**Design and Implementation of an Open-Source Based Software Framework for a Robotic Assistant in Laparoscopic Surgery**

Dayanne Fernandes da Cunha and Mariana C. Bernardes

Robotics and Automation Laboratory (LARA), University of Brasilia, Brazil  
dayannefernandesc@gmail.com, bernardes@unb.br

**Abstract.** This paper presents the software framework design for a robot assistant for laparoscopic surgery to be implemented at Brazilian's public health system (SUS). It’s a modular system, expandable and able to be applied in multiple environments since powerful open-source and cross-platforms frameworks/libraries as Qt, Julius and OpenCV were used. The application of a Brazilian Portuguese acoustic model for voice recognition to control the robot is also a remarkable achievement.

**Keywords:** Software Design, Surgical Robot, Laparoscopic Surgery, Speech Recognition

1. **Introduction**

The first laparoscopic surgeries were introduced in the beginning of 1980s, and since then, have been applied in a variety of surgical procedures such as appendectomy, cholecystectomy, reflux surgery, gastric surgery, urology [1]. These techniques are becoming the prefered approach over their open surgery counterparts due to a series of advantages such as less pain, smaller scars, and faster recovery time [2].

Traditional laparoscopic surgery requires the presence of a main surgeon that performs the surgical gestures, while an assistant surgeon is responsible for endoscope operation. Such arrangement requires trained coordination between both surgeons, as well as increases the chance of other problems such as fatigue of the assistant surgeon [3], inaccurate movements, conflicts on optimal surgical field visualization, and image tremors.

In this context, the use of robotic assistants in laparoscopy is becoming more common, with benefits that include tremor canceling, motion scaling, increased dexterity, comfort and endurance [4]. Using a robotic system controlled by the main surgeon to hold and move the endoscope allows less time to be wasted in communication between surgeons while the assistant surgeon may focus on another task or be free to carry out another procedure in a different operating room.

Some solutions have been proposed to implement such robotic endoscope holders, e.g., EndoAssist [5], Vicky [6]. Unfortunately, such products are expensive and must be imported in order to be used in Brazil. Also, there is no national technical assistance, which increases maintenance costs, and make the implementation of these systems prohibitive for Brazilian's public health system (SUS).

In light of this, the award-winning CLARA Project (honorable mention in Prêmio de Incentivo em Ciência, Tecnologia e Inovação para o SUS 2007, na categoria Produtos e Inovação em Saúde) is been developed at Universidade de Brasília’s Laboratório de Automação e Robótica (LARA) with funding from Brazilian Ministry of Health. The CLARA system proposes a low-cost robotic endoscope holder focused in abdominal laparoscopic surgeries for Brazilian SUS, with encouraging results so far, and the conclusion of its first functional prototype expected for 2018.

CLARA was designed to help the main surgeon perform bimanual laparoscopic procedures while keeping full control of endoscope motion through effective interfaces, voice recognition and a compact joystick attached to the laparoscopic grasper. Its system comprises not only the robotic mechanism and its embedded electronics, but also the software and interfaces that allow the algorithms and physical modules to interact, while offering the operator with a friendly Graphical User Interface (GUI) and tools for medical surgical workflow recordings, such as patient specific, staff and procedure information.

In order to implement CLARA system into SUS, topics like compatibility, modularity, fault-tolerance, security, usability, performance and maintainability were considered during its software modeling process. This paper focus on the development of a software architecture with all the aspects cited above, as part of a robotic endoscope holder system designed to assist abdominal laparoscopic surgery.

1. **Methods and Tools**

Medical robotics is entering new areas of expertise and reaching new levels of integration, which require more sophisticated software solutions. The newest generation of robots is expected to function not only as an improved extension of the physician eyes and hands, but also as a skillful and smart co-worker of their human counterpart [7].

Hence, the need of developing a software architecture that can integrate different modules with functionalities that expand the robot capabilities according to the performed procedure. Current trends include the integration of smart tools, image-based decision-support and surgical big data analytics and optimization [8].

Many surgical robots have been developed for research, but not much effort was devoted to the development of the software architecture. The few works that approached such question focused on the development of control software only [9], not including functionalities for a complete system with GUI, database and medical data recording tools. The concern of using architectures based on open-source code, which is a project requirement for CLARA, is even more limited with almost no prior works, to our knowledge [10].

**2.1 CLARA System Overview**

The CLARA System consists of a robotic device to automatically steer the endoscope while mechanically securing its movement to pivot around the abdominal entry point without moving the trocar. It also includes a series of independent modules that allow different forms of interface between the surgeon and the device. It is possible for the user to control the endoscope manually from a joystick interface, using predefined voice commands through a voice recognition interface, and automatically centering the camera image by using a surgical tool visual tracking module.

The use of a voice control interface has advantages over other interface options, such as a foot pedal, being more accurate and not requiring the surgeon to look away from the operative field [11]. The joystick functions as a secondary control, more suitable to longer distance navigation since it does not result in intermittent motions [12]. Additionally, the visual tracking module is particularly useful in laparoscopic surgery since it is important to keep the surgical tool centered in the endoscope image [13].

Besides the equipment itself, the CLARA System also comprises a software application that manages modules connections, data transmission and communication, and contains database support to medical information recording of the patient, surgical staff, planned and performed procedure data. Moreover, a GUI visually provides the physician all information of the modules current status and system’s configuration, allowing user-friendly interaction with the application.

**2.2 Software Modeling Process**

Before the software modeling process, the project requirements and scope where confirmed through visits at local public hospitals and interviews with laparoscopic surgeons, followed by an extensive research on open-source modules and codes that could be used to implement the desired system functionalities.

Qt [14] is an open-source, cross-platform framework with a thread library that helps control the asynchronous modules and a GUI component to produce its visualization. Hence, Qt is used as the main framework of CLARA System and also to build the GUI, while C++11 is its main programming language tool.

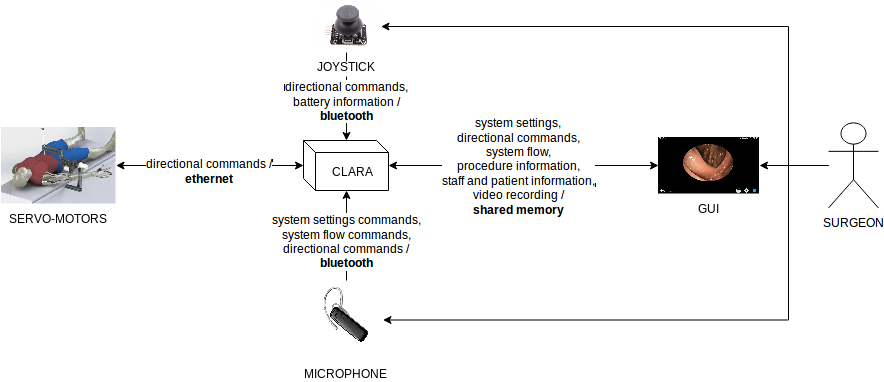
To implement CLARA’s voice control interface, an open-source speech recognition engine named Julius [15] was chosen in combination with a Brazilian Portuguese acoustic model [16]. This acoustic model presented consistent results for male users, but performed poorly for females. Hence, a secondary acoustic model dedicated to female voices was built and included so that CLARA is able to switch between both models according to user customized system configuration. The addition of other acoustic models (including other languages) is possible and straightforward.

OpenCV [17], which is a very standard and worldwide adopted image processing library free for academic and commercial purposes, was used to implement the module for surgical tool visual tracking. The developed algorithms were presented and discussed in a previous work [18].

The joystick interface uses Gattlib [19], which is an open-source library developed by Lab à Part that is used to access Generic Attribute Profile (GATT) protocol of BLE (Bluetooth Low Energy) devices.

Commands send from the BLE joystick communicate with CLARA servo-motors using ethernet sockets [20]. A socket is a generalized interprocess communication channel used to exchange messages between these processes running on different machines.

Figure 1 shows the basic communication flow from the interface modules and the robotic device.

  
 Figure 1: CLARA's interfaces to control the robot and their communication flow.

**2.2 CLARA’s Multi-thread Software Architecture**

### An overview of CLARA software is illustrated in Figure 2. The system’s architecture is structured with 8 modules, namely:

### **Core -** All modules access Core to transmit information for the logging. It helps to debug and detect any problem that may occur during the system execution. User Login, Settings and Vision submodules from Control access the database to get or set data about system configuration settings, video recording options, and staff, patient and other pertinent medical information.

### **Vision -** Camera submodule receives endoscope images and extracts tool borders and/or custom markers (this process is made by Border and Tag submodules) to obtain the current tool image position (Position submodule). This information is used by the Control module as feedback to a position controller for automatic image centering.

**Speech -** Julius submodule receives vocal commands from a microphone device and JuliusAPI recognizes the predetermined words sequence and their confidence scores. These information are passed to Control module that uses such data to extract commands to control the CLARA System.

**Joystick -** Bluno receives commands bluetooth command sent from the joystick device and the Main submodule parses such commands to transmit clean data to Control module. Joystick as a submodule in Control, gives a semantic to these commands and send them to Motor. The commands are superimposed in endoscopic images and shown in Video Page (UI module) as a feedback to the surgeon.

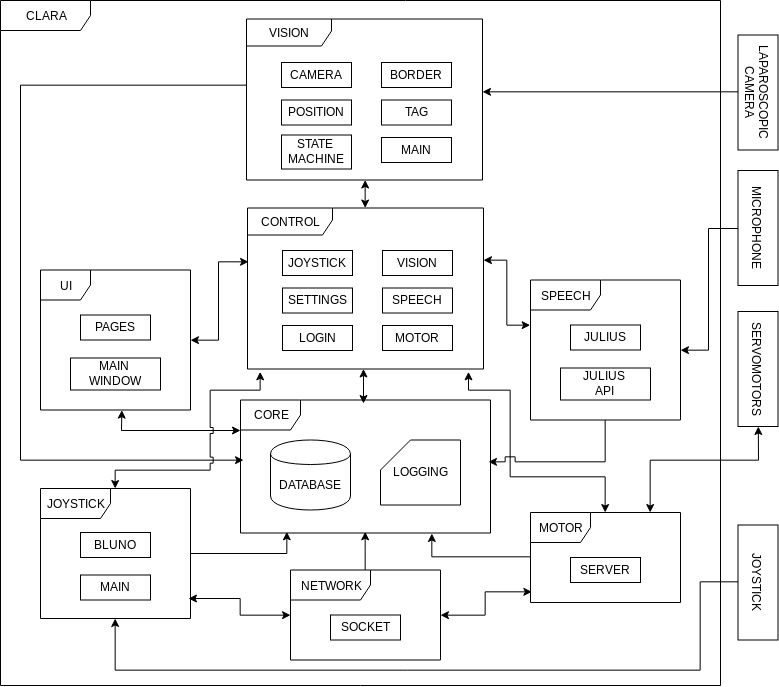
**Control -** This module is the main responsible for information exchange between the modules. Joystick, Vision, Speech and Motor submodules collect data from modules with corresponding names and transmit it to other modules that require them, e.g. Joystick commands received from the joystick device are translated into commands for the servo-motors through Motor module. Settings submodule is responsible for managing the system configuration options (set by the user through the UI), and Login creates and/or gives access to registered users to interact with the system.

**Network -** Contains classes to create and establish network connections with local or external IP address. Motor and Joystick modules uses it.

**Motor -** Receives information about servo-motor current connection status and send directional commands to them.

**UI -** Module responsible to control the information shown in the GUI and process action events, e.g. a click in a button at Settings page.

### The application runs 5 threads which are used to make critical modules (Control/Core/UI in the same thread, and Motor, Joystick, Speech and Vision each one in their own thread) always available during the system execution since these modules capture and process data in real-time.

  
Figure 2: CLARA software architecture overview.

**2.4 CLARA System's Requirements**

The usability and user experience were put as a priority to design CLARA. The basic usage flow, which was obtained after hospital visits and interviews with the surgical staff from HRAN (Hospital Regional da Asa Norte, Brasília), consists of login with a registered user, configure the system according the user preference, come back to the main menu and then access video window to proceed with the surgery.

Security is also a major priority in medical systems. Hence, for modules that use sockets to communicate, fault-tolerance protocol is important to establish secure tunnel connections between endpoints. The CLARA System has critical parts that cannot be off during surgery, so it is necessary that critical modules be always available during the duration of the procedure.

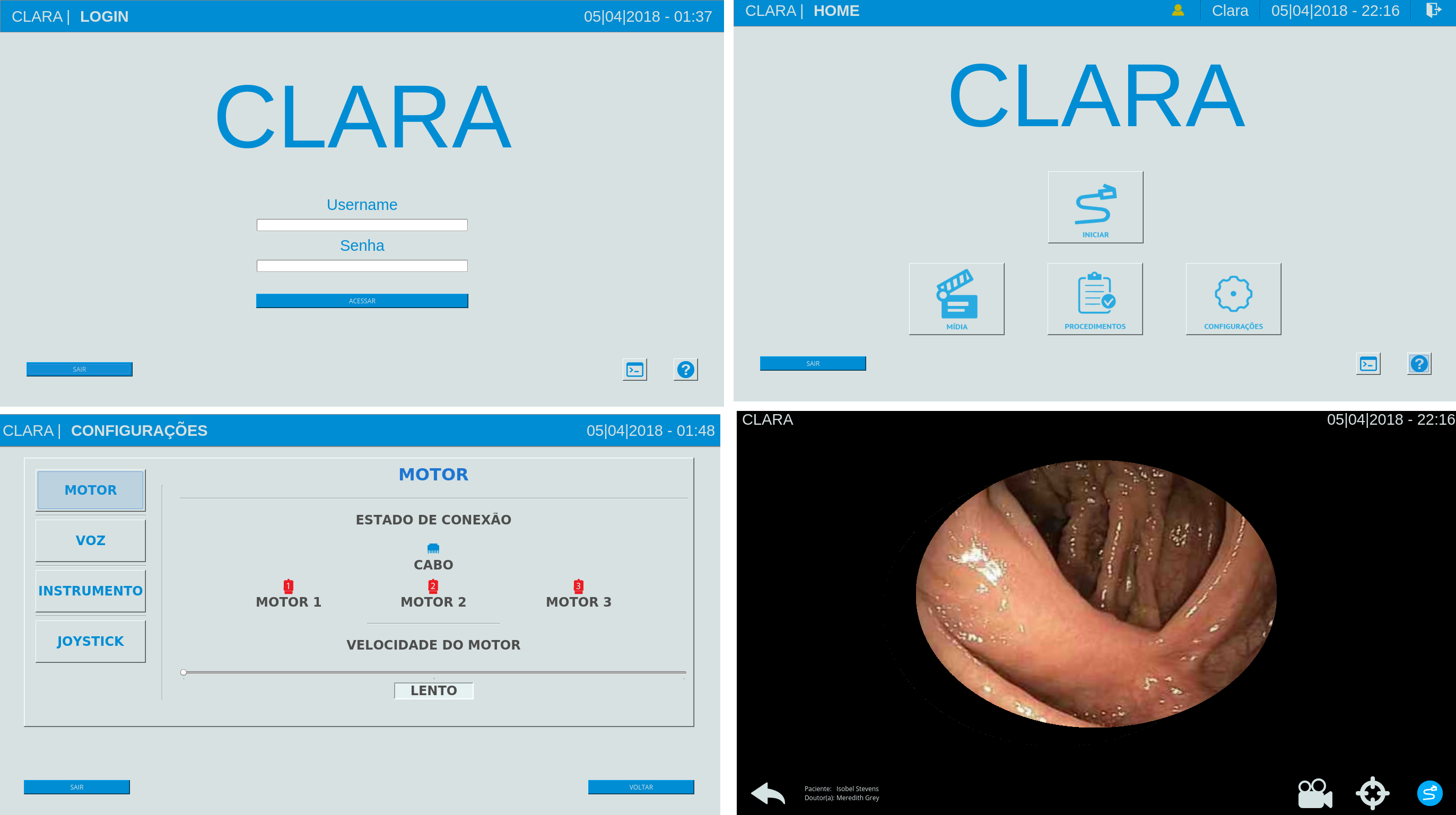
1. **Results**

The proposed software architecture has been implemented and validated in CLARA's first functional prototype, which is still in testing phase. The validation consisted on performing a series of commands using both the joystick device and voice recognition to evaluate the software overall performance and robustness to interfaces disconnections caused by possible occurrences in the operating room, such as cable disconnections and equipment power off. Such tests were monitored with help of the CLARA System's implemented logging functionalities.

It has been observed that the Motor client behaved as expected, sending the current connection status of the 3 robot servo-motors to the Motor server at 1Hz. If the Server did not receive the status after 1 second as expected, for instance due to a cable disconnection, the programmed fault treatment was activated:

1. Create socket, bind port and list to the port to communicate with the client (servo-motors at Raspberry Pi);
2. While connection is not accepted between the systems, wait a second and try to accept connection again;
3. After connection is setup, the system checks the stability of the connection every second;
4. If no data is received from the client socket after 1 second this means that the connection failed. Than a variable that informs the connection status is set to false, if any data is received than the connection status is set to true;
5. The event that runs every second to check the stability of the connection checks this status, if it’s false the motor server is restart.

The user interface was also validated with tests performed to check the correct operation of the proposed pages and graphic elements such as buttons and menus. As illustrated in Figure 3, the user experience started in the login page, which gives access to the main menu with buttons "Iniciar", "Mídia", "Procedimentos", and "Configurações". The system's settings could be successfully customized through the "Configurações" button that takes the user to the settings page, while the "Iniciar" button takes the user to the video page that features the current endoscope image frame superimposed with system information such as system time/date, connections status, and recognized commands.

  
 Figure 3: CLARA basic GUI flow: login page (top left), main menu (top right), settings page (bottom left), and video page (bottom right).

Moreover, the correct reception, interpretation and transmission of endoscope motion commands was verified for both joystick and voice recognition interfaces. From the tests it was confirmed that all predetermined commands ("cima", "baixo", "direita" and "esquerda" for top, down, right and left directions) were properly recognized by Speech module and transmitted to the servo-motors. The same was observed to Joystick module.

Finally, the automatic centralization of the surgical tool was successfully obtained for several initial positions of the tool in the endoscope image. The only cases were the centralization was not possible were due to false negatives of the tool segmentation algorithm, which is an ongoing work and will be improved for robustness of its image processing methods.

1. **Conclusion**

The design and implementation of a software framework for surgical systems such as CLARA demand a series of requirements such as user-friendly interfaces, open-source libraries to lower costs, and fault-tolerance protocols to assure security and improve user experience.

In this paper, the CLARA modular design was presented and the resultant implementation was analysed and discussed. Topics like extensibility, maintainability and reuse of code were a great concern of the developed approach and allow easier expansion of the system to incorporate possible future modules, such as image-based decision-support modules based on pre and intraoperative exams, and the use of smart laparoscopy tools, equipped with sensors for surgeon feedback. Other possible future work includes data encryption for security of the information transmitted from the core machine and the embedded system.

1. **Acknowledgements**

CLARA project is funded by the Brazilian Ministry of Health (Term of Cooperation no 121/2013, Process 25000.169843/2013-83).

I would like to express my deep gratitude to tenured assistant professor Mariana Bernardes and Professor Geovany Araújo for their valuable and constructive suggestions during the planning and development of this research work. Advice given by Tomaz Canabrava about Qt e C++ tools has been a great help in this project.

**References**

1. Buia, A., Stockhausen, F., Hanisch, E.: Laparoscopic surgery: A qualified systematic review. World J Methodol, 5(4):238–254 (2015).
2. Fichtinger, G., Kazanzides, P., Okamura, A. M., Hager, G. D., Whitcomb, L. L., Taylor, R. H.: Surgical and interventional robotics: Part II. IEEE Robot. Autom. Mag. (2008) 94–102.
3. Liu, S., Hemming, D., Luo, R. B., Reynolds, J., Delong, J. C., Sandler, B. J., Jacobsen, G. R., Horgan, S.: Solving the surgeon ergonomic crisis with surgical exosuit. Surg Endosc, 32(1):236–244 (2018).
4. Ortmaier, T.: Motion compensation in minimally invasive robotic surgery. PhD Thesis, Technical University of Munich, 2003, 137p.
5. Kommu, S. S., Rimington, P., Anderson, C., Rane, A.: Initial experience with the EndoAssist camera-holding robot in laparoscopic urological surgery. J Robot Surg, 1(2):133–137 (2007).
6. Long, J. A., Cinquin, P., Troccaz, J., Voros, S., Berkelman, P., Descotes, J. L., Letoublon, C., Rambeaud, J. J.: Development of miniaturized light endoscope-holder robot for laparoscopic surgery. J. Endourol., 21(8):911–914 (2007).
7. Hockelmann, M., Rudas, I. J., Fiorini, P., Kirchner, F., Haidegger, T.: Current Capabilities and Development Potential in Surgical Robotics, Intl. J. of Advanced Robotic Systems, 12 (61): 1–39 (2015).
8. Haidegger, T.: Surgical Robots of the Next Decade - New trends and paradigms in the 21th century. In Proceedings IEEE 30th Jubilee Neumann Colloquium, Budapest, 81-82 (2017).
9. Ma, J., Berkelman, P.: Control software design of a compact laparoscopic surgical robot system. In Proceedings of the 2006 IEEE Int. Conf. on Robotics and Automation, Orlando, 3813-3818 (2006).
10. Kapoor, Q. Deguet, Q. Kazanzides, P.: Software Components and Frameworks for Medical Robot Control. In Proceedings of the 2006 IEEE Int. Conf. on Intelligent Robots and Systems, Benjin, 2345-2350 (2006).
11. Wagner, A. A., Varkarakis, I. M., Link, R. E., Sullivan, W., Su, L. M.: Comparison of surgical performance during laparoscopic radical prostatectomy of two robotic camera holders, EndoAssist and AESOP: a pilot study. Urology, 68(1):70–74 (2006).
12. Zinchenko, K., Wu, C. Y., Song, K. T.: A Study on Motion Control of a Robotic Endoscope Holder Using Speech Recognition. In Proceedings IEEE International Conference on Industrial Technology (ICIT), Taipei, 1472-1475 (2016).
13. Dutkiewicz, P., Kielczewski M., Kowalski, M.: Visual tracking of surgical tools for laparoscopic surgery. In Proceedings of the Fourth International Workshop on Robot Motion and Control, Poland,23-28 (2004).
14. Nokia Corp. Qt : cross-platform application and ui framework (2012).
15. Akinobu, L., Tatsuya, K.: Recent development of open-source speech recognition engine julius. In Proceedings: APSIPA ASC: Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference, International Organizing Committee, 131–137 (2009).
16. Neto, N., Patrick, C., Klautau, A, Trancoso, I..: Free tools and resources for Brazilian Portuguese speech recognition. Journal of the Brazilian Computer Society 17(1): 53-68 (2011) .
17. Bradski, G.: The OpenCV Library. Dr. Dobb’s Journal of Software Tools, 2000.
18. Vergara, R. F., Miranda, G. M., Perruci, P. H. S., Bernardes, M. C.: Edges based surgical instrument tracking in laparoscopic images. XXV Brazilian Congress on Biomedical Engineering, 1223–1226 (2016). ISSN: 2359-3164.
19. Lab à Part, Gattlib, GitHub repository, <https://github.com/labapart/gattlib>. Last accessed 20 April 2018.
20. GNU. Sockets, <http://kirste.userpage.fu-berlin.de/chemnet/use/info/libc/libc_11.html>. Last accessed 05 April 2018.