL FOR THE SUTURING MANOEUVRE

In the case of the HMM-based recognizer with a single HMM for the suturing manoeuvre, and a hidden state for each gesture, the considered configurations have been trained according to the methodology described in section V-B. For those configurations, the following parameters have been considered:

- **Number of possible emissions, M_e.** It has been considered a number of 500, 2000 and 5000 possible emissions. This captures a small number of clusters, a medium number of clusters, and a sufficiently large number of clusters to capture the complexity of the process, in the same way as in the previous classifier. A higher number of possible emissions has been used, as they showed better performance for this classifier model.
- **Sampling frequency, f_s.** It has been considered a frequency of 0.5, 1 and 5 Hz. This captures a low sampling frequency, capturing a sample every 2 seconds, but but sufficiently high to be used with a high-level robotic task reasoner, and two higher sampling frequencies.
- **Space of kinematic variables.** It has been considered the complete set of 24 kinematic variables, V', and the reduced set of 10 kinematic variables, V'', according to the criterion described in section IV.

By following the procedure described in section V-B, all the different possible configurations of this classifier were built using the corresponding training folds, and their evaluation was carried out using the k-fold cross-validation scheme. As with the rest of classifiers, the performance of the models was measured in the form of a percentage of frame-by-frame accuracy averaged over all 10 folds, using the inference process described in Fig. 4, which was fed with windows of N_s of 20 samples extracted from the complete test dataset sequence.

As in the case of the previous classifier, Fig. 6 shows the results obtained for each of the configurations considered, ordered from best to worst according to frame-by-frame accuracy, from top to bottom of the figure. On the vertical axis of the figure, the characteristics of the models are labelled, and on the horizontal axis, the accuracy of the models is shown.

Fig. 6 shows that the performance of the models that use a HMM to represent the suturing manoeuvre is very low, starting at 20.66% accuracy for the model with 500 possible emissions processed at 1 Hz, with the complete set of variables V', and it only gets up to 27.71% accuracy for the model with 5000 possible emissions, processed at 1 Hz, and with the reduced set of variables V''.

The fact that the performance of this classifier is so low makes it not suitable for the surgical gesture recognizer, if it were to be used in a real scenario. In spite of the low

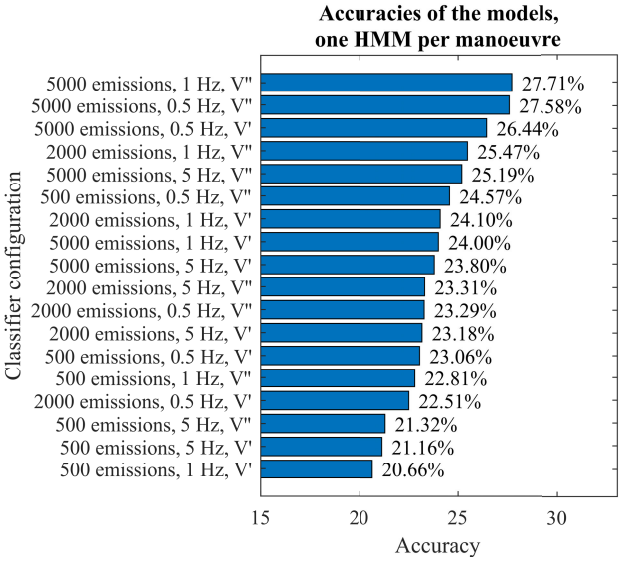


FIGURE 6. Comparative graph of performance of HMM-based models of the suturing manoeuvre, in the form of percentage of accuracy on the test set, for different sets of variables (V' or V''), different numbers of possible emissions (M_e), and different sampling frequencies (f_s), ordered from best to worst.

accuracy, there seems to be a trend in the performance, getting better with the increase of the number of possible emissions, although this fact could be indicative of an overfitting problem, and definitive conclusions cannot be drawn from such a poor performing classifier.

3) MULTILAYER PERCEPTRON

In the case of the last classifier, the MLP, the models considered have also been trained according to the methodology described its corresponding section VI, with all possible network architectures as defined by the following parameters:

- **Number of hidden layers, M_{HL}.** It has been considered a number of 1, 2 and 3 hidden layers. This captures the minimum number of hidden layers, capable of modelling linearly inseparable problems; a medium number of hidden layers being capable of modelling non-linearly inseparable problems; and a sufficiently large number of hidden layers to capture the complexity of the process, being capable of modelling problems with disjoint sets.
- **Number of neurons per layer, M_N.** It has been considered a number of 10, 30, 50 and 100 neurons per layer. This captures a small number of neurons per layer, for a problem of the complexity at hand, a couple of medium numbers of neurons per layer, and a sufficiently large number of neurons per layer to capture the complexity of the process.
- **Space of kinematic variables.** It has been considered the complete set of 24 kinematic variables, V', and the reduced set of 10 kinematic variables, V'', according to the criterion described in section IV.

Also, when training the models, a series of training and hyperparameters configurations were considered. The

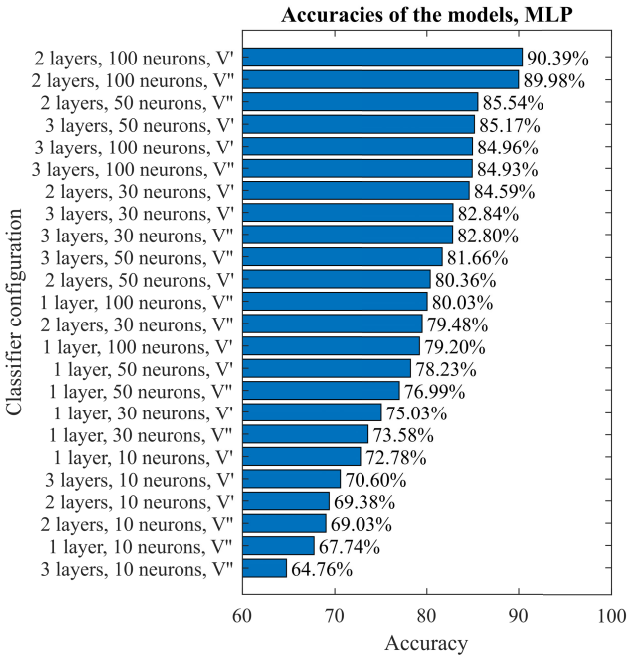


FIGURE 7. Comparative graph of performance of Multilayer Perceptron models, in the form of percentage of accuracy on the test set, for different network architectures (M_{HL} and M_N) and different sets of variables (V' or V''), ordered from best to worst.

number of training epochs was be limited to 2000, so at most, the training process would perform 2000 complete cycles through the training dataset. Also, to avoid overfitting, a validation data partition was used to perform an early stop of the training process if the validation error did not decrease in 6 consecutive epochs. In addition, the Scaled Conjugate Gradient algorithm was used to optimise the weights of the network. Other hyperparameters, such as the learning rate, were set to their default values, as they have been found to work well in most cases.

By following the training process from section VI over the dataset partitioned with the k-fold scheme, the set of all possible MLP possible trained models is obtained. Then, all the model configurations were evaluated, using the frame-by-frame accuracy as the performance measure, averaged over all 10 folds. In the case of this classifier, there is no need to use a moving window to process the test dataset, as the model is not sequential, so the test dataset was processed frame by frame.

As well as with the HMM-based classifiers, the results obtained for the MLP-based classifiers are shown in a comparative graph, in this case in Fig. 7, ordered from best to worst according to their accuracy, from top to bottom of the figure. On the vertical axis of the figure, the architecture of the network is labelled, and on the horizontal axis, the accuracy of the models is shown.

In Fig. 7 it can be seen that the performance of the MLP models is generally higher than that of the HMM-based models. The lowest accuracy percentage obtained is 64.76%, which corresponds to a model with a network architecture of 10 neurons and 3 hidden layers, and the reduced set of variables V'', while the highest is 90.39%, which corresponds to a model with a network architecture of 100 neurons and 2 hidden layers, and the complete set of variables V'. Looking at the rest of the models, a correlation between the complexity of the network architecture and the performance of the model can be deduced.

In addition, certain characteristics about the separability of the data can be inferred from the results obtained. In particular, from the existence of classifiers with a good performance with only two layers, it can be inferred that the classes are distributed in the space of variables by means of sets, at most, non-linear. If the classifiers with two hidden layers were not enough, it could be inferred that the classes form disjoint sets, but that is not the case for the dataset and variables space used.

Having verified the building process of the three classifiers, the next section will be dedicated to the study of the impact of the number of kinematic variables in the performance of the classifiers, so that the feature selection process can be validated.

VIII. DISCUSSION

In the previous section, the results of evaluating the three implemented classifier models for all their possible configurations have been studied, using the frame-by-frame accuracy as the performance measure, averaged over all 10 folds of the k-fold cross-validation. In this section, an analysis of the impact of the number of kinematic variables in the performance of each one of the classifiers is presented, so that the feature selection process described in section IV can be validated.

1) HIDDEN MARKOV MODEL FOR EACH GESTURE

The numerical results of the evaluation process for the classifier using a HMM for each gesture are shown in Table 3, in the form of the frame-by-frame accuracy. The first two columns capture the model architecture, as the number of hidden states M_q and the number of possible emissions M_e. The next two columns show the accuracy of the model for the described architecture using the complete set of 24 kinematic variables, $Acc_{V'}$, and the reduced set of 10 kinematic variables, $Acc_{V''}$. The last column shows the difference between the accuracy obtained with the complete set of variables and the accuracy obtained with the reduced set of variables for the same architecture, $Acc_{V''} - Acc_{V'}$.

Analysing the results presented in Table 3, it can be observed that the use of a reduced set of variables does not significantly affect performance. In some cases the use of a reduced data set increases performance, while in other cases it decreases it, but in no case is there a deviation of more than 3.2%. Also, in the case of the models with the lowest number of possible emissions, there is an increase in performance, although not very pronounced, while in the rest there is not an significant change. This result seems to indicate that, in fact,

TABLE 3. Comparative table of performance of HMM-based models for each gesture, in the form of percentage of accuracy on the training set, for different numbers of hidden states (M_q), different numbers of emissions (M_e), and different sets of variables.

M_q	M_e	$Acc_{V'}$	$Acc_{V''}$	$Acc_{V''} - Acc_{V'}$
30	30	52.01%	54.81%	2.80%
30	60	62.79%	62.39%	-0.40%
30	120	67.47%	66.98%	-0.49%
10	30	51.46%	54.62%	3.16%
10	60	62.96%	62.97%	0.01%
10	120	68.20%	67.13%	-1.07%
5	30	51.88%	54.75%	2.87%
5	60	62.10%	61.94%	-0.16%
5	120	68.20%	67.13%	-1.07%

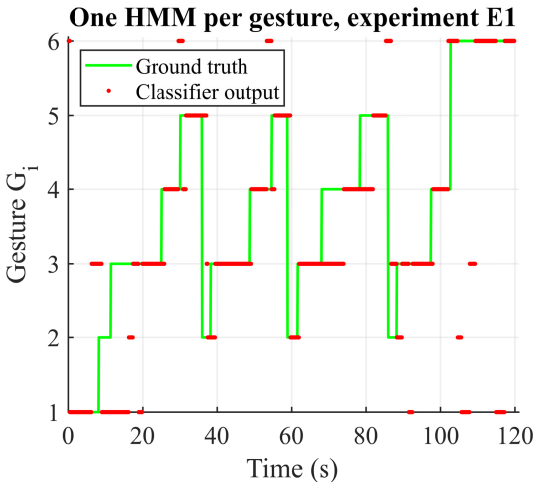


FIGURE 8. Labelling of the experiment E1 of the JIGSAWS dataset with the model based on HMM for each gesture, with mobile mode filter.

the variables that have been removed from the reduced set of variables to form the space V'' are not very relevant to the recognition of surgical gestures, validating the feature selection algorithm used.

As proof of the model's classification capability, Fig. 8 shows the results of classifying one of the experiments in the JIGSAWS dataset, specifically the first iteration performed by surgeon E, one of the models based on HMM for each gesture. In green, the label extracted directly from the dataset is represented, being the ''ground truth'', while the red dots represent the gesture labels inferred by the classifier system. This figure shows the performance of the model over a real trial, which is consistent with the measured results in Table 3.

2) HIDDEN MARKOV MODEL FOR THE SUTURING MANOEUVRE

As in the case of the previous classifier, Table 4 shows the numerical results obtained for the classifier based on an HMM for the suturing manoeuvre, in the form of the frame-by-frame accuracy. The first two columns capture the model configuration, as the number of possible emissions M_e and the sampling frequency f_s. The next two columns show the accuracy of the model for the described configuration using the complete set of 24 kinematic variables, $Acc_{V'}$, and

TABLE 4. Comparative table of performance of HMM-based models of the suturing manoeuvre, in the form of frame-by-frame accuracy on the test set, for different number of possible emissions (M_e), sampling frequencies f_s and sets of variables.

M_e	f_s	$Acc_{V'}$	$Acc_{V''}$	$Acc_{V''} - Acc_{V'}$
5000	0.5 Hz	26.44%	27.58%	1.14%
5000	1 Hz	24.00%	27.71%	3.71%
5000	5 Hz	23.80%	25.19%	1.39%
2000	0.5 Hz	22.51%	23.29%	0.78%
2000	1 Hz	24.10%	25.47%	1.37%
2000	5 Hz	23.18%	23.31%	0.13%
500	0.5 Hz	23.06%	24.57%	1.51%
500	1 Hz	20.66%	22.81%	2.15%
500	5 Hz	21.16%	21.32%	0.16%

the reduced set of 10 kinematic variables, $Acc_{V''}$. The last column shows the difference between the accuracy obtained with the complete set of variables and the accuracy obtained with the reduced set of variables for the same configuration, $Acc_{V''} - Acc_{V'}$.

From the results presented in the table it can be observed that the use of a reduced set of variables does not significantly affect performance. In all the cases tested in this work, the use of a reduced set of variables increases the accuracy of the model, although not in a significant way. The greatest increase in performance is observed in the case of the sampling frequency of 1 Hz, with 5000 possible emissions, with an increase of 3.71%. However, most of the remaining models present boosts in performance of 1.5% or lower.

Independently to that, none of the models presents a significant change in performance when using the reduced set of variables V'', being in all cases less than 3.71%, which seems to validate the feature selection algorithm used. In spite of that, definitive conclusions cannot be extracted just from these results, because of the overall very low accuracy of the models.

As with the previous classifier, Fig. 9 shows the results of using one of the classifiers based on a HMM for the suturing manoeuvre on one of the experiments in the JIGSAWS dataset, the first iteration performed by surgeon E, the same that was used for the previous classifier. In green, the label extracted directly from the dataset is represented, being the ''ground truth'', while the magenta dots represent the gesture labels inferred by the classifier system. In this figure, it can bee seen that this classifier is not suitable for the gestures recognition task.

3) MULTILAYER PERCEPTRON

For the last classifier model, Table 5 shows, for each configuration considered, the frame-by-frame accuracy depending on the number of hidden layers, the number of neurons per layer, and the set of variables used. The first two columns capture the model architecture, as the number of hidden layers M_{HL} and the number of neurons per layer M_N. The next three columns show the accuracy of the model for the described architecture using the complete set of 24 kinematic variables, $Acc_{V'}$, the reduced set of 10 kinematic variables, $Acc_{V''}$, and the difference between the accuracy obtained

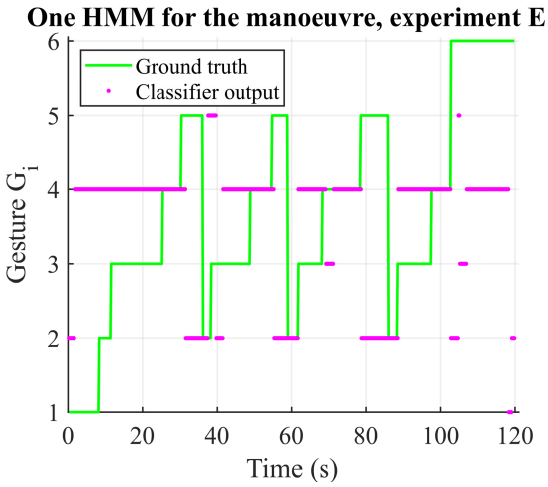


FIGURE 9. Labelling of the experiment E1 of the JIGSAWS dataset with the model based on HMM for the suturing manoeuvre, with mobile mode filter.

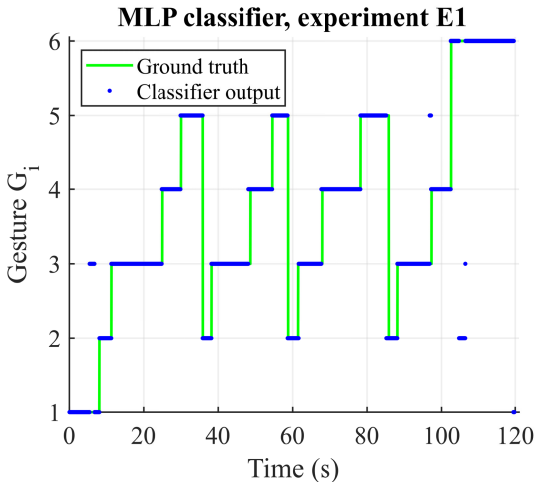


FIGURE 10. Labelling of the experiment E1 of the JIGSAWS dataset with the multilayer perceptron model, with mobile mode filter.

TABLE 5. Comparative table of performance of Multilayer Perceptron models, in the form of frame by frame accuracy on the test set, for different architectures and different sets of variables.

M_{HL}	M_N	$Acc_{V'}$	$Acc_{V''}$	$Acc_{V''} - Acc_{V'}$
3	100	84.96%	84.93%	-0.03%
2	100	90.39%	89.98%	-0.41%
1	100	79.20%	80.03%	0.83%
3	50	85.17%	81.66%	-3.51%
2	50	80.36%	85.54%	5.18%
1	50	78.23%	76.99%	-1.23%
3	30	82.84%	82.80%	-0.04%
2	30	84.59%	79.48%	-5.11%
1	30	75.03%	73.58%	-1.44%
3	10	70.60%	64.76%	-5.84%
2	10	69.38%	69.03%	-0.35%
1	10	72.78%	67.74%	-5.04%

with the complete set of variables and the accuracy obtained with the reduced set of variables for the same architecture, $Acc_{V''} - Acc_{V'}$.

As in the previous cases, the table shows that the use of a reduced set of variables does not significantly affect performance. In most configurations, the reduction of the number of variables decreases performance, but in no case it exceeds a reduction of 5.84%. This result, together with the results obtained in the previous classifiers, confirms that the feature selection algorithm has adequately ordered the set of kinematic variables according to their significance in discriminating the different gestures considered by the classifiers.

As in the case of the models based on HMMs, Fig. 10 is presented, which shows the labelling of one of the experiments in the JIGSAWS dataset, specifically the first iteration performed by surgeon E, as in Fig. 9, when an MLP classifier is used. In green, the label extracted directly from the dataset is represented, being the "ground truth", while the black dots represent the gesture labels inferred by the Multilayer Perceptron classifier system.

In this section, it was shown that the feature selection process was validated by the results obtained in the three classifiers. In all cases, the use of a reduced set of variables did not significantly affect performance, and in some cases, it even increased it. This result indicates that the variables that have been selected to form the space V'' are the most relevant to the recognition of surgical gestures.

IX. CONCLUSION

This paper examined kinematic variables that can be extracted from surgical instruments during a suturing manoeuvre, using the JIGSAWS dataset. A feature selection algorithm was used to determine the most important variables in gesture recognition, which were found to be the needle holders aperture angles, the angle between surgical tools, the orientation of the tools, the distance between tool tips, and the module of their linear velocities, unlike other variables like the components of the linear and angular velocities. Using the extracted variables, three classifiers were implemented and tested, two of them based on HMMs, and other based on MLP, using different architectures and both the complete and reduced sets of kinematic variables V' and V''. The results of evaluating all the configurations of the models validated the feature selection algorithm, as the use of a reduced set of variables did not significantly affect performance, and in some cases, even increased it. The methodology used can be reproduced for any other manoeuvre, and the results allow for the reduction of the number of variables to be used in the recognition of surgical gestures, which can be useful in the design of a suture assistance system, as they reduce the computational cost of the system, its complexity, the time required to capture kinematic samples and to train the classifiers, and even the number of physical sensors.

ACKNOWLEDGMENT

The authors would like to acknowledge the use of artificial intelligence (AI) in the preparation of this manuscript. Some

portions of the text were written with the assistance of GitHub Copilot, an AI-powered code completion tool developed by OpenAI and based on GPT-4, which was mainly used for basic editing and grammatical enhancements. They manually reviewed and edited all the generated content for accuracy, relevance, and coherence.

REFERENCES

[1] M. M. Tiwari, J. F. Reynoso, R. High, A. W. Tsang, and D. Oleynikov, "Safety, efficacy, and cost-effectiveness of common laparoscopic procedures," *Surgical Endoscopy*, vol. 25, no. 4, pp. 1127–1135, Apr. 2011.

[2] F. J. Pérez-Duarte, F. M. Sánchez-Margallo, I. D.-G. Martín-Portugués, M. Á. Sánchez-Hurtado, M. Lucas-Hernández, and J. U. Gargallo, "Ergonomía en cirugía laparoscópica y su importancia en la formación quirúrgica," *Cirugía Española*, vol. 90, no. 5, pp. 284–291, May 2012.

[3] H. R. H. Patel, A. Linares, and J. V. Joseph, "Robotic and laparoscopic surgery: Cost and training," *Surgical Oncol.*, vol. 18, no. 3, pp. 242–246, Sep. 2009.

[4] A. Pandya, L. Reisner, B. King, N. Lucas, A. Composto, M. Klein, and R. Ellis, "A review of camera viewpoint automation in robotic and laparoscopic surgery," *Robotics*, vol. 3, no. 3, pp. 310–329, Aug. 2014.

[5] M. C. Capolei, H. Wu, N. A. Andersen, and O. Ravn, "Positioning the laparoscopic camera with industrial robot arm," in *Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2017, pp. 138–143.

[6] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri, "Autonomy in surgical robotics," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 4, no. 1, pp. 651–679, May 2021.

[7] M. Deniša, K. L. Schwaner, I. Iturrate, and T. R. Savarimuthu, "Semi-autonomous cooperative tasks in a multi-arm robotic surgical domain," in *Proc. 20th Int. Conf. Adv. Robot. (ICAR)*, Dec. 2021, pp. 134–141.

[8] P. M. Scheikl, E. Tagliabue, B. Gyenes, M. Wagner, D. Dall'Alba, P. Fiorini, and F. Mathis-Ullrich, "Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 560–567, Feb. 2023.

[9] H. Huynhnguyen and U. A. Buy, "Toward gesture recognition in robot-assisted surgical procedures," in *Proc. 2nd Int. Conf. Societal Autom. (SA)*, Funchal, Portugal, May 2021, pp. 1–4.

[10] Y. Yan, H. Wang, H. Yu, F. Wang, J. Fang, J. Niu, and S. Guo, "Machine learning-based surgical state perception and collaborative control for a vascular interventional robot," *IEEE Sensors J.*, vol. 22, no. 7, pp. 7106–7118, Apr. 2022.

[11] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov, "Gesture recognition in robotic surgery: A review," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 6, pp. 2021–2035, Jun. 2021.

[12] H. Badgery, Y. Zhou, A. Siderellis, M. Read, and C. Davey, "Machine learning in laparoscopic surgery," in *Artificial Intelligence in Medicine*. Singapore: Springer, Jun. 2022, pp. 175–190.

[13] D. Papp, R. N. Elek, and T. Haidegger, "Surgical tool segmentation on the JIGSAWS dataset for autonomous image-based skill assessment," in *Proc. IEEE 10th Jubilee Int. Conf. Comput. Cybern. Cyber-Med. Syst. (ICCC)*, Jul. 2022, pp. 49–56.

[14] R. Docea, M. Pfeiffer, J. Müller, K. Krug, M. Hardner, P. Riedel, M. Menzel, F. R. Kolbinger, L. Frohneberg, J. Weitz, and S. Speidel, "A laparoscopic liver navigation pipeline with minimal setup requirements," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2022, pp. 578–582.

[15] Y. Li, Y. Li, W. He, W. Shi, T. Wang, and Y. Li, "SE-OHFM: A surgical phase recognition network with SE attention module," in *Proc. Int. Conf. Electron. Inf. Eng. Comput. Sci. (EIECS)*, Sep. 2021, pp. 608–611.

[16] J. Neumann, A. Uciteli, T. Meschke, R. Bieck, S. Franke, H. Herre, and T. Neumuth, "Ontology-based surgical workflow recognition and prediction," *J. Biomed. Informat.*, vol. 136, Dec. 2022, Art. no. 104240.

[17] C. López-Casado, E. Bauzano, I. Rivas-Blanco, C. J. Pérez-Del-Pulgar, and V. F. Muñoz, "A gesture recognition algorithm for hand-assisted laparoscopic surgery," *Sensors*, vol. 19, no. 23, p. 5182, Nov. 2019.

[18] I. Gurcan and H. V. Nguyen, "Surgical activities recognition using multi-scale recurrent networks," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2887–2891.

[19] 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," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 6, pp. 1280–1291, Jun. 2016.

[20] K. Goel and E. Brunskill, "Learning procedural abstractions and evaluating discrete latent temporal structure," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–11.

[21] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Bejar, D. D. Yuh, C. C. G. Chen, R. Vidal, S. Khudanpur, and G. D. Hager, "JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling," in *Proc. MICCAI Workshop*, 2014, p. 3.

[22] 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," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 9, pp. 2025–2041, Sep. 2017.

[23] M. B. E. Campos, "Diseño e implantación de un sistema multimodal para un asistente robótico," Ph.D. thesis, Dept. Syst. Eng. Automat., Universidad de Málaga, Málaga, Spain, 2013. [Online]. Available: http://purl.org/dc/dcmitype/Text

[24] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the Myopia of inductive learning algorithms with RELIEFF," *Appl. Intell.*, vol. 7, no. 1, pp. 39–55, Jan. 1997.

[25] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[26] Z. Pang, C. Wang, L. Zhao, Y. Liu, and G. Sharma, "Cross-modality hierarchical clustering and refinement for unsupervised visible-infrared person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2706–2718, Apr. 2024.

[27] Z. Pang, L. Zhao, Y. Liu, G. Sharma, and C. Wang, "Inter-modality similarity learning for unsupervised multi-modality person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 10, pp. 10411–10423, Oct. 2024.

[28] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, Dec. 1966.

[29] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bull. Amer. Math. Soc.*, vol. 73, no. 3, pp. 360–363, 1967.

[30] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, Feb. 1970.

[31] I. Miklós and I. M. Meyer, "A linear memory algorithm for Baum–Welch training," *BMC Bioinf.*, vol. 6, no. 1, p. 231, Sep. 2005.

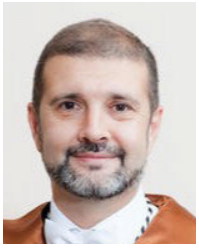
[32] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

JUAN M. HERRERA-LÓPEZ received the B.Eng. degree in electronics, robotics, and mechatronics engineering, and the M.Eng. degree in mechatronics engineering from the University of Málaga, in 2022 and 2023, respectively, where he is currently pursuing the Ph.D. degree in mechatronics engineering. He is developing his work with the Medical Robotics Laboratory. His research interests include intelligent robotics, machine learning, and human–robot collaboration in surgical robotics.

ÁLVARO GALÁN-CUENCA received the B.Eng. degree in electronics, robotics, and mechatronics engineering, and the master's degree in mechatronics engineering from the University of Málaga, in 2020 and 2021, respectively. He is currently pursuing the Ph.D. degree in mechatronics engineering. He is with the Medical Robotics Laboratory, University of Málaga. His research interests include robotics systems for surgical environments, human–robot collaboration, predicate logic in intelligent systems, and control engineering.

ANTONIO J. REINA received the degree in computer science and the Ph.D. degree in computer engineering from the University of Málaga (UMA), in 1991 and 2001, respectively. Since 1993, his research activity has been developed in the Department of Systems Engineering and Automation, where he began as a Research Fellow, in 1991. He has been an Associate Professor with the Department of Systems Engineering and Automation, since 2003. Throughout his career as a Researcher, he has actively and continuously participated in eighteen applied robotics research projects, five of which are in close cooperation with companies. His research production includes more than 35 publications, ten of which are articles from journals indexed in the ISI-JCR ranking, including a scientific contribution in *Nature* journal in the field of astronomy. He is also the co-author of three patents, one of them in exploitation.

ISABEL GARCÍA-MORALES received the M.S. and Ph.D. degrees in industrial electrical engineering from the University of Málaga, in 2000 and 2006, respectively. She is currently an Associate Professor, responsible for a variety of subjects related to robotics and control and coordinator of the electronics, robotics, and mechatronics engineering degree program with the University of Málaga. She has authored and co-authored more than 50 journal articles, conference papers, and book chapters, and has been involved in more than ten Spanish and European projects. Her research interests include process automation, control techniques, collaborative robotics, and surgical robotics.



VÍCTOR F. MUÑOZ (Member, IEEE) was born in Málaga, Spain, in 1966. He received the M.S. degree in computer science from the University of Málaga, in 1990, and the Ph.D. degree, in 1995. After holding a postgraduate fellowship, he joined the Systems Engineering and Automation Research Group, University of Málaga, where he began his research on mobile robot navigation problems. In 1997, he began research on medical robotics, which attained the highest point with the design and building of a robotic assistant for laparoscopic surgery and its use in human intervention, in 2004, and which has been published in conference papers, books, and magazines. Currently, he is a Full Professor of robotics with the Faculty of Electrical Engineers, University of Málaga.