Surgeon gesture recognition sensor system for a laparoscopic suture manoeuvre

Manuel Vargas-Puga
Dpto. Ing. de Sistemas y Automática
Universidad de Málaga
Málaga, Spain
0009-0000-5418-8684

Paula Luque-Contreras
Dpto. Ing. de Sistemas y Automática
Universidad de Málaga
Málaga, Spain
0009-0008-4024-0582

Juan M. Herrera-López
Dpto. Ing. de Sistemas y Automática
Universidad de Málaga
Málaga, Spain
0000-0003-3692-3088

Antonio J. Reina-Terol
Dpto. Ing. de Sistemas y Automática
Universidad de Málaga
Málaga, Spain
0000-0002-6064-3927

Víctor F. Muñoz
Dpto. Ing. de Sistemas y Automática
Universidad de Málaga
Málaga, Spain
0000-0003-2404-0050

***Abstract*—Laparoscopic surgery, a type of minimally invasive abdominal surgery, offers numerous benefits, including reduced mortality, shorter hospital stays, and lower costs. However, it also presents challenges such as a steep learning curve, loss of 3D vision and ergonomic issues. To address these challenges, robotic systems have been introduced to improve precision and control. This paper proposes a robust kinematic data acquisition system for capturing surgeon movements during laparoscopic procedures using two infrared trackers and a real-time Kalman filter, which reduces positioning errors and estimates tool poses when tracking is lost. Using this system, a dataset was created from the kinematic data of two laparoscopic instruments during a surgical suture manoeuvre. This dataset was used to train a gesture recognition system capable of identifying surgeon actions in the manoeuvre, contributing to the advancement of human-robot collaboration in laparoscopic surgery.**

***Index Terms*—Laparoscopic Surgery, Gesture Recognition, Suture Manoeuvre, Human-Robot Collaboration, Kalman Filter, Tracking System**

I. Introduction

In the last years, research in surgical robotics has been focused in developing human-robot collaboration systems, in which surgeons can work hand in hand with the robots on the same task, which is an important requirement to reach level 4 autonomy, as proposed in [1]. To achieve collaboration it is necessary for the robots to perceive the actions a surgeon is performing in terms that are useful for their reasoning systems, in order to act accordingly.

There are multiple works studying perception systems in surgical robotics, that analyse surgical images from laparoscopes in order to recognise surgical tools, anatomical structures, or the surgical phase [2] [3] [4] [5] [6]. However, other works do not use images, but kinematic data extracted either directly from surgical instruments, or from the end effectors of teleoperated surgical robots, and they suggest using different approaches such as Hidden Markov Models, [7], recurrent neural networks [8] and unsupervised learning techniques [9]. These systems can be combined with reasoners to achieve human-robot collaboration. Works as [10] or [11] 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 for human-robot collaboration, as none include systems for identifying the surgeon's gestures [12].

Many of the perception systems already mentioned use kinematic data for the recognition of gestures executed by the surgeons, although they do not delve into the acquisition systems used to capture that kinematic data. In this work, the development of a robust kinematic data acquisition platform is proposed to capture the movements executed by a surgeon using common laparoscopic instruments, based around the use of two commercial infrared trackers and the use of a real-time Kalman filter that provides the needed robustness, lowering the positioning error of the tracked instruments, and providing with their estimated poses whenever the tracking tools are lost. This system has been used to build a dataset that captures the movement of two laparoscopic instrument during a surgical manoeuvre, that can be used to train a surgical gestures recognition system, capable of telling a robotized platform the actions being executed by the surgeon during the manoeuvre, so that the robots can act accordingly.

The rest of the paper is structured as follows. In section II, the main objective of this paper is described, along with the contributions at both the low level, developing a Kalman filter, and the high level, creating a classifier. In section III, the data acquisition system for the kinematic data of the surgical tools used by a surgeon is described. In section IV, the modeling of the selected suturing manoeuvre is presented, followed by the explanation of the creation of a dataset and the subsequent training of a classification model. In section V, various experiments are conducted to verify the two systems

This work was supported by the Spanish Ministry of Science and Innovation under Grant PID2019-111023RB-C31 and Grant PID2022-138206OB-C31.

developed in the previous sections. Finally, in section VI, the conclusions drawn from this paper are presented.

II. Problem Statement

The main goal of this work is to develop a platform capable of recognizing the gestures performed by a surgeon during a laparoscopic manoeuvre that is being executed on a pelvi-trainer with regular laparoscopic instruments. Fig. 1 shows the proposed architecture to implement the gesture recognition system. The setup consists of two commercial optical infrared trackers, Polaris Spectra, which have a repeatability of 0.25 mm and a volumetric RMS accuracy of 0.30 mm. Each tracker operates within its reference system $\{0\}$ and $\{1\}$, one of them being the main tracker and the other being an auxiliary tracker. The optical trackers will act as sensors that read the pose of two tracking tools attached to the surgeon's instruments, and each of those tools will have a reference frame attached, $\{A_1\}$ and $\{B_1\}$. Reference frames $\{A_2\}$ and $\{B_2\}$ are defined at the tip of the surgical instruments, by shifting and rotating frames $\{A_1\}$ and $\{B_1\}$ respectively along the shaft of the rigid instruments, so that they are found on the tip of the instruments, with their Z axes pointing outwards. In addition to all these frames, a world frame $\{W\}$ will be used as a base reference frame, so that $\{A_2\}$ and $\{B_2\}$ can be referred to the same system independently of the tracker used to capture them. For frame $\{W\}$, an additional tracking tool will be used.

Once the kinematic data of the reference frames previously described is acquired by the trackers, it is then processed with a Kalman filter. The purpose of this filter is, mainly, to increase the robustness of the system by reducing the positioning error of frames $\{A_2\}$ and $\{B_2\}$ attached to the surgical instruments, and by correcting the loss of samples that can occur when the tools are not seen by any of the trackers.

Using the data acquisition system implemented, kinematic samples from a manoeuvre can be used in two ways. Firstly, the data can be used off-line to build a dataset in which surgeons perform the manoeuvre while the trackers record the poses of the surgical tools in real time. Then this dataset can be labelled by a human operator so that a gesture recognition algorithm can be built. Secondly, the data can be captured in real time and it can be fed to the already built gesture recognition algorithm, providing real-time semantic information about the gestures being performed by the surgeon. For this last application, it is important to implement the Kalman filtering algorithm in real time, so that the system can provide the most accurate information about the position of the surgical tools.

Once the system has been built and the gesture recognition algorithm has been trained, the semantic information supplied by the system could be used in a robotic surgical platform to automate certain parts of the manoeuvre, to provide support to the surgeon, to provide feedback to the surgeon about the quality of the manoeuvre being performed, or to assist in other tasks like endoscope handling.

The following sections will describe how the data acquisition system was built, focusing especially on the Kalman filter implementation, and how the system has been used to build a dataset that can be used to train gesture recognition algorithms.

III. Data Acquisition System

The first problem to be addressed is the development of the data acquisition system. As it was already introduced in section II, this system must be as robust as possible. In one hand, the use of two trackers is intended to provide redundancy to the system, so that if one of the trackers fails to see any of the tracking tools, the other tracker can still provide the information needed. On the other hand, A Kalman filter is designed to obtain an estimation of the instruments' poses.

A. Development of the mathematical models

To implement the Kalman filter, two mathematical models are created: one to estimate the system's evolution in position and another to estimate the orientation over time. Both of them use the state equation, $\hat{x}_{n+1,n} = A \cdot \hat{x}_{n,n} + B \cdot u_n$, where:

- $\hat{x}_{n+1,n}$, is the state vector (6×1) predicted for the next time step. In the position model, the state vector is composed of the position in the x, y, z directions and the linear velocity components v_x, v_y, v_z. On the other hand, the orientation model consists of the XYZ Euler angles,

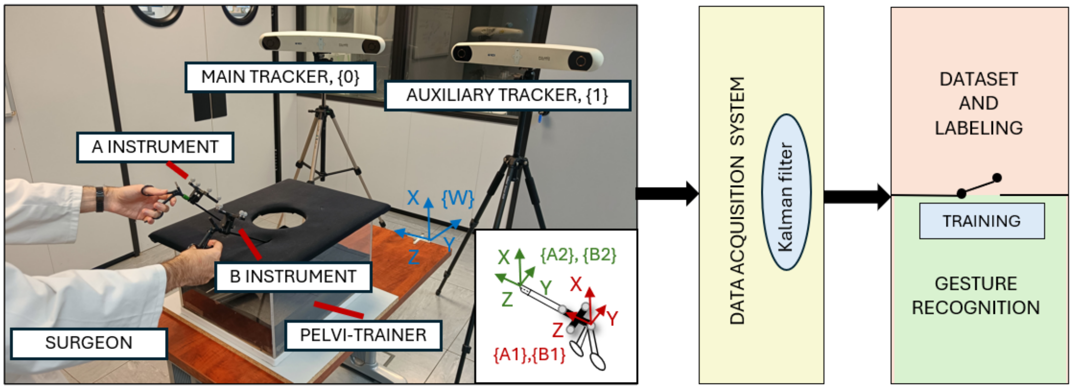


Fig. 1. Architecture of the gesture recognition system.

$roll, pitch, yaw$, and the angular velocities around these axes w_x, w_y, w_z.

- A, is the state transition matrix with dimensions (6×6) that relates the current state to the next state. This matrix will be the same for both the position model and the orientation model.
- $\hat{x}_{n,n}$, is the state vector (6×1) for the current instant.
- u_n, is the system input vector (3×1), composed of the linear accelerations for the position model and the angular accelerations for the orientation model.
- B, is the control matrix (6×3) that models how the input affects the system's state. This matrix will be the same for both the position model and the orientation model.

To obtain the matrices A and B, the movement is modelled as uniformly accelerated motion for the position estimation and, for the orientation model, the rotational kinematics equations are used analogously. For both models, there is a set of matrices A and B obtained by (1) and (2), that will be the same for position and orientation, and where Δ_t is the sampling time between measurements.

$$A = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$B = \begin{pmatrix} \Delta t^2/2 & 0 & 0 \\ 0 & \Delta t^2/2 & 0 \\ 0 & 0 & \Delta t^2/2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{pmatrix} \tag{2}$$

Once the state equation of the mathematical models is obtained, the system output equation $z_{n,n+1} = C \cdot \hat{x}_{n,n}$ is derived, where C represents the observation matrix (3) with dimensions (3×6) and is the same for both position and orientation models. This equation provides the position in x, y, z directions for the position model and the orientation in Euler angles XYZ for the orientation model.

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

B. Methodology of the Kalman filter

Obtained the mathematical models, the Kalman filter works as indicated in Fig. 2. First of all, the filter uses the current state estimate of position and orientation and applies the motion model previously described to predict the state at the next time step, updating the predicted pose based on the known dynamics of the system. The next estimate of the error covariance matrix, $P_{n+1,n}$ (6×6) is also predicted, representing the uncertainty in the predicted state by considering the process noise with the matrix Q (6×6), which is given by equation (4), where B is the control matrix defined in section III-A and σ_a is a scalar formed by the standard deviation of the input noise and is obtained experimentally.

$$Q = B \cdot \sigma_a^2 \cdot B' \tag{4}$$

Then, if the tracking tools are visible by any of the trackers, the correction step of the filter is carried out. The Kalman gain, K_n (6×3) is calculated to determine how much weight should be given to the new measurement versus the predicted state, where R_n is a diagonal matrix of order 3 composed of the errors of position and orientation in the x, y, z directions made by the sensors given by equation (5).

$$R = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix} \tag{5}$$

The state estimate is then updated, $x_{n+1,n}$ by combining the predicted state with the new measurement, z_n (3×1), weighted by the Kalman gain. Additionally, the error covariance $P_{n+1,n}$ is adjusted to reflect the reduced uncertainty after incorporating the new measurement.

As can be seen in Fig. 2, if the tool is visible by the main sensor, its measurements will be used for the correction phase. If the tool is not visible by the main sensor but is visible by the auxiliary sensor, the measurements from the auxiliary sensor will be used. In case the tool is not visible by any of the trackers, the correction is not performed, and the pose estimation obtained in the prediction step is maintained.

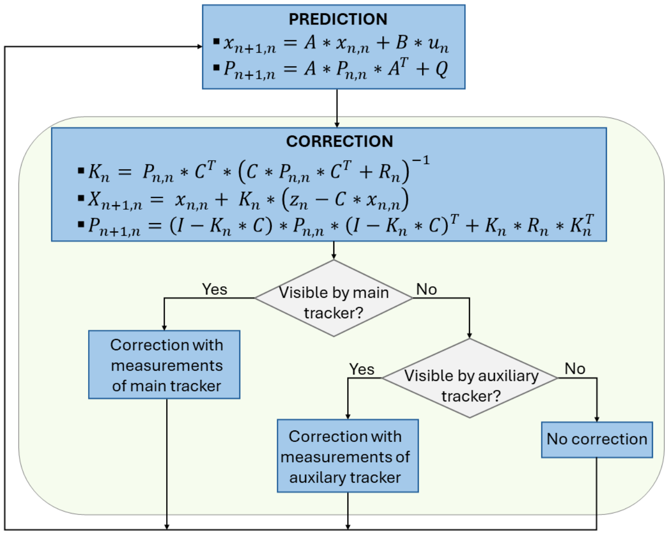

PREDICTION
▪ $x_{n+1,n} = A * x_{n,n} + B * u_n$
▪ $P_{n+1,n} = A * P_{n,n} * A^T + Q$
CORRECTION
▪ $K_n = P_{n,n} * C^T * \left(C * P_{n,n} * C^T + R_n\right)^{-1}$
▪ $X_{n+1,n} = x_{n,n} + K_n * \left(z_n - C * x_{n,n}\right)$
▪ $P_{n+1,n} = (I - K_n * C) * P_{n,n} * (I - K_n * C)^T + K_n * R_n * K_n^T$
Yes
Visible by main tracker?
No
Correction with measurements of main tracker
Yes
Visible by auxiliary tracker?
No
Correction with measurements of auxilary tracker
No correction

Fig. 2. Flow diagram for the Kalman filter algorithm.

The prediction and correction stages of the filter described earlier in this section are implemented in real-time. This means that each new measurement provided by the trackers is processed through the Kalman filter, calculating linear and angular velocities and accelerations using only the current and some previous samples. This results in a robust and precise system capable of immediately responding to dynamic changes in the tools where the poses processed with the Kalman filter are those of the systems $\{A_2\}$ and $\{B_2\}$ with respect to the system $\{W\}$.

IV. Gesture recogniser

As the data acquisition system has been already built, to develop a classification model for the gestures performed by the surgeon during a manoeuvre, it is necessary to carry out a study on the manoeuvre to be performed and model the various actions executed. Afterwards, a dataset must be created to subsequently train the model and make it functional.

A. Modelling of the suture manoeuvre

Firstly, to create a gesture recogniser, a suturing manoeuvre was selected to build a dataset that can be used to train a gesture recognition algorithm. Specifically, the Rosser suture, which is a common manoeuvre in laparoscopic surgery, was chosen. The Rosser suture has been divided into 8 different gestures that, when executed in the correct order, result in a successful suture.

Fig. 3 shows the graph that represents all the gestures made by the surgeon in the suture and how they are connected through transitions labelled as t_{ij}, via a finite state machine (FSM). These transitions indicate the order in which the gestures must be performed to complete the manoeuvre. Although the Rosser suture usually implies the cutting of the excess thread once the third knot has been performed, that gesture was not included in the manoeuvre, as it would imply the use of a third laparoscopic tool, making the data acquisition system more complex.

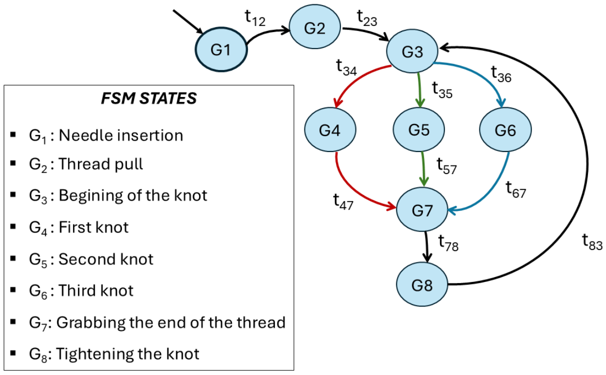


Fig. 3. Finite state machine of the gestures involved in the suture manoeuvre.

The first gesture, G1, involves passing the needle held by the B laparoscopic instrument through the tissue and is followed by gesture G2, which consists of pulling the thread with the A instrument, in preparation for the knots. Then, three knots have to be performed in the correct order. The first knot is performed by following G3, G4, G7, and G8. G3 corresponds to the beginning of the knot, bringing the needle close to the tissue, G4 involves making the first knot, G7 consists of grabbing the loose thread that protrudes through the original needle insertion point with the B instrument. Finally, in G8, the thread is pulled by both tools simultaneously, tightening and completing the knot. This results in completing a loop through the red path. The second knot is performed by following G3, G5, G7 and G8, represented by the green path. The third knot, following the blue path, is performed by G3, G6, G7 and G8. After that, the Rosser suture manoeuvre is completed. The difference between the three knots is that the first consists of two clockwise turns around B instrument, while the second consists of a single counterclockwise turn, and the third involves a single clockwise turn.

B. Dataset creation

The captured dataset consists of 9 different users performing the suture manoeuvre, 5 times each one, on a suture pad inside the pelvi-trainer. A video of each trial was recorded, and the kinematic data was acquired with the data acquisition system described in section III, and the poses of the surgical tools were recorded in real time and processed with the designed Kalman filter.

Once all trials were performed, the data was labelled manually by a single user watching the videos of the manoeuvre and labelling the gestures that were being performed at each moment based on their start and end points, as well as the type of movement. The labelling was done by using MATLAB's Video Labeler software, and users were named with letters A to I, so each trial can be identified by the index of the user followed by the index of the trial. This way, the fourth trial of user D was labelled as D04, for example.

When the labelling of the dataset was completed, the resulting files for each trial were **the MP4 video files** used in the labelling process; **a CSV file**, containing all kinematic data of of the surgical instruments during the suture; **a text file** with the corresponding label for each kinematic sample; and another text file, with the user's answers to **a questionnaire** to question about their age, experience in laparoscopic surgery, and others.

The most complex of these files is the kinematic data file, which contains the kinematic information of the tip of the two instruments being handled by the user. For each instrument, the file contains its position, its orientation (expressed in Euler XYZ angles), and its linear and angular velocities. The reference frames of which those variables were recorded are $\{A_2\}$ and $\{B_2\}$, as seen from the reference frame $\{W\}$. All these frames were described in section III. Also, all of the kinematic data corresponds to the output of the Kalman filter, described in the same section. Table I shows the variables captured in that file and their corresponding columns and units.

When this dataset was created, all the data was synchronised so that the each sample of the kinematic data corresponds to a frame of the video, which was captured at 30 Hz. This was done by synchronising the video capture with the output of the data acquisition system and interpolating the kinematic data to match the video frame rate. This way, as the labels were done by watching the video, the kinematic data can be labelled with the same information. These files can be found at the UMA-ROSSER-DATASET GitHub repository [1].

[1] https://github.com/manuvargas02/UMA-ROSSER-DATASET.git

TABLE I
DISTRIBUTION OF THE VARIABLES CAPTURED FOR EACH EXPERIMENT INSIDE THE CSV FILES.

Index	Label	Components	Units
1-3	LTTP_position_<x,y,z>	3	m
4-6	LTTP_orientation_<x,y,z>	3	rad
7-9	LTTP_velocity_linear_<x,y,z>	3	m/s
10-12	LTTP_velocity_angular_<x,y,z>	3	rad/s
13-15	RTTP_position_<x,y,z>	3	m
16-18	RTTP_orientation_<x,y,z>	3	rad
19-21	RTTP_velocity_linear_<x,y,z>	3	m/s
22-24	RTTP_velocity_angular_<x,y,z>	3	rad/s
25	timestamp	1	ms

C. Model training

Using the dataset created, a gesture recogniser is trained based on a Multilayer Perceptron to demonstrate the implementation of this dataset in laparoscopic suture manoeuvre classifiers. The first step is to decide which kinematic variables will be used as inputs to the classifier. In this case, 22 kinematic components described in Table II were chosen. Orientations and velocities of frames $\{A_2\}$ and $\{B_2\}$ were used, but positions were excluded due to their poor reproducibility from one experimental set-up to another. Also, the distance and the angle between the Z-axes of the instruments was included, calculated using the dot product of the Z vectors from their rotation matrices.

TABLE II
KINEMATIC VARIABLES CHOSEN TO TRAIN THE CLASSIFIER.

Name	No. Components	Units
Orientation of the end effector of each tool	6	rad
Components of linear velocity of end effector	6	m/s
Components of angular velocity of end effector	6	rad/s
Distance between tips of the tools	1	m
Angle between tips of the tools	1	rad
Magnitude of the linear velocity vector	2	m/s

The training is performed using MATLAB's *train* function. A network configuration with 3 hidden layers and 100 neurons per layer is created, capable of detecting nonlinearities between classes. Subsequently, the dataset is randomly partitioned into three sets to obtain training, validation, and test sets composed of 70%, 15%, and 15% of the data, respectively. The validation set is used to prevent overfitting of the network to the training data, which hinders generalisation to different types of samples. Early stopping is implemented if the validation set error increases for more than 6 consecutive epochs.

V. IMPLEMENTATION AND EXPERIMENTS

A. Experiments for the Kalman filter

The objective of this experiment is to verify the effectiveness of the implemented Kalman filter. To do this, an Universal Robot UR3 manipulator is used, with a tool attached at its end to be tracked by infrared sensors.

For the experiment, angular velocities are commanded to the robot, performing rotations around its x, y, or z axis, varying both the position and orientation of the attached tool. The measurements taken by the trackers are processed with the Kalman filter designed in section III, and using the measurements provided by the UR3 as a reference, the maximum errors in each experiment are calculated before and after the application of the filter. For this purpose, measurements from all devices are converted to a common reference system, synchronised, and interpolated at 60 Hz. Tool tip measurements are taken in a static manner using both the manipulator and the tracker to obtain the matrix that relates the dynamic measurements taken by the UR3 with the main tracker reference system.

The errors of the experiment are shown in Table III and IV for position and orientation respectively, where the first column represents the axis of the manipulator around which the rotation was performed, the second column represents the axis direction for which the maximum error is shown, the third and fourth columns show the error between the main and auxiliary trackers and the UR3, and the last column shows the error after applying the Kalman filter.

A reduction in errors was observed in all experiments conducted, achieving up to a **53%** reduction for the X Euler angle during rotation around the Z axis, with the error decreasing from 0.67º to 0.29º. Additionally, a reduction in the maximum position errors between the sensors and the robot by **69%** and **72%** was achieved, reducing the errors from 0.6858 mm and 0.7490 mm to 0.2102 mm and 0.2056 mm, respectively, demonstrating the effectiveness of the designed filter in the system.

TABLE III
MAXIMUM POSITION ERRORS BETWEEN THE SENSORS USED, THE KALMAN FILTER AND THE ROBOT, USED AS GROUND TRUTH.

Experiment	Axis	Error Main-UR3 (mm)	Error Aux-UR3 (mm)	Error Kalman-UR3 (mm)
Rotation around the X axis of the robot base	X	0.5455	0.5154	**0.0326**
	Y	0.6543	0.7490	**0.2056**
	Z	0.2985	0.3744	**0.2344**
Rotation around the Y axis of the robot base	X	0.2949	0.2431	**0.0315**
	Y	0.6858	**0.1956**	0.2102
	Z	**0.1209**	0.1587	0.1951
Rotation around the Z axis of the robot base	X	**0.1880**	0.1866	0.2743
	Y	0.1363	0.1917	**0.0681**
	Z	0.3307	0.3225	**0.0639**

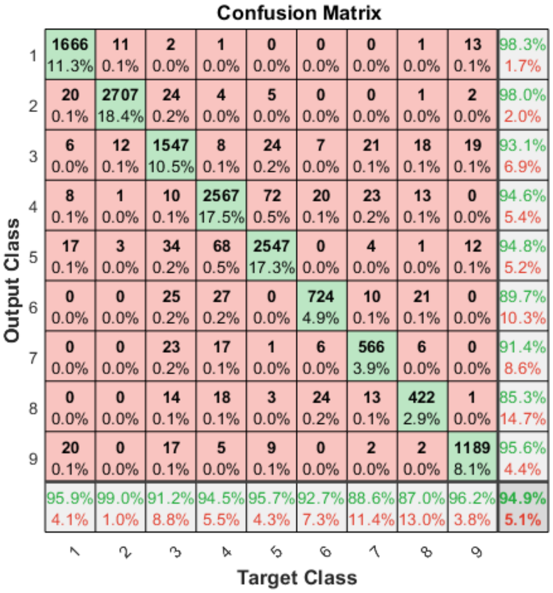
TABLE IV
MAXIMUM ORIENTATION ERRORS BETWEEN THE SENSORS USED, THE KALMAN FILTER AND THE ROBOT, USED AS GROUND TRUTH.

Experiment	Euler angle	Error Main-UR3 (º)	Error Aux-UR3 (º)	Error Kalman-UR3 (º)
Rotation around the X axis of the robot base	X	0.3018	0.5179	**0.1953**
	Y	0.2661	0.3151	**0.2291**
	Z	0.9549	0.9879	**0.9213**
Rotation around the Y axis of the robot base	X	0.7269	1.0331	**0.7213**
	Y	0.6442	0.5122	**0.4983**
	Z	0.9511	1.6683	**0.9170**
Rotation around the Z axis of the robot base	X	0.5161	0.6705	**0.2971**
	Y	**0.1547**	1.6056	1.0935
	Z	0.4138	0.4438	**0.3879**

B. Experiments for the dataset

The objective of this experiment is to verify the accuracy of the developed classification model. To achieve this, once the training process of the network is completed, the test set data is introduced into the model obtaining the confusion matrix shown in Fig. 4. This matrix contains in each cell the number of times the classifier identified a sample of class j as a sample of class i, with i being the row number and j being the column number. Therefore, the main diagonal represents the number of times the classifier correctly identified the samples of each class. On the other hand, the values outside the diagonal represent the number of samples that have been associated with an incorrect class. In the matrix, index 1 corresponds to gesture G1, 2 to G2, and so on. An additional gesture, G9, has been added to correspond to the samples before starting and after finishing the manoeuvre to avoid the need to trim the dataset.

A high accuracy rate of 94.9% is observed in the classifier's performance on the test data. There are no notable cases of confusion between classes. Analysing the accuracy percentages class by class, all are above 85%, demonstrating the effectiveness of the model for recognising the considered gestures.


Confusion Matrix
Output Class
Target Class

	1	2	3	4	5	6	7	8	9	
1	1666 11.3%	11 0.1%	2 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	13 0.1%	98.3% 1.7%
2	20 0.1%	2707 18.4%	24 0.2%	4 0.0%	5 0.0%	0 0.0%	0 0.0%	1 0.0%	2 0.0%	98.0% 2.0%
3	6 0.0%	12 0.1%	1547 10.5%	8 0.1%	24 0.2%	7 0.0%	21 0.1%	18 0.1%	19 0.1%	93.1% 6.9%
4	8 0.1%	1 0.0%	10 0.1%	2567 17.5%	72 0.5%	20 0.1%	23 0.2%	13 0.1%	0 0.0%	94.6% 5.4%
5	17 0.1%	3 0.0%	34 0.2%	68 0.5%	2547 17.3%	0 0.0%	4 0.0%	1 0.0%	12 0.1%	94.8% 5.2%
6	0 0.0%	0 0.0%	25 0.2%	27 0.2%	0 0.0%	724 4.9%	10 0.1%	21 0.1%	0 0.0%	89.7% 10.3%
7	0 0.0%	0 0.0%	23 0.2%	17 0.1%	1 0.0%	6 0.0%	566 3.9%	6 0.0%	0 0.0%	91.4% 8.6%
8	0 0.0%	0 0.0%	14 0.1%	18 0.1%	3 0.0%	24 0.2%	13 0.1%	422 2.9%	1 0.0%	85.3% 14.7%
9	20 0.1%	0 0.0%	17 0.1%	5 0.0%	9 0.1%	0 0.0%	2 0.0%	2 0.0%	1189 8.1%	95.6% 4.4%
	95.9% 4.1%	99.0% 1.0%	91.2% 8.8%	94.5% 5.5%	95.7% 4.3%	92.7% 7.3%	88.6% 11.4%	87.0% 13.0%	96.2% 3.8%	94.9% 5.1%

Fig. 4. Confusion matrix of the trained classifier evaluated on the test dataset.

VI. Conclusion

During the development of this project, a data acquisition system has been successfully developed by implementing a Kalman filter, which obtains the complete pose of a surgeon's surgical tools in real-time, achieving a reduction in errors of the trackers of up to 72% for position and 53% for orientation. Next, a laparoscopic suturing manoeuvre has been modelled, and a dataset has been created, which was subsequently used to train a gesture classification model for the surgeon. Using 22 kinematic variables and conducting a series of experiments with the test dataset, an overall accuracy of 94.9% was achieved. Additionally, when observing the class-by-class accuracies, all are above 85% verifying the effectiveness of the implemented systems for detecting actions in this type of manoeuvre.

For future work, some improvements could be made to address the limitations of this paper, such as designing an extended Kalman filter capable of handling the nonlinearities of the system, as well as modelling a state vector that considers both position and orientation instead of designing two independent filters. Additionally, if the tool is seen by both trackers, their respective measurements could be fused for greater accuracy applying the correction twice with the measurements from each of the trackers in each iteration.

References

[1] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri, "Autonomy in Surgical Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 651–679, 2021.

[2] H. Badgery, Y. Zhou, A. Siderellis, M. Read, and C. Davey, "Machine Learning in Laparoscopic Surgery," pp. 175–190, June 2022.

[3] 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 *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 578–582, Oct. 2022.

[4] D. Papp, R. N. Elek, and T. Haidegger, "Surgical Tool Segmentation on the JIGSAWS Dataset for Autonomous Image-based Skill Assessment," in *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*, pp. 000049–000056, July 2022.

[5] 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 *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pp. 608–611, Sept. 2021.

[6] J. Neumann, A. Uciteli, T. Meschke, R. Bieck, S. Franke, H. Herre, and T. Neumuth, "Ontology-based surgical workflow recognition and prediction," *Journal of Biomedical Informatics*, vol. 136, p. 104240, Dec. 2022.

[7] 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, p. 5182, Nov. 2019.

[8] I. Gurcan and H. V. Nguyen, "Surgical Activities Recognition Using Multi-scale Recurrent Networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2887–2891, May 2019.

[9] K. Goel and E. Brunskill, "Learning procedural abstractions and evaluating discrete latent temporal structure," 2019.

[10] M. Deniša, K. L. Schwaner, I. Iturrate, and T. R. Savarimuthu, "Semi-Autonomous Cooperative Tasks in a Multi-Arm Robotic Surgical Domain," in *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 134–141, Dec. 2021.

[11] 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 Robotics and Automation Letters*, vol. 8, pp. 560–567, Feb. 2023.

[12] H. Huynhnguyen and U. A. Buy, "Toward Gesture Recognition in Robot-Assisted Surgical Procedures," in *2020 2nd International Conference on Societal Automation (SA)*, (Funchal, Portugal), pp. 1–4, IEEE, May 2021.