






Development of an Arduino-based Control and Sensor System for a Robotic Laparoscopic Surgical Unit

Francisco Emmanuel Jr., III Munsayac* , Nilo Bugtai , Renann Baldovino , Noelle Marie Espiritu 
and Lowell Nathaniel Singson 

Institute of Biomedical
Engineering and Health
Technologies, De La Salle
University, Manila, Philippines

*Correspondence:
francisco.munsayac@dlsu.edu.ph

Abstract

The LAPARA System, a Philippine-made robotic surgical system, tested its control system in this paper. Three types of tests are then done to the system: PID Optimization Test, Position Checking, and Data Transfer Rate and Memory Bandwidth Testing. Results from the PID resulted in the values 2.32 for the P, 0.4 for the I, and 1.5 for the D to be chosen to ensure the system runs smoothly. The system was also able to run properly during Position Checking, though movement in the Pitch and Yaw required refinement due to the constraints. Also, the data transfer rate for the PC to Arduino Due connection yielded a 128kb/s speed, slower than the 480 Mbps rating, while the memory bandwidth testing yielded results that allowed for storage of 23,040 32_bit values. In conclusion, although minor adjustments were needed to refine the system, the LAPARA system was able to perform as intended.

Keywords

minimally invasive surgery (MIS), robot-assisted minimally invasive surgery (RMIS), robotic surgical systems, robotic surgery, control systems

INTRODUCTION

Robot-assisted minimally invasive surgery, or RMIS, has slowly gained fame in the various fields of cardiothoracic surgery, gynecology, and urology as minimally invasive surgery become preferred; benefits of minimally invasive surgery include less pain, fewer complications, and shorter hospitalization time. Also, compared to conventional laparoscopy, robotic surgery has advantageous factors such as 3D vision, motion scaling, and intuitive movements. These factors enhance the experience of the doctors and allow for more immersion with the added merit of accurate and precise robotics.

Among the numerous robotic surgical systems, the Da Vinci Surgical Robot, from Intuitive Surgical Inc., is considered the most distinguished; Da Vinci has set the standard for all international robotic surgical systems with its state-of-the-art technology and ergonomic design (Friedriche et al., 2018). The Da Vinci Surgical Robot features a master-slave system with a robotic cart (slave) and surgical console (master). The robotic cart has four (4) arms and a maneuverable base that can operate on patient-mounted laparoscopic

tools; the unit also includes a monitor to be used by assistants at the patient's side. In comparison, the surgeon console controls the robotic cart using instrument controllers and control pedals; it also features a high-resolution vision system, making it possible to see it using a laparoscope camera (Palep, 2009). Although some Da Vinci models have added features such as haptic feedback, most force feedback systems from surgical robotic systems can be considered negligible or lacking compared to human sense (Reiley et al., 2008).



Figure 1. *Da Vinci Surgical System.* (Palep, 2009)

Other robotic surgical systems are also being developed worldwide to improve and advance existing surgical robot systems. An example would be the system from the LAPARA Project, which aims to develop a cost-effective and locally manufactured robotic surgical system for the Filipino masses. This paper will discuss the LAPARA System, focusing on the developed control system. This paper will include literature on robotic surgical systems, a brief description of the LAPARA System, and the experimentation done to test the device.

Review of Related Literature

Robotic Surgery Standards

In order to manufacture and market robotic surgical systems, these systems must undergo evaluation from standardization organizations and receive approval. According to Haidegger (2019), for surgical robotic systems that aim to enter the international market, the International Electrotechnical Commission and the International Organization for Standardization are the organizations that create the standards concerning surgical robots. The latest standard released regarding the regulations for robotic surgical devices is IEC 80602-2-77, which specifically concerns robot-assisted surgical equipment and the required basic safety standards (Jacobs et al., 2017). For most countries, the Food and Drug Authority or the FDA is the regulating committee that ensures that the devices sold in their countries follow the standard. However, the responsibility for educating and training robotic surgical devices falls under the management of the manufacturing company or the administering unit (Espiritu et al., 2021). With regard to the status of robotic surgical standards in the Philippine setting, there is a lack of any official statement being released due to the small number of robotic surgical system users in the country, according to Chioson et al. (2020) and St. Luke's Unparalleled Technology and Expertise (2017).

Robotic Surgical Systems

Aside from the Intuitive Surgical robots, there are other commercially available surgical robot systems, and those that are still under development. Mirosurge is a robotic surgical system developed by the DLR, Institute of Robotics and Mechatronics that boasts of its modular design (Institute of Robotics and Mechatronics, n.d.-c). The system uses three (3) separate arms (called MIRO) instead of a single multi-armed robot, which allows for a more modular setup (Institute of Robotics and Mechatronics, n.d.-b). Figure 2 highlights the setup of the Mirosurge System.

Sehance, like Mirosurge, uses three (3) separate robotic arms, though of medium profile, as seen in Figure 3. The device was developed by TransEnterix and features advanced eye-sensing camera control that eases the surgeon's control over camera movement (Applied Dexterity, n.d.). TransEnterix also received FDA approval for their Sehance Ultrasonic System in 2019, allowing the device to be used for gynecological surgery, colorectal surgery, cholecystectomy, and inguinal hernia repair (Institute of Robotics and Mechatronics, n.d.-a). Another commercially available system would be the Flex Robotics System developed by Medrobotics Corporation. The system utilizes what they call a Flex Colorectal Drive that allows its singular arm to steer along a non-linear path (Ross & DeReus, 2018). An image of the system can be seen in Figure 4.

SPORT is another surgical robotic system that has a likeness to the Flex Robotics System. Similar to the Flex, it uses a singular, allowing for single-port laparoscopic surgery. It has an articulated instrument that can be divided into 3 sections (S works, distal section, and tip section) that allow for its unique translational movement (Titan Medical Inc., n.d.). Figure 5 is an image of the SPORT end effector tools.

Made in Italy, the Symani Surgical System features a double-armed robot configured for microsurgery. Developed by Medical Micro instruments SPA, the system has an ergonomic chair with manipulators and a 3D heads-up system for the visualization of the operation. The Symani system also features the world's smallest wristed instrument called the Symani Nano-Wrist Instrument (Medical Microinstruments, Inc., n.d.).



Figure 2. Mirosurge System (Institute of Robotics and Mechatronics, n.d.-c)



Figure 3. Sehance Robotic Surgical System (Applied Dexterity, n.d.)



Figure 4. Flex Robotics System (Ross & DeReus, 2018)



Figure 5. SPORT robotic surgical end effector (Titan Medical Inc., n.d.)



Figure 6. *Symani Surgical System (Medical Microinstruments, Inc., n.d.)*

Trends in Robotic Surgical Systems

With the development of newer and more advanced robotic surgical systems, some systems started developing based on certain trends such as Soft Robotics, Computer Vision and Augmented Reality, Sensors, and Techniques. Among the latest trends, soft robotics seems to be the most prominent, as systems like the ViaCath endoluminal system utilize this type of technology. This technology utilizes a snake-like robot, which makes the system more flexible and has higher dexterity and adaptability, which eases single-port surgeries. There are four types of actuations for soft robotics: motor-driven, tendon-driven, cable and pulley-driven, and shape-driven; although, there are some that try to use fluids or magnetics. Others have even considered the use of silicon with pneumatics (Chen et al., 2019; Omisore et al., 2022; Sagitov et al., 2019). For computer vision and augmented reality, the trend denotes the use of higher forms of visualization to enhance immersion during operation. A study by Kennedy-Mertz et al. (2021), expounds on the advantages of computer vision in the medical field, specifically for surgical applications. The results of their research entail the potential use of this type of computer vision in capturing and analyzing a patient's body pose, gestures, and identity, though there is a need to improve datasets as a deep-learning algorithm gave a 40% false positive rate due to the lack of reliable datasets. In another study by Hao et al. (2018), tool tracking of the Da Vinci Robotic System was made possible using silhouette rendering methods to measure the distance of the real tool from the rendered image and then particle filter algorithms to track the tool during partial obstructions. Other studies by Fontanelli et al. (2020) and Bandari et al. (2020a) note that computer vision can be used to improve force sensitivity. In an analogous way, sensors are used to improve haptic feedback, which gives the surgeon a sense of feeling, which is what makes RMIS more advantageous over traditional MIS. Two main types of tactile sensors are used in RMIS: electrical-based and optical, although most opt for the optical type due to safety concerns. There are also ongoing developments of hybrid sensors, which are more beneficial when considering the changing conditions of the environment. These are more lengthily discussed in the study of Bandari et al. (2020b). There are also techniques to learn for robotic surgery, as robotic surgery is not common for all surgeons. One technique is the Port Placement, which is needed for all laparoscopic operations. During MIS or laparoscopic surgery, the abdominal cavity is insufflated or filled with CO₂ to create more space for the tools to move. This technique is essential for tool movement and increased visibility (Maddah et al., 2020). Given that camera navigation is vital in MIS, autonomous camera navigation should be taught during the initial stages of training, as proposed by Mariani et al. (2020). The study's results proved there would be an increase in time-accuracy metrics and performance for those who underwent camera navigation training. Another technique that can be considered uncommon is the robotic transabdominal retromuscular umbilical prosthetic repair or TARUP, which utilizes a surgical procedure for vertical

incisional hernia (VIH). This technique is not commonly used in MIS due to the nature of the operation and is usually performed in open surgery (Rodrigues & López-Cano, 2021)

Surgical Robot Control Systems

According to a study by Shi et al. (2021), there are six levels of robotic surgical systems that pertain to the robot's autonomy. A robot with Level 0 autonomy (non-autonomy) is explained to give the surgeon full control of the robot system; this system usually does not provide any constraint or support. Level 1 autonomy robots fall under the category of robot assistance. Robots under this category give support or have a guiding feature that can help assist the surgeon. Most surgical robots are categorized under Level 1 autonomy except for certain robots with other specialized features. For Level 1 autonomy robots, common assistive technologies used by these robots would be tool tracking, eye tracking, and tissue interaction sensing or haptics. Figure 7 describes the flow of a control system of a Level 1 system in surgery. The flow of control is almost singular, with the surgeon maneuvering the controller, which then controls the robot that operates on the patient. Additionally, the robot gives feedback to the doctor, giving a more immersive feeling.

Compared to Level 1 systems, Level 2 systems can be attuned to having some autonomy when given certain tasks. Given certain specifications, the robot can perform surgical tasks wherein it takes control instead of the surgeon. Systems with this type of autonomy may gesture classification technology. Level 3 systems (conditional autonomy) give the robot more control and give the system the ability to perceive and understand so that it may perform tasks. Robots that use tissue modeling and advanced imaging usually fall under this category. After Level 3 systems are those categorized under high autonomy and listed under Level 4 systems. These systems can perform tasks independently, only needing the supervision of the surgeon. Robots that do organ or tumor segmentation may fall under this category. Level 5 systems are then considered fully autonomous systems. These are robots that can complete tasks independently without the need for a surgeon. Currently, no Level 5 systems are available on the market, although some research is being done.

Nonetheless, a proper control scheme is needed to ensure the smooth operation of the system. For surgical systems, there are three (3) applicable control schemes that can be used: PID control, Model Predictive Control, and Sliding Mode Control. Among the three, PID can be considered the simplest, which also makes it the most widely used for varying applications. It is a linear type of controller that, when given a value and its output, can form deviations depending on those values. Although PID control is easier to use, it will be a challenge for non-linearity. A solution would be to implement a Fuzzy PID control used in Level 0-1 systems. The Model Predictive Control (MPC) uses an object model with behavior tracking to predict the object's movement over a certain amount of time. This kind of control allows for smooth control over non-linear systems and time-delay systems. The final type of control structure is the Sliding Mode Control (or variable structure control) whose dynamic processing adjusts to the system's current state, allowing for faster response. Value boundaries are vital to this type of control scheme, as excess or lack of value may result in system instability. Each control strategy has its own strengths and weaknesses; thus, finding the right balance is necessary for the system to function smoothly.

Proportional integral derivative (PID) controllers are used to control different parameters or variables. In industrial settings, some of these variables are pressure, flow temperature, and speed (Elprocus, n.d). The level of control systems changes depending on the level of autonomy. It was noted that the higher the level of autonomy the controller requires to be more refined. The surgical robotic system is aimed at assisting surgeons, which would place it under level 1 autonomy. The suggested basic control system for this is seen in Figure 8 based on a review paper (Shi et al., 2021).

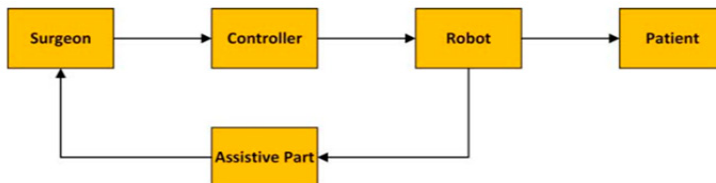


Figure 7. Control system for level 1 autonomous robot

The LAPARA System

The LAPARA System is composed of two (2) units: the patient-side unit and the surgeon-side unit. The patient-side unit is a robotic arm manipulator that performs surgical procedures with accuracy and precision according to the control of the surgeon. The surgeon-side unit is known as the console controller. It is handled directly by the surgeon and is used to control the robotic arm manipulator. The units are categorized by their placements inside the operating room. Other additional components are also classified under the patient or surgeon sides, except for the control system, which helps the two units communicate.

Patient-Side Unit

The patient-side unit or the robotic arm manipulator has three (3) sub-components that make up the unit: the robot arm, the trolley, and the electronic components box. The robot arm is mounted on the trolley while the electronic components box is incorporated into the trolley. The robot arm is a surgical device that can be manipulated to perform various operating procedures. The trolley serves as a table whose elevation can be adjusted according to the height needed. The electronic components box serves, as the name implies, and houses all electronics used to power and control the robot arm.

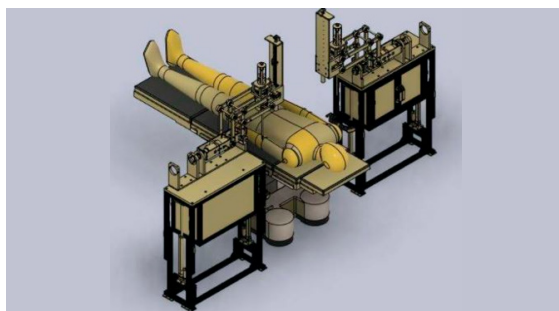


Figure 8. LAPARA Patient-Side Unit

Surgeon-side Unit

The surgeon-side unit or the console controller is mainly composed of the delta robot and the computer that serves as the central unit that connects the delta robot (controller) to the robot arm (manipulator). The delta robot is the main controller of the entire LAPARA system, and each motion of the delta robot is mapped to a corresponding motion of the robot arm. It is named after the type of mechanism it uses and houses the gimbal and pinching mechanism, which correspond to the movement of the end effector.



Figure 9. Delta Robot



Figure 10. Gimbal and Pinching Mechanism

Control System

Block Diagram

The control system of the LAPARA system is a Level 1 (robot assistance) system that uses PID control for its control structure (Kil et al., 2019). It comprises various electrical components and modules, including an Arduino that serves the purpose of a microcontroller board, Pololu motor drivers, DC motors, servo motor encoders, a 126x64 LCD display, and an AC-DC power supply. Each of the components of the LAPARA system performs a critical function with regard to the overall operation of the device.

Figure 11 shows the connection of the module and components used in LAPARA system. The Arduino Due, Motor Driver and Servo driver of the Lapara system is powered by a 12-24V AC-DC power supply, with its center being the Arduino which functions as a microcontroller board. This component is responsible for handling the data processing of all the other modules used by the system. For instance, the Arduino microcontroller handles the angle readings from the encoders and is also responsible for sending instructions to the motor driver, 126x64 LCD display, and the servo motors.

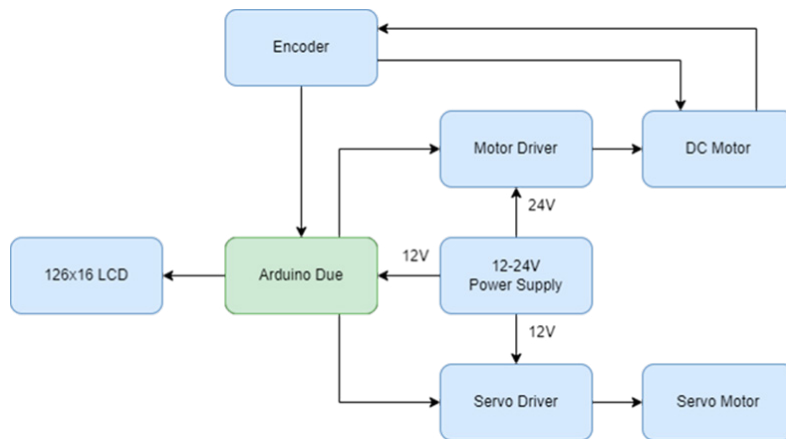


Figure 11. Block Diagram of LAPARA Module System

Control Diagram

Figure 12 shows the control system diagram of the LAPARA control system; the system is described as a closed-loop control system where the Arduino Due acts as the main processing system that receives that reference signal, generates control signals for non-linear actuators, encoders, servomotors, motor drivers and feedback from sensors. LCD is the output module that displays and monitors sensor/encoder

readings in real-time. Actuators (Control Plants) are components such as DC motors, servo motors, and motor drivers. That responds to control signals from Arduino to achieve the desired locational output, with the inclusion of a "Control Plant," in which actuators exhibit non-linear behavior. Feedback loops are sensors or encoders connected to the system actuators that provide real-time data on the Lapara delta and the robot arm motors' actual position. These feedback loops are compared in the Arduino due system through forward and inverse kinematics, which enables the system to be enveloped in a closed-loop control. Mechanical backlash represents the mechanical delay or gap in the lapara system design, where the movement of the motion changes and varies from one direction to another. Backlashes are a significant factor that affects the precision and response time of the system over time. Actuator response (Non-linear) outputs $Y(t) = K*(U(t) - X(t-\text{delta}))$, where the function represents gain(K), control signal $U(t)$, control signal $X(t-\text{delta})$ and time delay associated with backlash (delta).

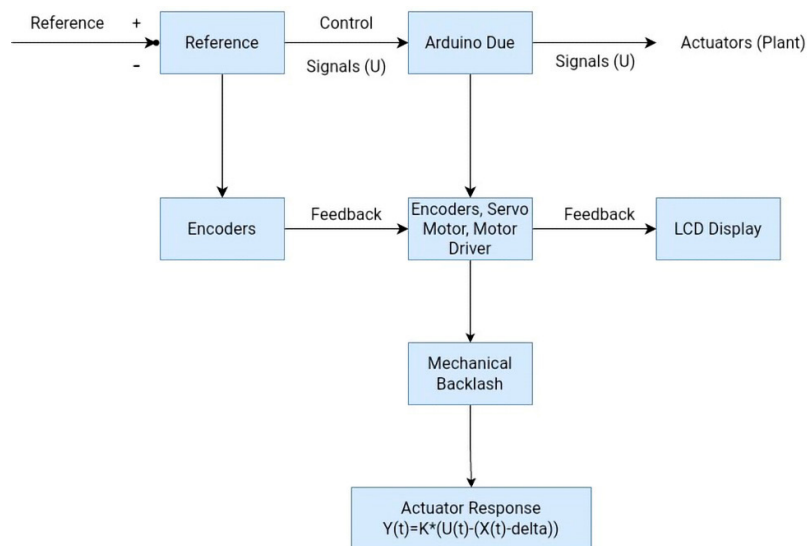


Figure 12. Control System Diagram

Mathematical Models for Control Design

Delta Robot

Delta robot is a parallel robot known for its speed and precision in the pick-and-place operation of relatively light objects. It consists of two platforms: the upper part, which has three mounted on it, and the end effector. The platform is connected through three parallelogram arms that restrain the orientation of the lower platform. It creates a base and movable platform, forming a pyramid-like structure that is an end-effector. The delta robot is responsible for the coordination of the movements of each individual arm to achieve the end-effector or robot arm position and orientation.

Forward Kinematics

The LAPARA delta robots use forward kinematics to determine the end-effector position based on joint configurations. The parallel architecture and equilateral triangle design on the LAPARA delta robot allow the utilization of the intersection of spheres derived from the rotational joints to determine the delta robots' Cartesian coordinates of the end-effector.

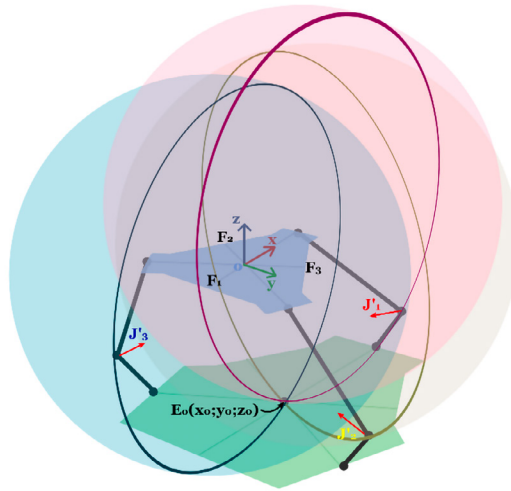


Figure 13. Delta Kinematics Diagram (Forward)

The mathematical formulation of the forward kinematic lies on the joint angles of each delta robot arm. The sphere is represented in a 3D space, where (x_i, y_i, z_i) is the center of the i^{th} sphere and R is the radius of each sphere, which varies from the delta robot arm length.

The formula to calculate the center of each i^{th} sphere is:

$$\begin{aligned}
 x_i &= R \cos(\theta_i) \\
 y_i &= R \sin(\theta_i) \\
 z_i &= L
 \end{aligned}$$

Where:

R is the radius of the circle

θ_i is the rotational joint angle

L is the length of the arms

Then, after the calculation of the radius, the formula to calculate the intersection point to determine the Cartesian coordinates of the end-effector (X, Y, Z) is as follows:

$$\begin{aligned}
 (X - x_1)^2 + (Y - y_1)^2 + (Z - z_1)^2 &= r^2 \\
 (X - x_2)^2 + (Y - y_2)^2 + (Z - z_2)^2 &= r^2 \\
 (X - x_3)^2 + (Y - y_3)^2 + (Z - z_3)^2 &= r^2
 \end{aligned}$$

Inverse Kinematics

Delta robots use inverse kinematics to determine the desired position of the end-effector to the required joint angles. Finding the joint variable to determine the inverse kinematics position of the center of the end effector allows us to acquire the desired position on the end-effector workspace. Where the position of the end effector is described by its Coordinate system (x, y, z) .

The lapara delta robot controller uses the regular delta robot configuration, which has three symmetrical arms that are connected to the delta base. The parameters of the delta arms are the Base radius (F), Bicep length (rf), Forearm length (re), End Effector radius (e), and Base-to-floor distance (b).

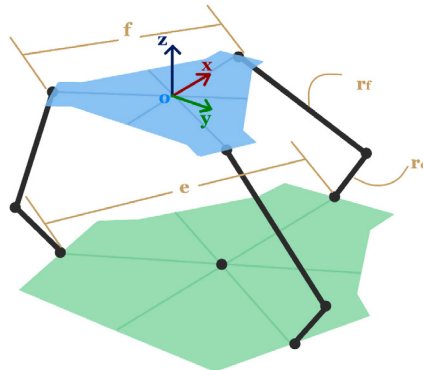


Figure 14. Delta Kinematics Diagram (Inverse)

The inverse kinematics formulation objective is to find the joint angles ($\theta_1, \theta_2, \theta_3$) in each Cartesian Coordinate position (x, y, z) at the very center of the end effector. The inverse kinematics formulation requires numerical methods to satisfy the nonlinear system of the equation.

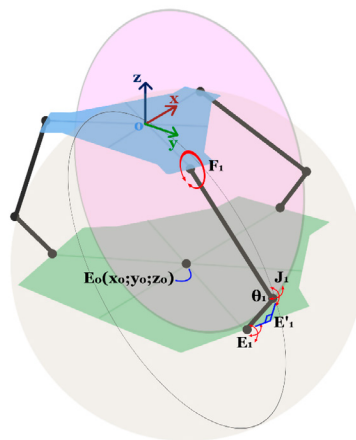


Figure 15. Delta Kinematics Diagram Cartesian Coordinates

The first step in the inverse kinematic formulation is to determine the vertical position of the end effector center (z_w) that refers to the given Cartesian Coordinates (x, y, z). The end effector center is compared, and its relationship is derived from the geometric configuration of the delta robot (y_w). At which the joint angles $\theta_1, \theta_2, \theta_3$ can be calculated with the use of trigonometric relationships and the delta robot geometric configurations and the formulation is given below:

$$\theta = 2 \tan^{-1}(Y_\omega, x)$$

$$\theta = 2 \tan^{-1}(Y_\omega, x - \sqrt{3}E) - 2 \tan^{-1}(Y_\omega, x + \sqrt{3}E)$$

$$\theta = 2 \tan^{-1}(Y_\omega, x + \sqrt{3}E) - 2 \tan^{-1}(Y_\omega, x - \sqrt{3}E)$$

To visualize the joint angles of the end-effector positioning a python script was written to calculate its joints angles along the x-axis.

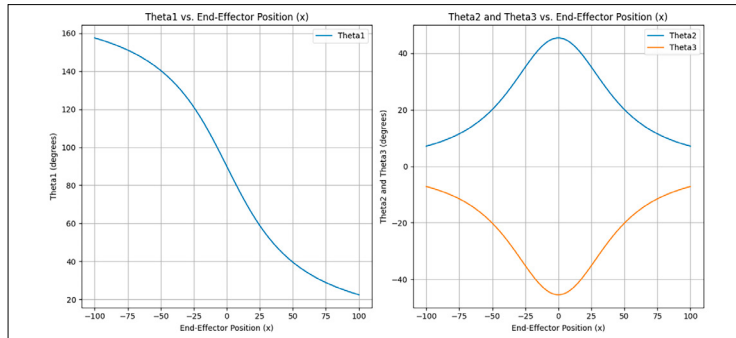


Figure 16. The graph shows the joint angle values given as theta1, theta2, and theta3 with respect to the End Effector position in one dimension

To visualize the relationship of joint angle θ_1 and the end-effector position, A 3D mesh is shown.

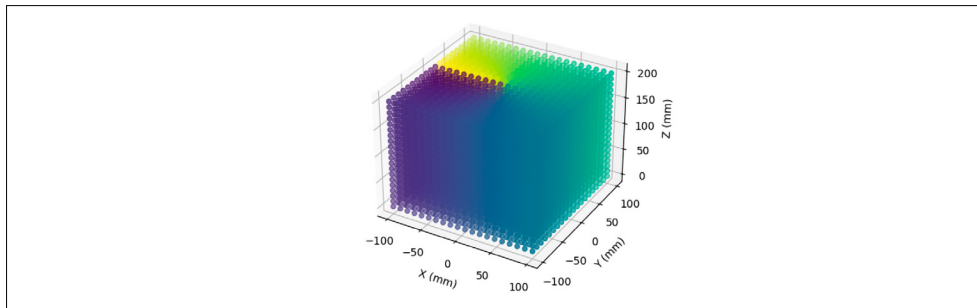


Figure 17. 3D mesh of combination of the different joint angles in a 3-dimensional coordinate system.

Software Development of LAPARA

This section discusses the flow of the LAPARA control system according to the command software that has been developed.

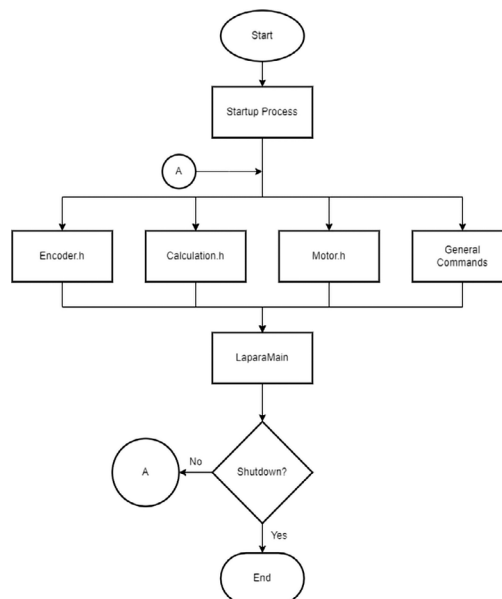


Figure 18. Main Program Flowchart

Figure 18 highlights the main program flowchart of the LAPARA control system. The control system consists of four program modules or header files: encoder processing, data calculation, motor controls, and general commands. During system operation, the LaparaMain module calls the necessary methods from the attached header files, and the coordinated execution of the executed functions constitutes the normal “steady state” operation of the device. Upon execution, the program will continue to run until it receives a stop or shutdown command. The explanations regarding the various header files of the system are as follows:

Startup Process

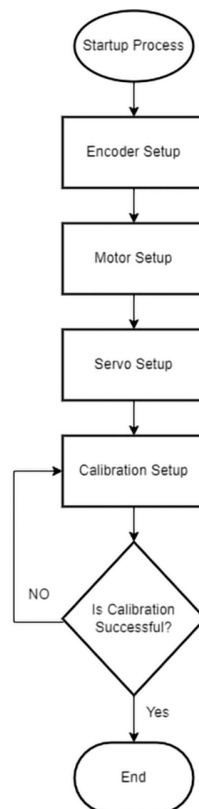


Figure 19. Startup Process Flowchart

The startup process module handles the initialization and setup of the variables and modules used in the control system. Within the process, the first step is the encoder setup, which handles the Bourns encoder variables, followed by the motor setup, which handles the servo and DC motors' PID configuration and variables. This is then followed by the calibration setup, which handles the switch's initiation and the surgical arm's initial positioning. Lastly, a five-second delay is executed. This delay aims to verify if all variables are initiated successfully.

Header Files

The purpose of this encoder class is to calculate the data processed by the Bourns encoder used by the control system in 16-bit cycles so that the angle readers of the device function as intended. Note that eight encoders are used within the device to determine the angles and position of each axis and joint of the surgical arm. Since the encoder must follow the same procedure when handling its calculated data to

lessen and improve the code quality of the system’s software, this encoder class was developed.

The general commands header is a temporary header file used to test new features developed in the LAPARA control system. Currently, the header file consists of general codes for data output, setup, and calibration.

The calculation header file handles the kinematic calculation of the LAPARA control system. The software uses a delta robot arm to control the movement of the surgical arm. Therefore, a kinematic algorithm called forward and inverse kinematics is used to translate the x-axis, y-axis, and z-axis coordinates given by the Bourns encoder connected to the center, left, and right arm of the delta robot. The forward and inverse kinematics computation is then translated into each calibrated map, where the LAPARA control system converts the data into its corresponding mapped coordinates on the surgical arm.

The Motor header/module handles all the motor control systems used in the LAPARA surgical robot arm; the design of the LAPARA surgical axis movements depends on the DC and servo motors. Each DC and servo motor is programmed to move in a certain direction and angle in the LAPARA control system. The control system uses PID and HPCPA9685 libraries to make it possible to control and label each servo and motor used. The functions found in the motor header file handle the setup, calculation, initiation, and output of the DC and Servo motors installed in the system.

Encoder	
+ int: Pin_Data	
+ int: Angle_Pos	
+ int: Zerocos	
+ int: Pos	
+ int: Crosspos	
+ int: TotalAngle	
+ int: OldPos	
+ int: AoldPos	
+ byte: Bit_BitStream[16]	
+ Encoders():Encoders	
+ Setup(int,string): void	
+ ReadEncoders(int):void	
+ ComputerEncoders(int):void	
+ ComputeOldpos():void	
+ ComputeEncoderPos(int):void	
+ OutputResult():void	
+ErrorVerificationStartup(int,int):void	
+ErrorVerification(int,int): void	

Figure 20. Encoder Class Headers

General Commands	Motor	Calculation
+ GyroSetup(): Void	+ MotorSetup():Void	+ Delta_Forward():Float
+ GyroOutput(): Void	+ X_MotorPID(): Void	+ Delta_AngleYZ():Int
+ LCDSetup(): Void	+ Y_MotorPID(): Void	+ Delta_CalcInverse():Int
+ LCDOutput(): Void	+ Z_MotorPID(): Void	+ Output_Forward():Void
+ LimitSwitch(): Void	+ Output_PID(): Void	+ X_Map():Int
	+ Motor_Initiate(): Void	+ Y_Map():Int
	+ ServoSetup(): Void	+ Z_Map():Int

Figure 21. General Commands Header, Motor Headers, & Calculation Headers (left-to-right)

LaparaMain

The LaparaMain function is the main body module of the LAPARA control system. It handles the entire processing of the LAPARA control system, from encoder readings, Kinematic calculations, DC motor and Servo motor controls, and data output of the LAPARA control system. At the start of the LAPARA main process, the Bourns encoder installed on the delta controller is initialized, and its reading will be processed by the forward kinematic algorithm.

The forward kinematic algorithm then converts the delta controller angles into x-y-z axis coordinates that will be mapped to the angle and position of the surgical robot.

After calculating the mapped axis and position of the LAPARA surgical arm joints, the control system will wait for changes in the delta controller, and if no change is detected, it will continue to compute until the change is detected, which will then initialize the motors installed in the LAPARA surgical arm. The motor installed will then follow the PID configuration programmed to the LAPARA control system, which will control the direction of the motor movements and determine if the axis is in the correct position and

angle. The LAPARA control system will then verify if the delta control and surgical coordinate axis are equal, and it will then output the readings on the serial monitor or LCD display installed on the LAPARA control system, ending the LaparaMain process. Otherwise, it will continue to compute the PID until the delta control and surgical coordinate axis are equal.

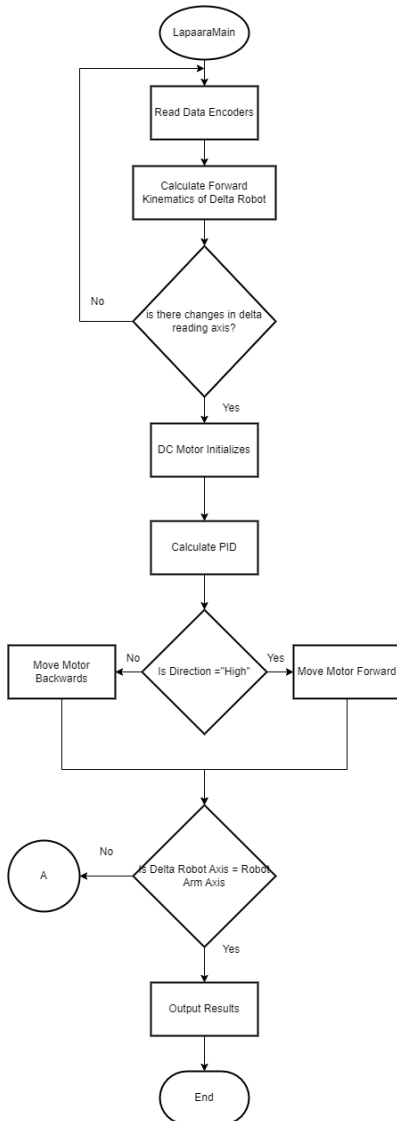


Figure 22. LaparaMain Program Flowchart

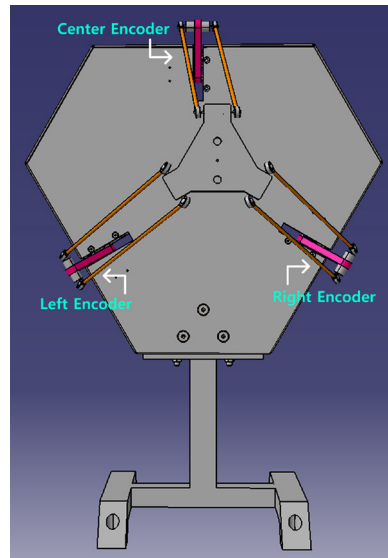


Figure 23. Delta Robot Controller

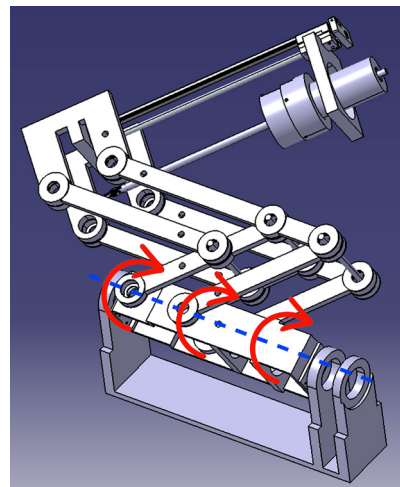


Figure 24. Robot Arm

METHODS

The LAPARA system has two parts: the surgeon side and the patient side as shown in Figure 25. The surgeon side comprises two delta robot controllers and a screen to be able to operate on the patient. The patient side comprises two robot arms. These robot arms are controlled by the delta robot and go into the incision created during prep. The surgeon's side of the system will be placed on a table to allow for the correct robot operating height.

Two (2) tests were done to measure the system's reliability: the PID Optimization Test and the Position Checking. These two tests were done to test the accuracy and precision of the movement of the robot arm. The PID Optimization test was used to measure the precision of the motors to achieve certain positions, while the Position Checking was used to assess the overall accuracy of movement of the entire patient-side unit (robot arm).

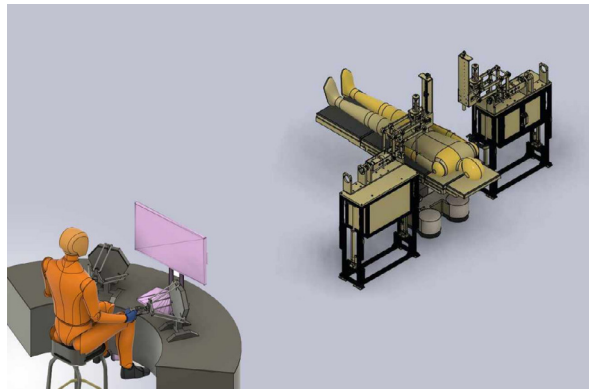


Figure 25. *Lapara System Setup*

PID Optimization Test

The necessary data on the technology to be assessed is gathered and collated in an organized manner; the data are then separated into 3 areas: the business model canvas (BMC), the technology roadmap (TRM), and the industry overview. Data about the said technology's business strategy is applied to a business model canvas for proper visualization, given that the technology does not have any existing BMC available. Data relating to the technology's trends and other related technology mapping are to be included under TRM, while data regarding the potential market of the technology industry will be included in the industry overview. References for these data were taken from interviews, company articles, government releases, and journal articles.

The PID Optimization test will also use the Ziegler –Nichols tuning method to determine the values needed to tune the created PID controller. This method aims to obtain good values for the following gain parameters: controller path gain (K_p), controller's integrator time constant (T_i), and the controller's derivative time constant (T_d), which is based on the measured feedback loop parameters. These measured feedback loop parameters are derived from the period (T_u) of the oscillation frequency at the stability limit and the gain margin (K_u) for loop stability. The method assumes that the system has a transfer function form of

$$\frac{K \cdot e^{-sT}}{(a+s)}$$

The researchers used the following equations to be able to determine the response parameters:

$$a = \sqrt{K^2 \cdot K_u^2 - 4\pi \cdot F_u^2}$$

$$\phi = -\tan^{-1}\left(2\pi \cdot \frac{F_u}{a}\right)$$

$$T = \frac{(-\pi - \phi)}{2\pi \cdot F_u}$$

After determining the response parameters, these values will be applied to the PID controller.

Position Checking

This test examined if the robot arm is moving properly and reaching the intended range it was designed for. Observations of the actual movement will be compared to the intended movement. Trials for each positional movement, such as pitch (Y-axis), Yaw (X-axis), roll, Z-axis, and end effector, will be done to assess the overall movement of the unit. For each movement, motion from the initial position to the maximum allowable position will be repeated 5-6 times, and any notable occurrences will be recorded.

Data Transfer Rate and Memory Bandwidth Testing

This test was done to calculate the delay and speed of the data transfer of the Arduino due board, which is used as the main processing unit of the LAPARA system. The data transfer rate and the memory bandwidth are the information to be gathered in this test. The theoretical value and actual value are then compared to each other.

RESULTS AND DISCUSSION

PID Optimization Test

The values for the PID were determined by using 12 volts as the source for each motor. For the P-Value, motors were set to reach 90 degrees and tested in intervals of 0.2.

The K_{MAX} was obtained by finding a value where the robot was unstable: $K_{MAX} = 36$. Following the Ziegler-Nichols PID controller, the following values were obtained:

1. $P = 18.0$
2. $I = 2$
3. $D = 0.0125$
4. $K_{MAX} = 36$
5. $f_0 = 1$

Aside from these, the results of the initial tests of the PID system are shown in the figures below. Figures 26 and 27 are the inverse kinematics versus the forward kinematics and the angle difference over time. Based on the graphs, it was noted that after a certain angle, the weight and backlash of the system generated oscillations. The PID was unable to compensate for the weight and backlash.

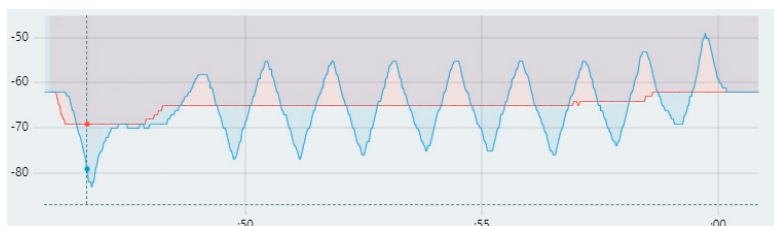


Figure 26. Inverse Kinematics (Red) vs Forward Kinematics (Blue)

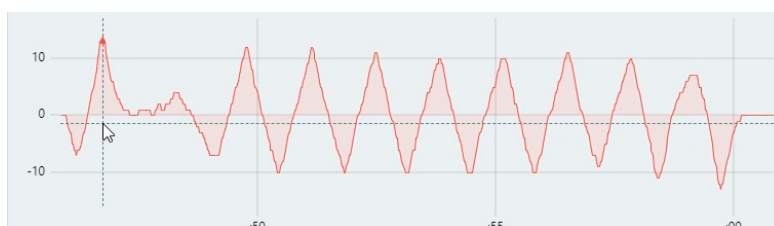


Figure 27. Difference of the Angle over time

Figures 28 and 29 are like figures 26 and 27, but in a different instance. These graphs show the PID being able to move the robot to the correct position.



Figure 28. *Inverse Kinematics (Red) vs Forward Kinematics (Blue)*



Figure 29. *Difference of the Angle over time*

Position Checking

Trials were done to check the actual movement of the robot arm compared to its intended.

a. Pitch (Y-axis)

The robot arm's pitch reached about ± 1 degrees (after correction) during clockwise movement. However, during counterclockwise movement, the pitch of the arm had a difference of ± 3 degrees. This might have occurred due to the backlash from the Y-axis of the robot arm. While moving the arm clockwise (moving up), the arm's weight might have contributed to the difference in backlash during upward and downward motion. Different PID Values for moving up (CW) and moving down (CCW) may be needed to compensate for this difference.

The pitch of the robot arm was able to reach about ± 1 degrees after correction. However, at around ~ 65 degrees, the PID cannot compensate for the weight and backlash of the gearbox, resulting in an oscillation of ~ 10 degrees.

b. Yaw (X-axis)

The yaw of the robot arm was able to have impressive results of only ± 0.6 degrees allowance in misalignment; however, it also suffers from the backlash due to its weight being the same as the Pitch. When operating from the range of ± 20 degrees, the arm could attain ± 0.6 degrees accuracy of motion. But when the arm exceeded 20 degrees on each side, the weight of the arm affected the precision of the yaw. Depending on the speed of the arm motion, a backlash value of ± 5 degrees to ± 1 degrees can be observed during operation. Different PID values may also be applicable for the Yaw, and having speed limits on the Yaw can help alleviate the backlash of the device.

c. Roll

With the use of a servo motor, the need for PID control is not required. With the servo's built-in encoder, the Roll of the robot arm can rotate 270 degrees and has absolute positioning. Due to the nature of the servo motor's control driver, speed control is not feasible.

d. Z-axis

The results of the Z-axis were the best among the other axes. It was able to reach ± 0.1 degrees through its movement along the screw rod. This result may have been due to having most of the weight of the system being carried by the screw rod. Hence, the I and D values are mostly negligible on its controls.

e. End Effector

The End effector system utilizes 3 servo motors to control the pinching and wrist. The result of the testing of the end effector shows a misalignment of about $\pm 1-2$ degrees; the backlash of the servo motor is the main cause of the misalignment. Proper mapping and calibration may be utilized to help reduce misalignment, but applying the mapping algorithm on the LAPARA control system is currently tedious.

Data Transfer Rate and Memory Bandwidth Testing

a. Data Transfer Rate

The data transfer rate, as indicated by the Arduino due documentation, stated that USB 2.0 Device/Mini host can reach 480 Mbps; this will be set as our theoretical data for the computation and comparison of its data transfer rate. Upon testing the actual Arduino due to the data transfer rate, a PC-to-De connection setup yielded a 128kb/s speed. The slower speed of the Arduino to the connection may have been due to the inferior quality of the USB and connector used in the actual test. Before that, the transfer size of the data may affect the data transfer rate of the board.

b. Memory Bandwidth Testing

The memory bandwidth of the Arduino due documentation stated that due boards have 96 Kbytes contiguous SRAM and 4k non-contiguous SRAM. Upon testing, the due board does store 90K bytes of data, which could amount to 23040 32_bit values.

CONCLUSION

Developing a robotic surgical unit requires many factors. Each factor plays a significant role in cohesion in one unique system. For the control system, the PID values are used to control the system's movement and need considered for finer movements. The ideal P-value was found to be 2.32 as it gave the closest position to what was intended. A 0.4 value for the I-value was chosen for having the least time needed for stabilization of position while 1.5 was decided upon as other values either exceeded the necessary positional value or fell short of the ideal. The proper execution of the control system was ensured by position checking. Each motor corresponds to a certain motion by the unit. All Pitch, Yaw, Roll, Z-axis, and End Effector motions worked properly and can move according to the intended motion. Although, there are observations that need to be considered for improvement. Due to the robot's weight, Pitch and Yaw motion may have increased backlash, which causes inconsistencies in the movement. Roll motion can be performed successfully, though speed control is not feasible. The Z-axis had the most ideal accuracy with 0.1 degrees allowance. Also, mapping and calibration should be considered for the End Effector, as backlash from the motor causes 1 - 2 degrees of misalignment. Overall, the tests were successful, and the system proved to be able to move as intended, though refining the movement was necessary.

REFERENCES

- Applied Dexterity. (n.d.). History. Retrieved March 29, 2023, from <http://applieddexterity.com/about/history/>
- Bandari, N., Dargahi, J., & Packirisamy, M. (2020a). Miniaturized optical force sensor for minimally invasive surgery with learning-based nonlinear calibration. *IEEE Sensors Journal*, 20(7), 3579–3592. <https://doi.org/10.1109/JSEN.2019.2959269>

- Bandari, N., Dargahi, J., & Packirisamy, M. (2020b). Tactile sensors for minimally invasive surgery: A review of the state-of-the-art, applications, and perspectives. *IEEE Access*, 8, 7682–7708. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8943415>
- Chen, M., Wang, D., Zou, J., Sun, L., Sun, J., & Jin, G. (2019). A multi-module soft robotic arm with soft end effector for minimally invasive surgery. *2019 2nd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, 461–465.
- Chioson, F. B., Espiritu, N. M., Munsayac, F. E., Jimenez, F., Lindo, D. E., Santos, M. B., Reyes, J., Tan, L. J. A. F., Dajay, R. C. R., Baldovino, R. G., & Bugtai, N. T. (2020). Recent advancements in robotic minimally invasive surgery: A review from the perspective of robotic surgery in the Philippines. *020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–7. <https://doi.org/10.1109/HNICEM51456.2020.9400042>
- Elprocus. (n.d.). *How does a PID controller work? - structure & tuning methods*. ElProCus - Electronic Projects for Engineering Students. <https://www.elprocus.com/the-working-of-a-pid-controller/>
- Espiritu, N. M., Balcita, V. D. G., Goy, T. P. P., Perez, E. M. S., Baldovino, R. G., Munsayac, F. E. T., & Bugtai, N. T. (2021). A 2021 review on recent advancements in robotic-assisted minimally invasive surgery: The Philippines perspective. *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–5. <https://doi.org/10.1109/hnicem54116.2021.9731947>
- Fontanelli, G. A., Buonocore, L. R., Ficuciello, F., Villani, L., & Siciliano, B. (2020). An external force sensing system for minimally invasive robotic surgery. *IEEE/ASME Transactions on Mechatronics*, 25(3), 1543–1554. <https://doi.org/10.1109/TMECH.2020.2979027>
- Friedrich, D. T., Dürselen, L., Mayer, B., Hacker, S., Schall, F., Hahn, J., Hoffmann, T. K., Schuler, P. J., & Greve, J. (2018). Features of haptic and tactile feedback in TORS—a comparison of available surgical systems. *Journal of Robotic Surgery*, 12(1), 103–108. <https://doi.org/10.1007/s11701-017-0702-4>
- Haidegger, T. (2019). Autonomy for Surgical Robots: Concepts and Paradigms. *IEEE Transactions on Medical Robotics and Bionics*, 1(2), 65–76. <https://doi.org/10.1109/TMRB.2019.2913282>
- Hao, R., Özgüner, O., & Çavuşoğlu, M. C. (2018). Vision-based surgical tool pose estimation for the da Vinci® robotic surgical system. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1298–1305. <https://doi.org/10.1109/IROS.2018.8594471>
- Institute of Robotics and Mechatronics. (n.d.-a). *Mica*. Www.dlr.de. Retrieved June 5, 2024, from <https://www.dlr.de/en/rm/research/robotic-systems/hands/mica#gallery/29823>
- Institute of Robotics and Mechatronics. (n.d.-b). *Miro*. Retrieved March 29, 2023, from <https://www.dlr.de/en/rm/research/robotic-systems/arms/miro#gallery/29804>
- Institute of Robotics and Mechatronics. (n.d.-c). *MiroSurge*. Retrieved March 29, 2023, from <https://www.dlr.de/en/rm/research/robotic-systems/multi-arm/mirosurge#gallery/28728>
- Jacobs, T., Veneman, J., Virk, G. S., & Haidegger, T. (2017, December 14). *The flourishing landscape of robot standardization*. IEEE Standards University. <https://www.standardsuniversity.org/e-magazine/december-2017/flourishing-landscape-robot-standardization/>
- Kennedy-Metz, L. R., Mascagni, P., Torralba, A., Dias, R. D., Perona, P., Shah, J. A., Padoy, N., & Zenati, M. A. (2021). Computer vision in the operating room: Opportunities and caveats. *IEEE Transactions on Medical Robotics and Bionics*, 3(1), 2–10. <https://doi.org/10.1109/tmr.2020.3040002>
- Kil, I., Singapogu, R. B., & Groff, R. E. (2019). Needle entry angle & force: Vision-enabled force-based metrics to assess surgical suturing skill. *2019 International Symposium on Medical Robotics (ISMR)*, 1–6. <https://doi.org/10.1109/ISMR.2019.8710175>
- Maddah, M. R., Dumas, C., Gauthier, O., Fusellier, M., & Cao, C. G. L. (2020). Measuring organ shift and deformation for port placement in robot-assisted minimally invasive surgery. *Laparoscopic, Endoscopic and Robotic Surgery*, 3(4), 99–106. <https://doi.org/10.1016/j.lers.2020.09.002>
- Mariani, A., Colaci, G., Da Col, T., Sanna, N., Vendrame, E., Menciassi, A., & De Momi, E. (2020). An experimental comparison towards autonomous camera navigation to optimize training in robot assisted surgery. *IEEE Robotics and Automation Letters*, 5(2), 1461–1467. <https://doi.org/10.1109/ra.2020.2965067>

- Medical Microinstruments, Inc. (n.d.). *Robotic surgical systems: How it works*. <https://www.mmimicro.com/our-technology/how-it-works/>
- Omisore, O. M., Han, S., Xiong, J., Li, H., Li, Z., & Wang, L. (2022). A review on flexible robotic systems for minimally invasive surgery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(1), 631–644. <https://doi.org/10.1109/TSMC.2020.3026174>
- Palep, J. H. (2009). Robotic assisted minimally invasive surgery. *Journal of Minimal Access Surgery*, 5(1), 1–7. <https://doi.org/10.4103/0972-9941.51313>
- Reiley, C. E., Akinbiyi, T., Burschka, D., Chang, D. C., Okamura, A. M., & Yuh, D. D. (2008). Effects of visual force feedback on robot-assisted surgical task performance. *The Journal of Thoracic and Cardiovascular Surgery*, 135(1), 196–202. <https://doi.org/10.1016/j.jtcvs.2007.08.043>
- Rodrigues, V., & López-Cano, M. (2021). TARUP technique. Advantages of minimally invasive robot-assisted abdominal wall surgery. *Cirugía Española (English Edition)*, 99(4), 302–305. <https://doi.org/10.1016/j.cireng.2021.03.009>
- Ross, S., & DeReus, H. (2018, April 2). *Flex robotic system and flex colorectal drive*. Society of American Gastrointestinal and Endoscopic Surgeons (SAGES). <https://www.sages.org/publications/tavac/flex-robotic-system-and-flex-colorectal-drive/>
- Sagitov, A., Gavrilova, L., Tsoy, T., & Li, H. (2019). Design of simple one-arm surgical robot for minimally invasive surgery. *2019 12th International Conference on Developments in ESystems Engineering (DeSE)*, 500–503. <https://doi.org/10.1109/DeSE.2019.00097>
- Shi, Y., Singh, S. K., & Yang, L. (2021). Classical control strategies used in recent surgical robots. *Journal of Physics: Conference Series*, 1922(1), 012010. <https://doi.org/10.1088/1742-6596/1922/1/012010>
- St. Luke's unparalleled technology and expertise*. (2017, October 2). St. Luke's Medical Center. <https://www.stlukes.com.ph/news-and-events/news-and-press-release/st-lukes-unparalleled-technology-and-expertise>
- Titan Medical Inc., (n.d.). *Intellectual property*. Retrieved March 29, 2023 from <https://titanmedicalinc.com/intellectual-property/>

How to cite this article:

Munsayac, F.E.J. I., Bugtai, N., Baldovino, R., Espiritu, N. M., & Singson, L. N. (2024). Development of an Arduino-based Control and Sensor System for a Robotic Laparoscopic Surgical Unit. *Recoletos Multidisciplinary Research Journal* 12(1), 125-144. <https://doi.org/10.32871/rmrj2412.01.10>