

Technological Institute of Costa Rica

Electronic Engineering School



Modeling and controlling a robotized medical instrument

Thesis document under consideration to qualify for the Master degree in Electronics with
emphasis on Embedded Systems

Oscar Mauricio Caravaca Mora

Friday, May 06, 2016

This document states that the thesis has been entirely done by me, using and applying literature on the topic and introducing knowledge and own experimental results.

Where I used literature I proceeded indicating the sources with corresponding bibliographical references. Consequently, I take full responsibility for the thesis work performed and the contents of this document.

Oscar Mauricio Caravaca Mora

Cartago, May 6, 2016

ID: 5-0334-0128

Technological Institute of Costa Rica
Electronics Engineering School
Master's Thesis
Evaluator Court

Thesis defended before this Court Assessor as a requisite for electing academic master's degree from the Technological Institute of Costa Rica.

Tribunal Members



PhD. Philippe POIGNET
Director of LIRMM



PhD. Salih ABDELAZIZ
Supervisor



PhD. Nabil ZEMITI
Examiner



PhD. Luis Alonso SÁNCHEZ SECADES
Examiner

Members of the Tribunal attest that this master's thesis is approved and meets the standards set by the School of Electronics Engineering.

Cartago, April 5, 2016

Abstract

During the last decade, Minimally Invasive Surgeries (MIS) have become a trend in the medical field, given the numerous benefits such as: reduced trauma, shorter recovery time and minimal postoperative complications. It has been promoted by the development of new technologies mainly in the field of robotics. Robotic technologies has emerged as a great option to create solutions that enhance the surgeon's perception and dexterity and allows access to often inaccessible human body places by keeping the smallest incision as possible.

This document describes the modeling and control of a hand-held robotized medical instrument using stereo vision feedback, with the purpose of providing automatic assistance to the surgeon during the surgical procedure. The system modeling, simulation, experimental evaluation and results will be presented in this thesis document.

Keywords: Minimally Invasive Surgery (MIS), stereo vision, visual servoing, medical robotics, hand-held robotized instrument, augmented reality.

To my family that always pray for me.

Special Thanks

To my thesis supervisor, Salih Abdelaziz, for his orientation during the project course; to Director Philippe Poignet for giving me the opportunity to work at LIRMM laboratory, to the jury members: Alonso Sánchez and Nabil Zemiti, for their support and the everyone else at DEXTER team for the different constructive suggestions given during the development of such thesis.

Oscar Mauricio Caravaca Mora

Cartago, May 6, 2016

Content

Introduction	1
General objective	2
Specific objectives	2
Chapter 1 Medical context	3
Chapter 2 Modeling	7
2.1 Kinematic model	7
2.1.1 Direct kinematic model	8
2.1.2 Jacobian computation	11
2.2 DC motor Model.....	14
2.3 Camera model.....	14
2.4 Solution Model	15
Chapter 3 Simulation	16
3.1 Simulation of kinematic model with ideal plant.....	16
Chapter 4 Experimental Evaluation	18
4.1 Platform Description	18
4.1.1 Controller Device	18
4.1.2 Camera Stereo Calibration.....	20
4.1.3 System Identification.....	21
4.1.4 Software.....	23
4.2 Experimental Results.....	26
Conclusions and Perspectives	30
Bibliography.....	31
Appendix A.....	33
A.1 PI Controller Parameters, units conversion EPOS3 to SI Units	33
A.2 EPOS3 state machine.....	33
A.3 Software components	34
A.4 Graphic card information.....	35
A.5 Connections	35

Figures Index

Figure 1. The proposed solution.....	1
Figure 2. System proposed in [4], image taken from the same reference.	3
Figure 3. System described in [5], image taken from the same reference.....	4
Figure 4. System presented in [6], image taken from the same reference.....	4
Figure 5. Anubis platform, image taken from [10].....	5
Figure 6. a) Bending, tip rotation and grasping are the 3DOF of JAIMY™ ,.....	5
Figure 7. JAIMY™ general description [2].	6
Figure 8. Flexible section of the JAIMY™ instrument.....	7
Figure 9. Kinematic model of the JAYMI™ instrument.....	7
Figure 10. Mathematical model of a commercial DC Motor where $km = kb$	14
Figure 11. Image-based visual servoing of the JAIMY™ 15	15
Figure 12. Closed loop to validate the kinematic model.	16
Figure 13. Kinematic model simulation result considering ideal dc motor model.....	17
Figure 14. Controller architecture implemented by the EPOS3.	19
Figure 15. Current controller response after auto tuning, blue → current demand value, red → current actual value.....	19
Figure 16. Velocity controller response after manual tuning, blue → velocity demand value, red → velocity actual value.	20
Figure 17. First pair image of the pictures used to do the stereo calibration.	20
Figure 18. Endoscopic lenses plus Basler cameras setup.	21
Figure 19. Image of the tip of the JAIMY™ being tracked by using markers.....	21
Figure 20. Linear regression method to estimate the radius of the pulley.....	23
Figure 21. User interface developed to control the system.	24
Figure 22. Block diagram of the solution.	25
Figure 23. Automatic positioning of the tip of the JAIMY™ in the desired point.	27
Figure 24. Position response of the system.	27
Figure 25. Velocity response of the system.....	28
Figure 26. Disparity map and 3D information for different blocksize of disparity, a) Original image, b) Disparity map blocksize = 3, c) 3D map blocksize = 3, d) Disparity map blocksize = 17, e) 3D map blocksize = 17.....	29
Figure 27. EPOS3 state machine, taken from [25].	33

Index of Tables

Table 1. Technical characteristics of the JAIMY TM	18
Table 2. Experimental values of θ_{mj} , θ_{lj}	22
Table 3. Device control commands, taken from [25]	34
Table 4. Graphic card specifications used in the project, refer to [27].....	35
Table 5. JAIMY TM and EPOS3 connections, refer to [28], [29]	35

Introduction

In the medical area, there is always an interest for exploring new approaches that provide better results, with reduced trauma, shorter recovery time for the patient and enhanced perception and ability for the surgeon during the procedure. Minimally Invasive Surgeries offers solutions to reach the aforementioned objectives.

DEXTER is a research team which is part of the Robotic department at Montpellier Laboratory of Informatics, Robotics and Microelectronics (LIRMM). One of DEXTER's research topics of interest is the development of robotic assistance for medical applications[1], therefore MIS is the field of interest for this research team. DEXTER has several platforms dedicated to research, one of them is the hand-held robotized instrument named JAIMY™ from Endocontrol company[2]. It has three 3 degrees of freedom (DOF): bending, tip rotation and grasping. It is equipped with two DC motors for bending and tip rotation, grasping is performed manually. JAIMY's™ manipulation is manual. More details about this instrument will be presented in following sections. Since the instrument has two motors, the interest of incorporating external automatic control emerged from the need of providing new capabilities to assist the surgeon. The desired new capabilities are aiming to:

- add stereoscopic visualization,
- eliminate the fulcrum effect,
- improve the hand-eye coordination.

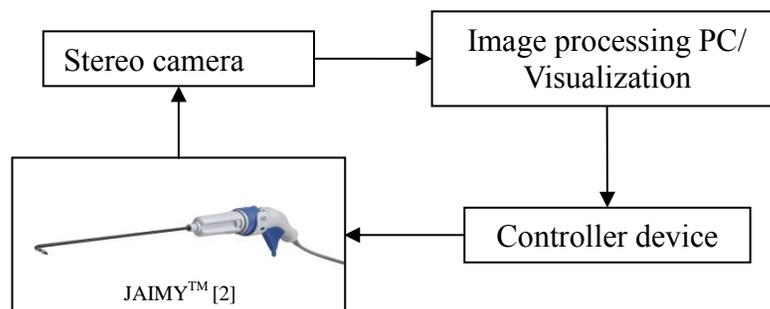


Figure 1. The proposed solution.

To reach these purposes, a vision platform has been developed to control the JAIMY™. This platform provides a stereoscopic feedback. This system takes from the image the actual configuration of the end-effector, then using the instrument kinematic model it computes the current position of the tip in regards to any point in the image; this will allow the system to know how far or how close is the point of interest from the tip. If it is in the workspace of the surgical instrument, the system will automatically pose the tip of the end-effector in the desired location; this is possible by implementing a velocity controller. The Figure 1 above shows a scheme of the proposed solution.

General objective

Developing a system capable of assisting surgeons to manipulate the JAIMY™ instrument, based on stereoscopic visual feedback.

Specific objectives

- Determine the instrument kinematic model in order to define its workspace and pose estimation.
- Develop a visual feedback system in order to control the position of the end-effector.
- Identify the model missing parameters.
- To validate the system functionality.

Chapter 1 Medical context

Minimally Invasive Surgery is described in [3] as the surgical process in which the size of the incisions are reduced to less than 1 cm, or by using natural access port of the human body in order to reach the region of interest. On the contrary, open surgery requires incisions large enough for the surgeon to see and place hands and instruments directly into the operating zone. As mentioned before, MIS provides important benefits to patient if compared to open surgery. However, the limited access and the reduced perception make MIS gestures tough to perform. Medical robotic intends to solve some difficulties associated to MIS, such as the eye-hand coordination and loss of internal mobility due to the kinematic constraints induced by the trocar. For the objective of this work, the intention is to improve visual perception when using the JAIMY™ instrument adding automatic positioning and guidance by stereoscopic visual servoing. Others works related have been done with different instruments and purposes. In [4] Krupa et al. developed a robotic vision system that automatically position a surgical instrument by using laser pointers and LEDs attached to the tip as optical markers. The surgical instrument is rigid and is mounted in a robotic together with an endoscope to get the visual feedback. In this case, a monocular endoscope was used. The image of the Figure 2 illustrates the system set up.

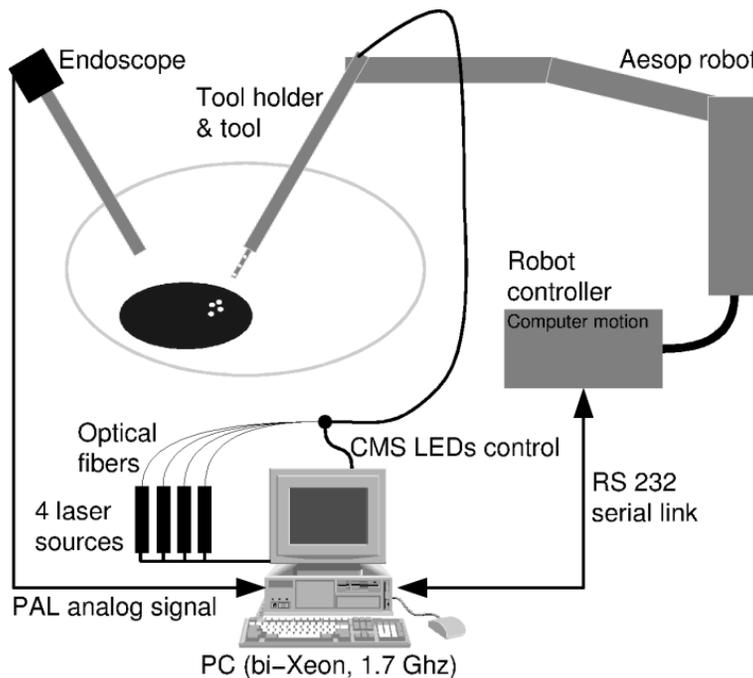


Figure 2. System proposed in [4], image taken from the same reference.

Another related system is described in [5]. It has been done by using a dual arm robot configuration, uncalibrated stereo visual servoing using colored markers attached to the shaft and its corresponding control and image processing units. The instruments are

rigid, the objective of this project was to perform automatic partial knot tying, Figure 3 shows the system hardware configuration.

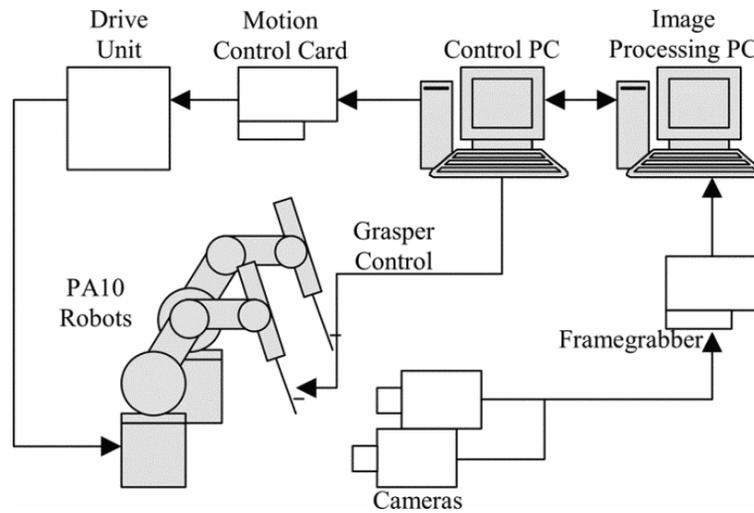


Figure 3. System described in [5], image taken from the same reference.

Becker et al. presented in [6] and [7], a system for robot-assisted laser photocoagulation using a hand-held micromanipulator known as Micron (cf. Figure 4). It uses specialized software for planning patterns of laser burns on the image of the patient's retina; to apply then the patterns planned using visual servoing techniques. This system implements stereoscopic visual feedback through microscope which also provides tremor cancellation and, the target application is for vitreoretinal surgery.

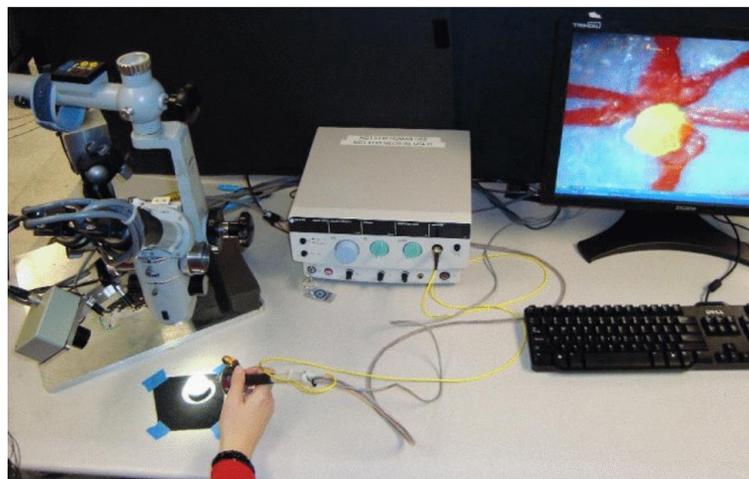


Figure 4. System presented in [6], image taken from the same reference.

Image-based control for a robotic endoscope holder has been proposed in [8]. This system is for laparoscopic applications. It implements visual control to keep the surgical instruments at the center of the image. In this case, the endoscope is being controlled.

Vision-based control in surgeries is not limited only to control rigid instruments, there are also flexible instruments being developed in order to improve access in more complex procedures. For instance, in [9] marker-based and marker-less method for pose estimation of flexible instruments are explained. A comparison between model-based and learning algorithm are presented in [10], to estimate the position and configuration of Karl Storz Anubis platform (cf. Figure 5). Visual markers are used in this case to track the instrument; this platform is showed below in Figure 5.

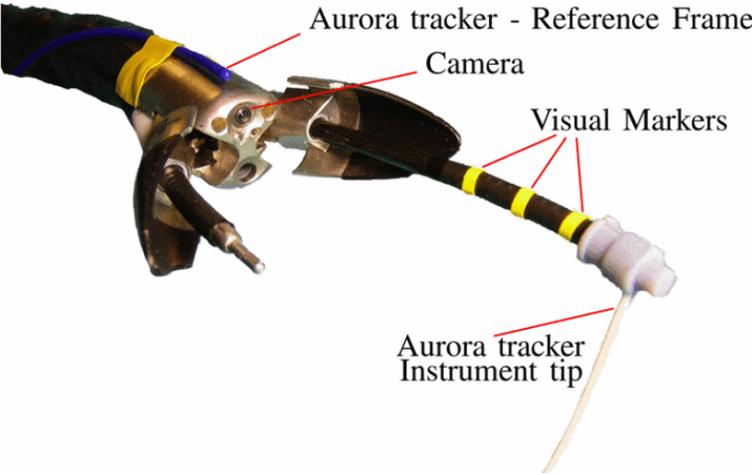


Figure 5. Anubis platform, image taken from [10].

The objective of this work is implementing a visual-based control to pose the end-effector of the JAIMY™ instrument, using stereoscopic vision to provide 3D real world information to assist the surgeon. It is important to provide details about the surgical instrument. Developed by Endocontrol Company, JAIMY™ is a hand-held rigid instrument with 5 mm diameter. Designed for suturing in laparoscopic surgery, the JAIMY™ has 3 internal mobility functions: 80 degrees bending, infinite tip rotation and grasping. Bending and rotation are motorized, whereas grasping is performed manually using a trigger. It has a control ring to execute each movement. The system comes with its own power supply console, Figure 6 shows the 3DOF and Figure 7 illustrates an instrument general view.

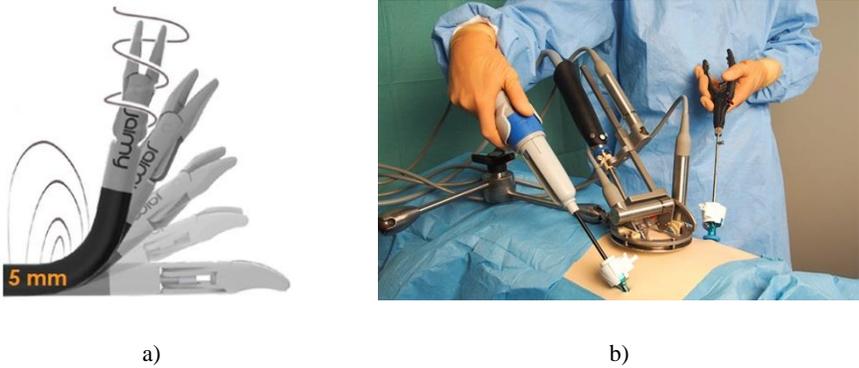


Figure 6. a) Bending, tip rotation and grasping are the 3DOF of JAIMY™, b) Laparoscopy surgery application, images taken from [2].

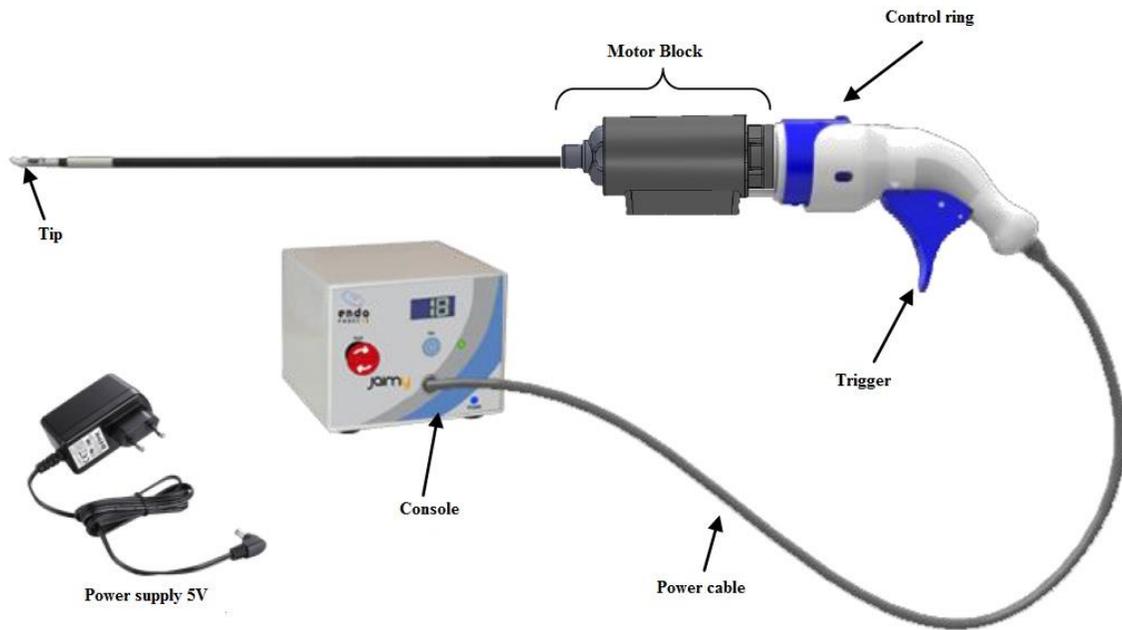


Figure 7. JAIMY™ general description [2].

Taking advantage of motorized joints, the objective is providing guidance for the surgeon to position the tip of the instrument. The idea is to select any point in the workspace around the tip and the system will automatically position the tip of the instrument in the desired point. Besides, using stereoscopic vision the surgeon will be able to know how far or close is the tip from the desired point. The scope of this present work is limited only to design the vision-based controller to position the tip of the JAIMY™. To do so, the kinematic model of JAIMY™ must be established. In the next chapters, a description of the platform, tools, kinematic model determination, system identification, camera calibration, vision algorithms and experimental validation will be presented.

Chapter 2 Modeling

The JAIMYTM instrument, as mentioned before, is a needle holder designed for suturing task. This task can be achieved thanks to the 3DOF provided by it, one of those is the flexion motion, and this is done by bending the end-effector from its straight position until it reaches 80 degrees with respect to the tool base line. It can be modeled as a geometric arc with a variable angle β depending of the curvature radius, the dimensions of the parameters I, L and $2r$ are known, see Figure 8. The next subsection describes the system kinematic model.

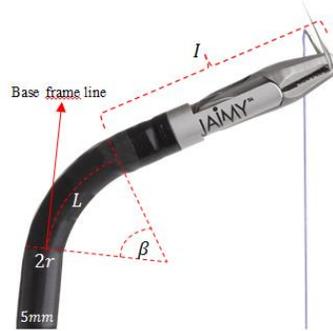


Figure 8. Flexible section of the JAIMYTM instrument.

2.1 Kinematic model

The objective of this section is determining the mathematical equation that relates the position of the tip with respect to the instrument base frame. The origin of the flexible part is considered as the system base frame. This point is represented as the intersection of $x - axis$, and $y - axis$ as shown Figure 9 below.

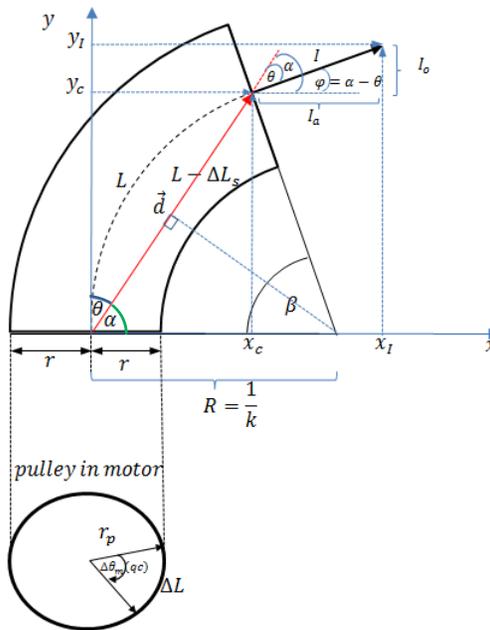


Figure 9. Kinematic model of the JAIMYTM instrument.

2.1.1 Direct kinematic model

The position of the end-effector i.e. tip of the instrument can be computed as:

$$x_I = x_c + I_a \quad (2.1)$$

$$y_I = y_c + I_o \quad (2.2)$$

where (x_I, y_I) represents the coordinates of the tip of the flexible part in the base frame.

The followings relations can be determined:

- $I \rightarrow$ length of the fixed part,
- $r \rightarrow$ radius of the flexible part,
- $k \rightarrow$ curvature of the radius,
- $\vec{d} \rightarrow$ vector norm of (x_c, y_c) ,
- $L \rightarrow$ length of the flexible part,
- The curvature radius can be computed as:

$$R = \frac{1}{k} \quad (2.3)$$

$$R = \frac{L}{\beta} \quad (2.4)$$

- $\beta \rightarrow$ arc angle formed by bending the flexible part,

$$\beta = kL \quad (2.5)$$

- $L_s \rightarrow$ border length of flexible part,

$$L_s = (R - \Delta R)\beta \quad (2.6)$$

$$L_s = \left(\frac{1}{k} - r\right)\beta \quad (2.7)$$

- $\Delta L_s \rightarrow$ length variation of border flexible part,

$$\Delta L_s = L - L_s \quad (2.8)$$

$$\Delta L_s = L - \left(\frac{1}{k} - r\right)\beta \quad (2.9)$$

$$\Delta L_s = L - \frac{kL}{k} + rkL \quad (2.10)$$

$$\Delta L_s = rkL \quad (2.11)$$

- ΔL_s considering the angle of motor θ_m and encoder qc ,

$$\Delta L_s = r_p \theta_m \quad (2.12)$$

- $r_p \rightarrow$ radius of pulley attached to motor,

$$r_p \theta_m = rkL \quad (2.13)$$

$$\frac{r_p \theta_m}{rL} = k \quad (2.14)$$

- $\theta \rightarrow$ vector (x_c, y_c) angle with respect to $y - axis$,

Flexion angle:

$$\alpha + \frac{\beta}{2} = \frac{\pi}{2} \quad (2.15)$$

$$\theta = \frac{\pi}{2} - \alpha \quad (2.16)$$

- Substituting 2.5 in 2.15 and 2.16,

$$\theta = \frac{kL}{2} \quad (2.17)$$

- Finally substituting 2.14 and 2.17,

$$\theta = \frac{r_p \theta_m}{2r} \quad (2.18)$$

- components (x_c, y_c) ,

$$\frac{d}{2} = R \cos(\alpha) \quad (2.19)$$

$$\frac{d}{2} = \frac{1}{k} \cos\left(\frac{\pi}{2} - \frac{kL}{2}\right) \quad (2.20)$$

$$\frac{d}{2} = \frac{1}{k} \cos\left(\frac{kL}{2} - \frac{\pi}{2}\right) \quad (2.21)$$

$$\frac{d}{2} = \frac{1}{k} \sin\left(\frac{kL}{2}\right) \quad (2.22)$$

$$d = \frac{2}{k} \sin\left(\frac{kL}{2}\right) \quad (2.23)$$

$$x_c = d \cos(\alpha) \quad (2.24)$$

$$y_c = d \sin(\alpha) \quad (2.25)$$

- components (x_c, y_c) in terms of θ ,

$$x_c = d \sin(\theta) \quad (2.26)$$

$$y_c = d \cos(\theta) \quad (2.27)$$

$$x_c = \frac{2}{k} \sin\left(\frac{kL}{2}\right) \sin\left(\frac{r_p \theta_m}{2r}\right) \quad (2.28)$$

$$y_c = \frac{2}{k} \sin\left(\frac{kL}{2}\right) \cos\left(\frac{r_p \theta_m}{2r}\right) \quad (2.29)$$

- removing absolute bars knowing this value will be always positive,

$$x_c = \frac{2rL}{r_p \theta_m} \sin\left(\frac{r_p \theta_m}{2r}\right) \sin\left(\frac{r_p \theta_m}{2r}\right) \quad (2.30)$$

$$y_c = \frac{2rL}{r_p \theta_m} \sin\left(\frac{r_p \theta_m}{2r}\right) \cos\left(\frac{r_p \theta_m}{2r}\right) \quad (2.31)$$

- position of end-effector, components (x_I, y_I) ,

$$I_a = I \cos \varphi \quad (2.32)$$

$$I_o = I \sin \varphi \quad (2.33)$$

- $\varphi \rightarrow$ angle formed by fixed part with respect to axis x ,

$$\varphi = \alpha - \theta \quad (2.34)$$

$$\varphi = \left(\frac{\pi}{2} - \frac{kL}{2}\right) - \frac{kL}{2} \quad (2.35)$$

$$\varphi = \frac{\pi}{2} - kL \quad (2.36)$$

$$\varphi = -\left(kL - \frac{\pi}{2}\right) \quad (2.37)$$

- $I_a \rightarrow$ adjacent side of tringle formed by fixed part,

$$I_a = I \cos\left(kL - \frac{\pi}{2}\right) \quad (2.38)$$

$$I_a = I \sin(kL) \quad (2.39)$$

- $I_o \rightarrow$ opposite side of tringle formed by fixed part

$$I_o = I \sin\left(\frac{\pi}{2} - kL\right) \quad (2.40)$$

$$I_o = I \cos(-kL) \quad (2.41)$$

$$I_o = I \cos(kL) \quad (2.42)$$

- components (x_I, y_I) ,

$$x_I = x_c + I \sin(kL) \quad (2.43)$$

$$y_I = y_c + I \cos(kL) \quad (2.44)$$

$$x_I = \frac{2rL}{r_p(\theta_m)} \left(\sin\left(\frac{r_p\theta_m}{2r}\right) \right)^2 + I \sin(kL) \quad (2.45)$$

$$y_I = \frac{2rL}{r_p(\theta_m)} \sin\left(\frac{r_p\theta_m}{2r}\right) \cos\left(\frac{r_p\theta_m}{2r}\right) + I \cos(kL) \quad (2.46)$$

- $I_o \rightarrow$ using the relation $\sin(2\sigma) = 2 \sin \sigma \cos \sigma$,

$$y_I = \frac{rL}{r_p\theta_m} \sin\left(\frac{r_p\theta_m}{2r}\right) + I \cos(kL) \quad (2.47)$$

- final equations for components (x_I, y_I) ,

$$x_I = \frac{2rL}{r_p\theta_m} \left(\sin\left(\frac{r_p\theta_m}{2r}\right) \right)^2 + I \sin\left(\frac{r_p\theta_m}{r}\right) \quad (2.48)$$

$$y_I = \frac{rL}{r_p\theta_m} \sin\left(\frac{r_p\theta_m}{r}\right) + I \cos\left(\frac{r_p\theta_m}{r}\right) \quad (2.49)$$

2.1.2 Jacobian computation

- From equations 2.49-2.50 and defining a and b for better comprehension as follows:

$$a = \frac{r_p}{r} \quad (2.50)$$

$$b = \frac{L}{a} \quad (2.51)$$

- therefore components (x_I, y_I) ,

$$x_I(\theta_m) = \frac{2b}{\theta_m} \left(\sin\left(\frac{a}{2}\theta_m\right) \right)^2 + I \sin(a\theta_m) \quad (2.52)$$

$$y_I(\theta_m) = \frac{b}{\theta_m} \sin(a\theta_m) + I \cos(a\theta_m) \quad (2.53)$$

- Defining the Jacobian as the mathematical expression that relates (x_I, y_I) and the velocity of the motors,

$$J = \begin{bmatrix} \frac{dx_I}{d\theta_m} \\ \frac{dy_I}{d\theta_m} \end{bmatrix} \frac{d\theta_m}{dt} \quad (2.54)$$

- The first component of J is:

$$\frac{dx_I}{d\theta_m} = \frac{d\left(\frac{2b}{\theta_m} \left(\sin\left(\frac{a}{2}\theta_m\right)\right)^2 + I \sin(a\theta_m)\right)}{d\theta_m} \quad (2.55)$$

$$\frac{dx_I}{d\theta_m} = \frac{d\left(\frac{2b}{\theta_m} \left(\sin\left(\frac{a}{2}\theta_m\right)\right)^2\right)}{d\theta_m} + \frac{d(I \sin(a\theta_m))}{d\theta_m} \quad (2.56)$$

- based on the product derivate and the function $\sin(\theta)$ derivative as in the equations 2.50-2.51,

$$\frac{d}{d\theta} \sin(\theta) = \cos(\theta), \quad (2.57)$$

$$(f \cdot g)' = f'g + fg' \quad (2.58)$$

- then,

$$\frac{dx_I}{d\theta_m} = \frac{d\left(\frac{2b}{\theta_m} \sin^2\left(\frac{a}{2}\theta_m\right)\right)}{d\theta_m} + Ia \cos(a\theta_m) \quad (2.59)$$

$$\frac{dx_I}{d\theta_m} = -\frac{2b}{\theta_m^2} \sin^2\left(\frac{a}{2}\theta_m\right) + 2\frac{a}{2} \sin\left(\frac{a}{2}\theta_m\right) \cos\left(\frac{a}{2}\theta_m\right) \frac{2b}{\theta_m} + Ia \cos(a\theta_m) \quad (2.60)$$

$$\frac{dx_I}{d\theta_m} = -\frac{2b}{\theta_m^2} \sin^2\left(\frac{a}{2}\theta_m\right) + \frac{2ab}{\theta_m} \sin\left(\frac{a}{2}\theta_m\right) \cos\left(\frac{a}{2}\theta_m\right) + Ia \cos(a\theta_m) \quad (2.61)$$

- using the relation $\sin(2\sigma) = 2 \sin \sigma \cos \sigma$,

$$\frac{dx_I}{d\theta_m} = -\frac{2b}{\theta_m^2} \sin^2\left(\frac{a}{2}\theta_m\right) + \frac{ab}{\theta_m} \sin(a\theta_m) + Ia \cos(a\theta_m) \quad (2.62)$$

- derivative of y_I ,

$$y_I(\theta_m) = \frac{b}{\theta_m} \sin(a\theta_m) + I \cos(a\theta_m) \quad (2.63)$$

$$\frac{dy_I}{d\theta_m} = \frac{d\left(\frac{b}{\theta_m} \sin(a\theta_m) + I \cos(a\theta_m)\right)}{d\theta_m} \quad (2.64)$$

$$\frac{dy_I}{d\theta_m} = \frac{d\left(\frac{b}{\theta_m} \sin(a\theta_m)\right)}{d\theta_m} + \frac{d(I \cos(a\theta_m))}{d\theta_m} \quad (2.65)$$

$$\frac{dy_I}{d\theta_m} = -\frac{b}{\theta_m^2} \sin(a\theta_m) + \frac{b}{\theta_m} a \cos(a\theta_m) - Ia \sin(a\theta_m) \quad (2.66)$$

- finally the Jacobian,

$$J = \begin{bmatrix} -\frac{2b}{\theta_m^2} \sin^2\left(\frac{a}{2}\theta_m\right) + \frac{ab}{\theta_m} \sin(a\theta_m) + Ia \cos(a\theta_m) \\ -\frac{b}{\theta_m^2} \sin(a\theta_m) + \frac{ab}{\theta_m} \cos(a\theta_m) - Ia \sin(a\theta_m) \end{bmatrix} \frac{d\theta_m}{dt} \quad (2.67)$$

2.2 DC motor Model

The flexion and rotation of JAIMY™ are driven by two DC motors. A DC motor model is a well-known model (see [11] for instance). Hence the motor model is represented in Figure 10.

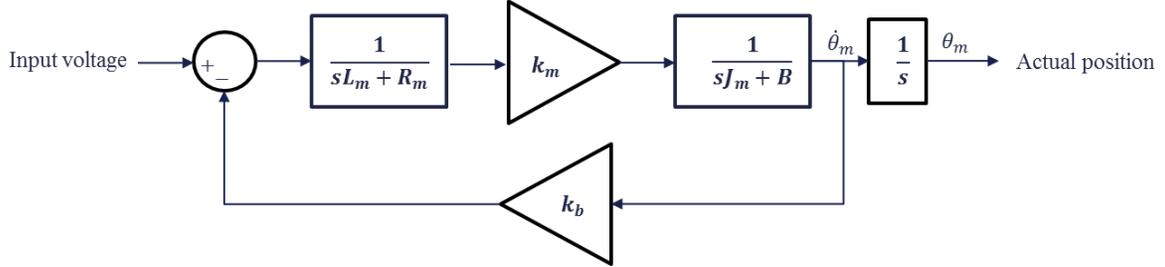


Figure 10. Mathematical model of a commercial DC Motor where $k_m = k_b$.

From Figure 10 the motor's parameters are defined as:

- $L_m \rightarrow$ Armature inductance,
- $R_m \rightarrow$ Armature resistance,
- $J_m \rightarrow$ Moment of inertia referred to the output shaft,
- $B \rightarrow$ Viscous-friction coefficient referred to the output shaft,
- $k_m \rightarrow$ Motor torque constant,
- $k_b \rightarrow$ Back EMF constant.
- $\dot{\theta}_m \rightarrow$ Motor velocity
- $\theta_m \rightarrow$ Motor position

The equation 2.68 represents the motor transfer function:

$$G(s) = \frac{1}{s} \left(\frac{k_m}{(sL_m + R_m)(sJ_m + B) + k_m k_b} \right) \quad (2.68)$$

2.3 Camera model

The camera model is also known, it is based on [13] and presented below:

$$s m' = A [r|t] M' \quad (2.69)$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.70)$$

Equations 2.69-2.70 are equivalent and relate the intrinsic parameters of the camera and images points with the 3D real world coordinates:

- $(X, Y, Z) \rightarrow$ are the coordinates of a 3D point in the real world,
- $(u, v) \rightarrow$ coordinates of projection point in pixels.
- $A \rightarrow$ camera matrix of intrinsic parameters,
- $(c_x, c_y) \rightarrow$ principal point,
- $f_x, f_y \rightarrow$ focal lengths expressed in pixel units,
- $r \rightarrow$ rotation matrix,
- $t \rightarrow$ translation matrix,
- $(k_1, k_2, p_1, p_2, k_3) \rightarrow$ Distortion coefficients are also needed.

2.4 Solution Model

In this section an image based visual servoing scheme is proposed, as shown in Figure 11. The system input is the desired position. The difference between the desired position and the current position is the system error to be minimized. This error is the controller input. The controller output (\dot{X}) is then multiply with the pseudoinverse of the Jacobian which has as input the current motor velocity ($\dot{\theta}_m$). The pseudoinverse of the Jacobian relates the velocity in real world coordinates space with the joint space. The visual feedback provides the actual position of the tip in the real world coordinates. The equations 2.71-2.72 represent the role of the pseudoinverse in the solution model.

$$\dot{X} = J(\theta_m)\dot{\theta}_m \quad (2.71)$$

$$\dot{\theta}_m = J(\theta_m)^+\dot{X} \quad (2.72)$$

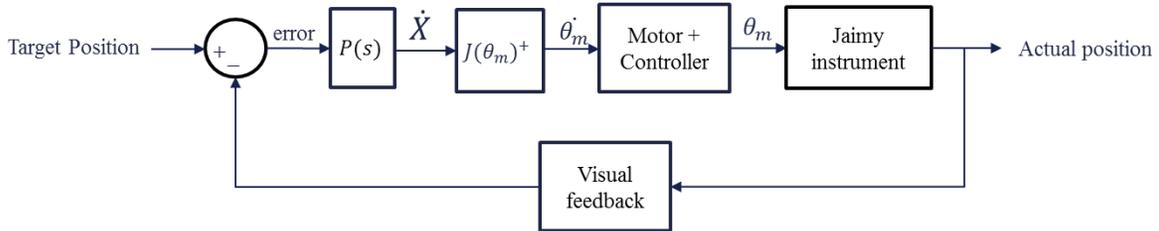


Figure 11. Image-based visual servoing of the JAIMY™.

Chapter 3 Simulation

This chapter describes simulations done using Matlab in order to validate the kinematic model, for this purpose an ideal motor model is used.

3.1 Simulation of kinematic model with ideal plant

In section 2.1 the kinematic model of the JAIMYTM instrument was determined. The equations 2.48-2.49 represent the position of the end-effector with respect to JAIMYTM base frame. In order to validate the kinematic model a closed loop configuration is implemented. The dc motor model is considered ideal. Figure 12 shows the structure of the closed loop. The simulation consisted in entering a desired angle of flexion in the left input block, the second block (Desired position) compute the components (x_d, y_d) of the desired position from a given angle. The Kinematic model block determines the components (x_a, y_a) of the actual position. A simple proportional gain is considered. The pseudoinverse of the Jacobian receives the controller output and the current angular position to finally feed the last block to implement the system animation. An oscilloscope has been added to visualize the input position angle versus the actual angle.

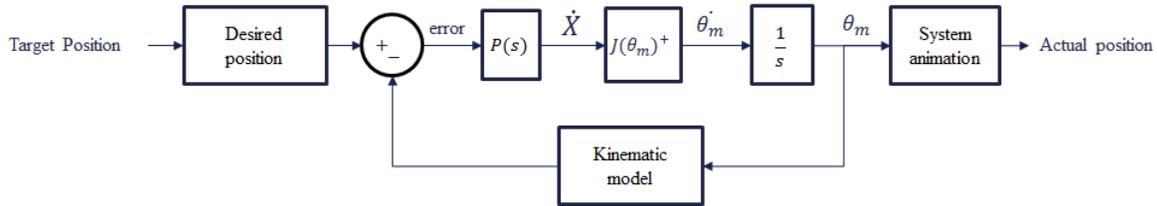


Figure 12. Closed loop to validate the kinematic model.

To verify if the simulation is correct the relation expressed in equation 2.18 is used. This equation represents half the angle of end-effector with respect to the axis y of instrument base frame; it is illustrated in Figure 9, so that equation can be expressed as follow:

$$\theta_l = 2\theta = \frac{r_p \theta_m}{r} \quad (3.1)$$

where θ_l is the angle of instrument tip.

Equation 3.1 is used to compute the expected angle of the tip after simulation. The result is presented in Figure 13, for this simulation the desired position angle entered is $\theta_m = 1.7452 \text{ rad}$, therefore based on 3.1 the expected angle is $\theta_l = \frac{0.1 \cdot 1.7453}{2.55} = 0.0684 \text{ rad}$, in the simulation the angle position expected is being reached after 0.4 seconds with 0% of error in steady state. The kinematic model has been successfully validated.

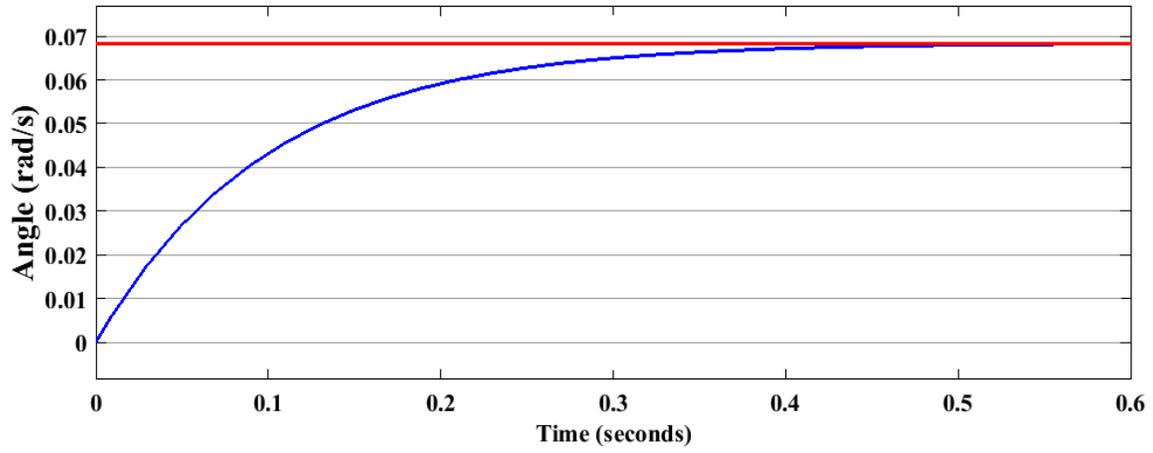


Figure 13. Kinematic model simulation result considering ideal dc motor model.

Chapter 4 Experimental Evaluation

The experimental evaluation of proposed control scheme is described in this chapter. First, a platform overview is described. Second, camera calibration stereo is detailed. Then, the software developed is introduced. Finally, system identification and experimental results are presented.

4.1 Platform Description

As already mentioned before, Figure 1 shows a general description of the proposed solution. It has four general components. The first one is the hand-held surgical instrument. Table 1 summarizes the technical characteristics of the JAIMYTM.

Size	620 mm
Weight	0.4 kg
Cable maximum length attached to the console	3 m
Motors	Faulhaber 1224A012SR-K188
Encoders	Faulhaber HEM3-256W
Reductor	Faulhaber 12/4 flexion reduction ratio 64:1 rotation reduction ratio 256:1
Protection	Non Immersible

Table 1. Technical characteristics of the JAIMYTM.

The **Controller Device** is the input to the surgical instrument, which is in charge of controlling the flexion-rotation motions of JAIMYTM motors. The hardware used to control the motors is EPOS3 from maxon motor. It will be described in section 4.1.1. Refer to [14] for further details. The **Image processing PC/Visualization** incorporates the **User Interface** (section 4.1.4) which implements the algorithms to process the images taken from the **Stereo Camera** (section 4.1.2), the communication software, motion commands and visualization.

4.1.1 Controller Device

The EPOS3 70/10 EtherCAT from maxon motor is used to implement the velocity controller for each motor of JAIMYTM. It has an EtherCAT interface which is connected to the PC by network cable. A velocity controller is already implemented in EPOS3, as shown in Figure 14. The PI controller of velocity is located at the input of another PI controller for the current to drive the motor.

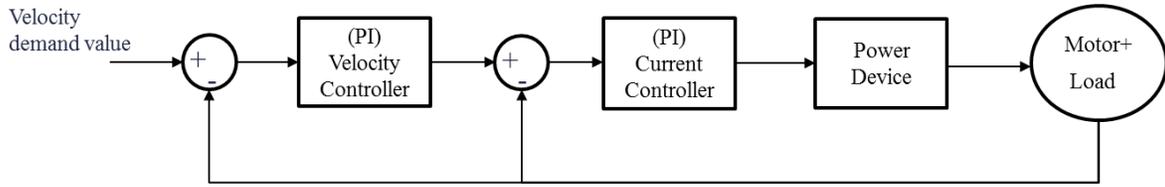


Figure 14. Controller architecture implemented by the EPOS3.

To determine the appropriate values for the parameters of each PI controller the auto tuning function of EPOS Studio software is used. This software is provided by maxon motor. Since the motors of JAIMY™ have load attached, the auto tuning give an appropriate response for the current controller but is not enough in the case of velocity controller. As a consequence, a manual tuning is required to get a better response for this case, Figure 15 shows the response for the current controller after auto tuning. It can be observed that the current actual value in red is following to the current demand value in blue and it is under the maximal admissible current 440mA as expected, see [15].

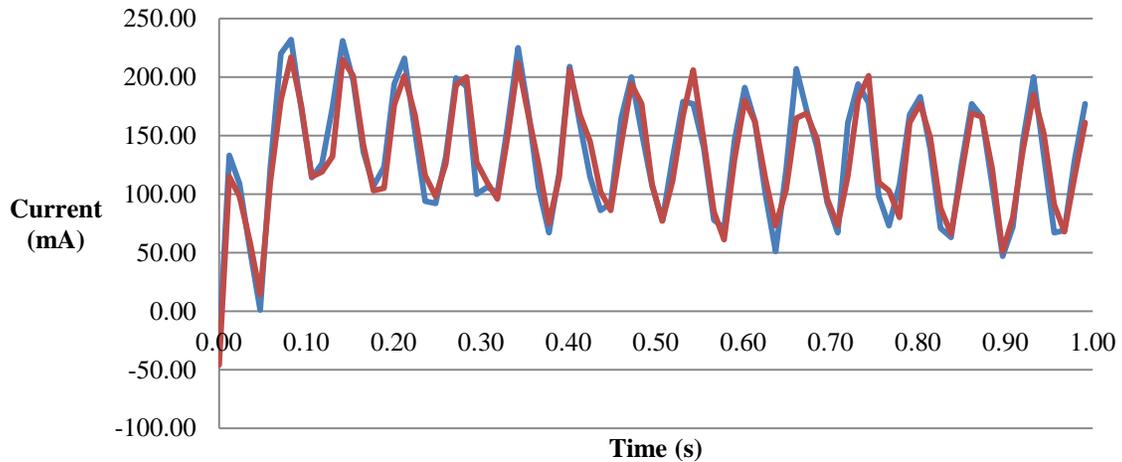


Figure 15. Current controller response after auto tuning, blue → current demand value, red → current actual value.

Figure 16 shows the velocity controller response after manual tuning, from this data the velocity demand value is 3000 rpm, the arithmetic average of the velocity actual value is 3000.84 rpm with a standard deviation of 10.48 rpm between 0.20 and 1 second in steady state, for the purpose of this project this response is acceptable.

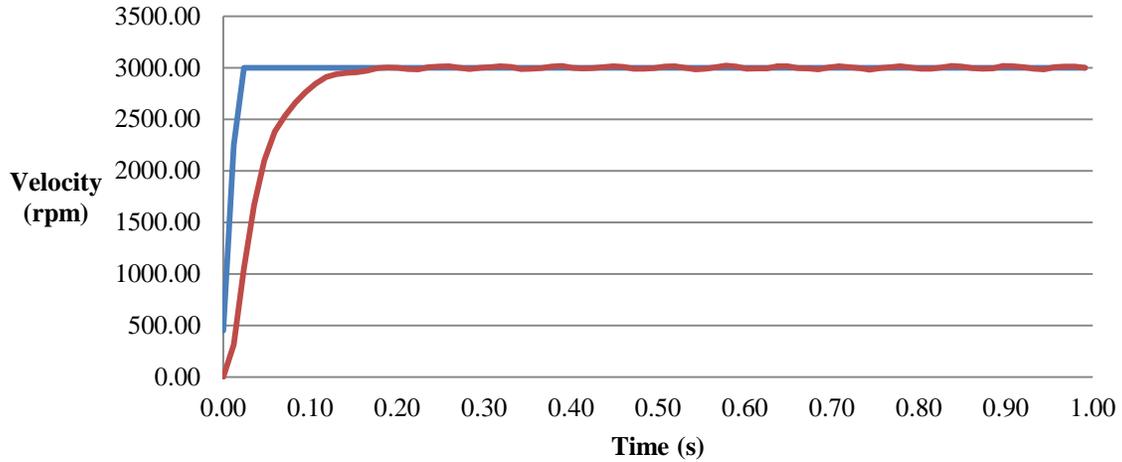


Figure 16. Velocity controller response after manual tuning, blue → velocity demand value, red → velocity actual value.

4.1.2 Camera Stereo Calibration

The camera stereo calibration is done based on the algorithms described in the chapter 12 of [16]. 25 pair images of a chessboard of 54 internal corners with a square size of 7.5mm are used. Figure 17 shows one of the image pairs used to do the stereo calibration. The cameras used are two scA640-70fc by Basler AG [17] plus two endoscopic lenses attached to them. These lenses are from Karl Storz. Figure 18 displays the whole setup of endoscope, cameras and JAIMY™.

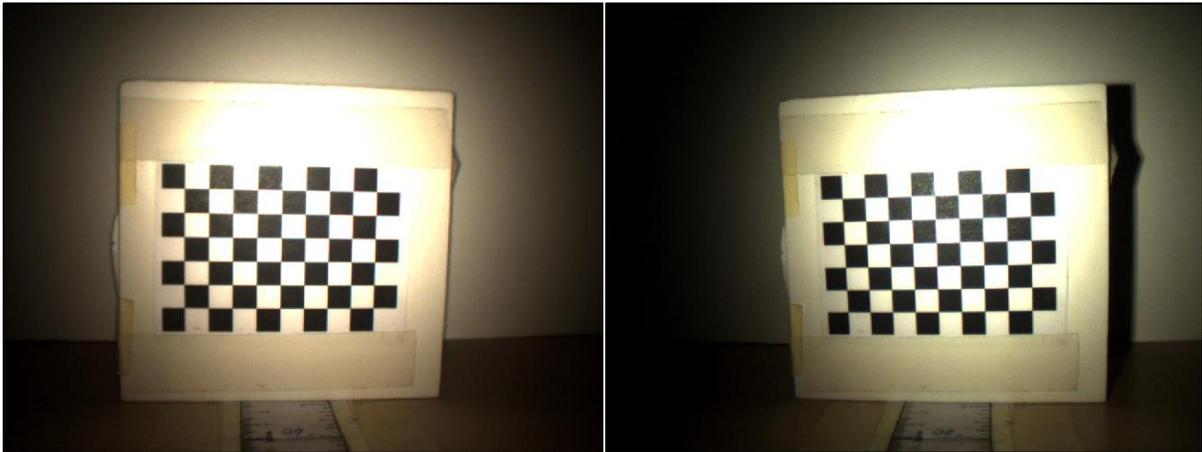


Figure 17. First pair image of the pictures used to do the stereo calibration.

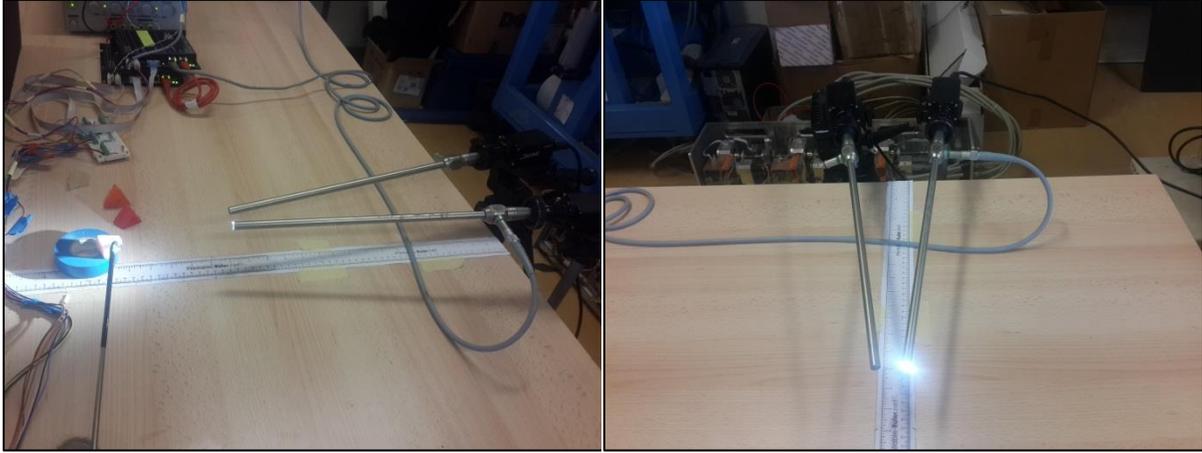


Figure 18. Endoscopic lenses plus Basler cameras setup.

4.1.3 System Identification

Equations 2.48-2.49 represent the position of the tip with respect to JAIMYTM base frame. From those equations, r , L and I are known. As mentioned before in section 3.1 the pulley (r_p) radius is missing therefore an identification procedure is needed. In order to identify the missing parameter, the equation 3.1 is used. After the stereo camera system is calibrated and the tracking algorithms are implemented, experimental values θ_I are taken by using the stereo vision feedback, these values are presented in Table 2. Figure 19 shows the tip of JAIMYTM being tracked by using green markers. The points P_1 , P_2 , P_3 and P_4 are located strategically in order to make easier tracking the object, the vector $\vec{A} = \overline{P_1P_2}$ is used to define JAIMYTM base frame, whereas $\vec{B} = \overline{P_3P_4}$ represent the instrument rigid section. The point P_2 is considered as the origin of JAIMYTM base frame and the beginning of flexible section, and P_3 the end of this section. Since the stereo camera system is already calibrated, it is possible to get the real 3D information of those points. Accordingly the real angle between the vectors \vec{A} and \vec{B} can be measured, this angle is θ_I .

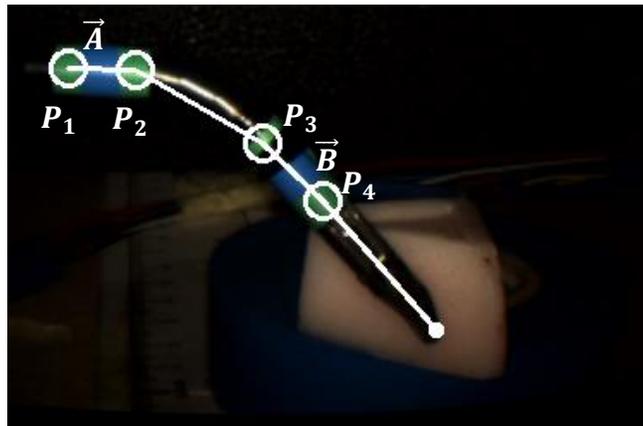


Figure 19. Image of the tip of the JAIMYTM being tracked by using markers.

It can be observed from equation 3.1 that the angle of instrument tip (θ_I) is linearly related with the motor angle (θ_m), being θ_I the product of θ_m multiplied by the constant factor $\frac{r_p}{r}$. Therefore this expression is in the form of a linear equation as shown below:

$$y = mx + b \quad (4.1)$$

Where y is θ_I which is a function of θ_m , m is the slope represented by $\frac{r_p}{r}$, x represents θ_m and $b = 0$ for this case, see the equation 4.2 presented below.

$$\theta_I = \frac{r_p}{r} \theta_m \quad (4.2)$$

The pulley radius was calculated using the Matlab optimization tool for minimizing the equation 4.3 with the experimental data collected in Table 2.

$$e = \sum_{j=1}^n \left(\frac{r_p \theta_{m_j}}{r} - \theta_{I_j} \right)^2 \quad (4.3)$$

$\theta_{m_j}(\text{rad})$	$\theta_{I_j}(\text{rad})$
0.35	0.03
3.44	0.07
6.55	0.21
9.66	0.27
12.77	0.39
15.95	0.50
19.06	0.61
22.18	0.70
25.30	0.76
28.41	0.87
31.53	0.94
34.64	1.05
37.75	1.16
40.82	1.34
43.94	1.43

Table 2. Experimental values of θ_{m_j} , θ_{I_j} .

Table 2 presents the collected data for the optimization procedure in order to estimate the pulley radius. The variable θ_{m_j} is the experimental value of motor angle, θ_{I_j} is the tip angle experimental value and e is the squared error function to be minimized. The value of θ_{I_j} and θ_{m_j} for each iteration j are measured for tip different positions. Fifteen measurements are taken, then entering the function 4.3 and its experimental values of θ_{m_n} , θ_{I_n} in the optimization tool of Matlab, the value of r_p for the minimum e is determined, thus $r_p =$

0.078 mm and $e = 0.0149 \text{ (rad)}^2$. The method of linear regression is also used in order to verify the previously estimated value. For this case the Matlab Curve Fitting Tool is used.

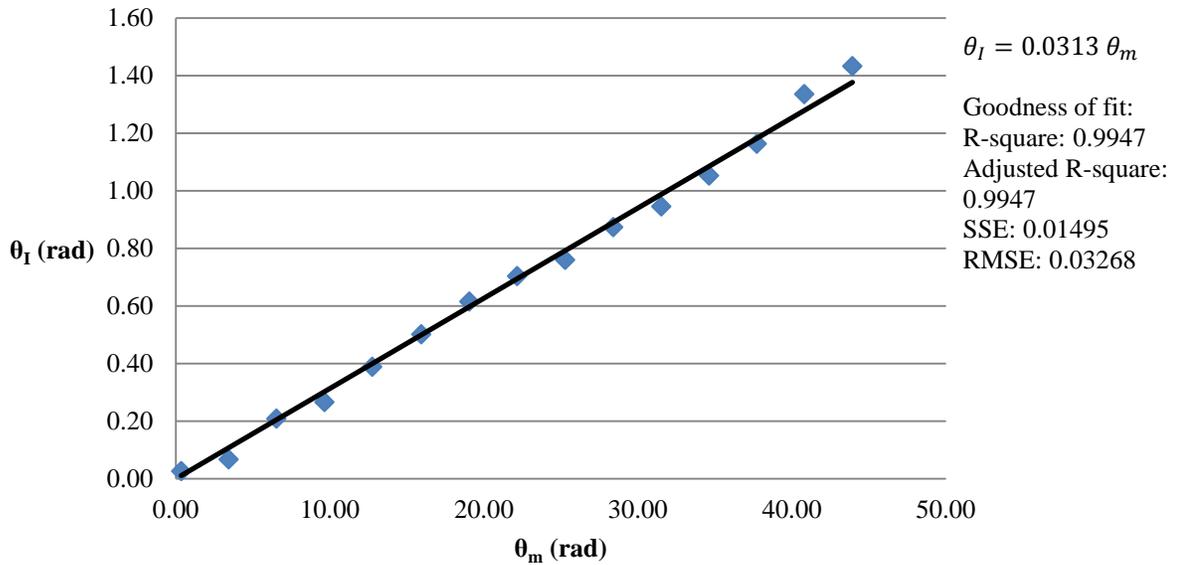


Figure 20. Linear regression method to estimate the radius of the pulley.

Figure 20 shows the result of optimization using the linear regression method. The resulting slope after optimization is $m = 0.0313$, it was mentioned before that this data represents the factor $\frac{r_p}{r}$, and knowing that $r = 2.5 \text{ mm}$ then $r_p = 0,078 \text{ mm}$. It is important to mention that the parameters m and b in the Matlab Curve Fitting Tool were limited to be equal or greater than 0, since they cannot be negative for the real system. The adjusted R_{square} and R_{square} are equal to 0.9947, it means that the estimated model explains the 99,47% of the variation in the experimental data, the values for SSE (sum of squares due to error) and RMSE (Root Mean Squared Error) are 0.0149 rad^2 and 0.03268 rad correspondingly. RMSE represents the standard deviation. For the estimated value of r_p , the same value was obtained and verified from both optimization tools. Therefore is possible to predict values of θ_l based on θ_m with 99,47% of goodness with a standard deviation of 0.03268 rad .

4.1.4 Software

Image processing, visualization and user interaction are done by software running in a PC workstation HP Z400. This software implements all the algorithms to process the images to track the JAIMY™ tip and determining its 3D location with respect to the point of interest. It also performs the EtherCAT communication with the EPOS3, and provides the user interface to control the motion of JAIMY™ motors. The software is developed using QT libraries [18] for the user interface, OpenCV [19] for image processing and Debian 8 [20] as operating system and the EhterCAT drivers for linux provided by [21] to communicate with EPOS3. The Figure 21 below describes the user interface developed to control the system.

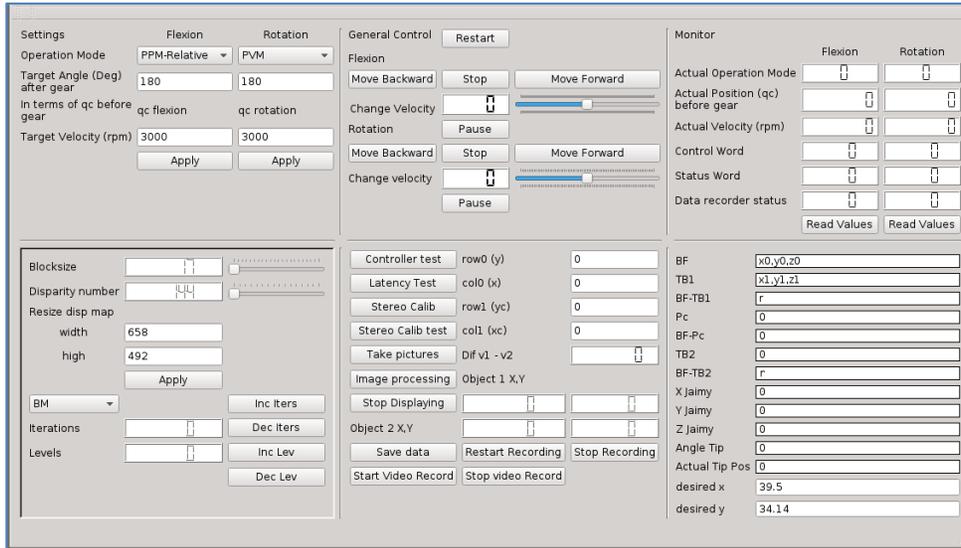


Figure 21. User interface developed to control the system.

Figure 22 shows the block diagram of the solution including the hardware and software. The software flow diagram starts after receiving the image information from the cameras via Firewire. The user makes the segmentation of the green markers manually; the user interface provides this functionality. After the segmentation is done, the BGR image is converted to HSV representation, then the hue histogram of each marker is extracted and filtered by back projection [22]. The Camshift algorithm [23] is used to track the markers using the histogram and the back projection image. Since the stereo calibration is already done, the intrinsic and extrinsic parameters of the camera are known, therefore the relation between the image plane and the real world coordinates is also known. Using the parameters of the camera and the block matching algorithm [24] to find stereo correspondences the disparity map is computed. From the disparity map and the relation 4.4 (also refer to [13]) the 3D map information is also calculated. Once having those information maps that relates the image pixels and the 3D real world coordinates, the location of each pixel of the image with disparity is known with respect to the camera base frame, it includes the location of the markers and any other point which disparity information has been found. It is important to mention that the algorithms used to find stereo correspondences and 3D reconstruction are implemented by using OpenCV libraries, in this case the CUDA support is used to make the image processing faster. Mathematical computations block in Figure 22 uses the disparity/3D map information, the location of the markers and the point of interest to determine the real position of the point of interest with respect to JAIMYTM base frame. Besides to implementing the pseudoinverse of the Jacobian to compute the velocity demand value ($\dot{\theta}_m$), this demand velocity value is sent by the EPOS3 commands library through the linux based EtherCAT drivers, it is important to mention that the software is also reading the position actual value (θ_m) from the EPOS3 via EtherCAT communication.

$$[X \ Y \ Z \ W]^T = Q[x \ y \ disparity(x,y) \ 1]^T \quad (4.4)$$

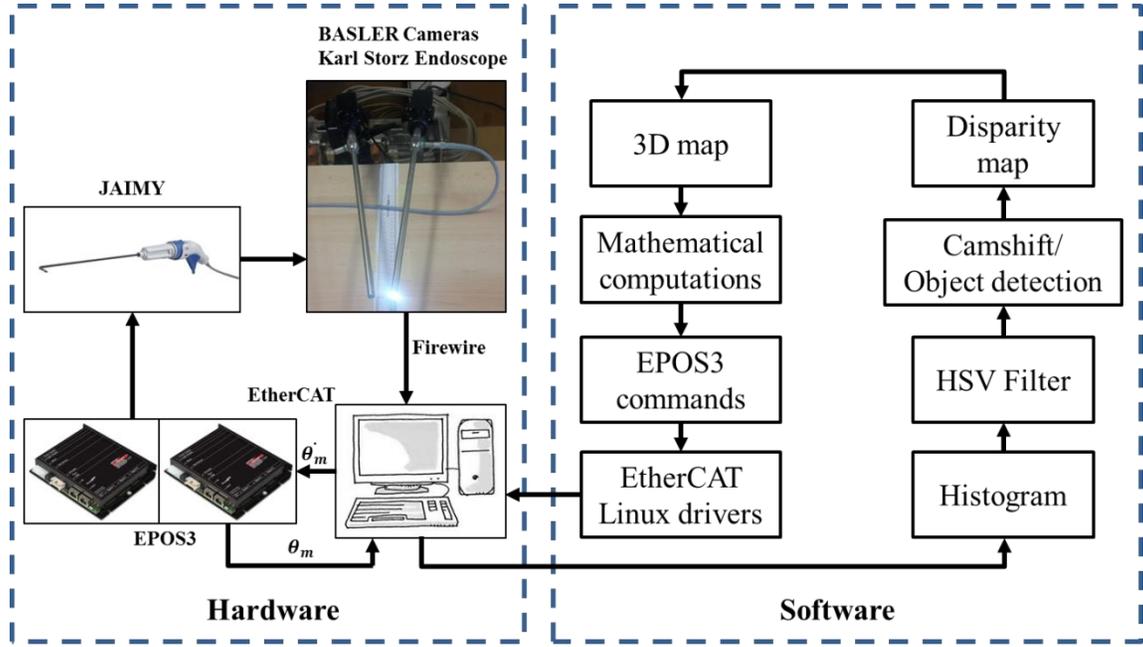


Figure 22. Block diagram of the solution.

Another important task done by the Mathematical computations block is calculating the relation between the JAIMYTM base frame and the camera's frame, this is done by defining the vectors \vec{P}_2 , \vec{A} and \vec{B} from Figure 19 as:

$$\vec{P}_2 = P_{2x} + P_{2y} + P_{2z} \quad (4.5)$$

$$\vec{A} = \vec{P}_1 \vec{P}_2 \quad (4.6)$$

$$\vec{B} = \vec{P}_3 \vec{P}_4 \quad (4.7)$$

Considering unit vector of \vec{A} as the axis x of JAIMYTM base frame:

$$\vec{x}_u = \frac{\vec{A}}{\|\vec{A}\|} = x_{u_x} \hat{x} + x_{u_y} \hat{y} + x_{u_z} \hat{z} \quad (4.8)$$

Unit vector of axis z of JAIMYTM base frame:

$$\vec{z}_u = \frac{\vec{A} \times \vec{B}}{\|\vec{A} \times \vec{B}\|} = z_{u_x} \hat{x} + z_{u_y} \hat{y} + z_{u_z} \hat{z} \quad (4.9)$$

Unit vector of axis y of JAIMYTM base frame:

$$\vec{y}_u = \frac{(\vec{A} \times \vec{B}) \times \vec{A}}{\|(\vec{A} \times \vec{B}) \times \vec{A}\|} = y_{u_x} \hat{x} + y_{u_y} \hat{y} + y_{u_z} \hat{z} \quad (4.10)$$

Rotation-Translation matrix that relates the JAIMYTM base frame with the cameras base frame:

$$P_c = [RT]P_j \quad (4.11)$$

$$P_j = [RT]^{-1}P_c \quad (4.12)$$

$$RT = \begin{bmatrix} x_{u_x} & y_{u_x} & z_{u_x} & 0 \\ x_{u_y} & y_{u_y} & z_{u_y} & 0 \\ x_{u_z} & y_{u_z} & z_{u_z} & 0 \\ P_{2x} & P_{2y} & P_{2y} & 1 \end{bmatrix} \quad (4.13)$$

where P_c is any 3D real world point in the camera base frame, P_j is any 3D real world point in the JAIMYTM base frame, $[RT]$ the rotation-translation matrix that relates both base frames and $[RT]^{-1}$ the inverse of the rotation-translation matrix, \vec{P}_2 is the location of JAIMYTM base frame origin with respect to the camera base frame.

4.2 Experimental Results

This section describes the experimental results obtained after implementing the solution proposed in page 1. The kinematic model was determined in chapter 1 in order to implement the Jacobian which is the element that relates the real world coordinates space with the joint space. The missing parameter, pulley radius r_p was experimentally identified in section 4.2. The velocity controller device used and its PI controller explained in section 4.1.1, the camera stereo calibration and the user interface were also presented in chapter 4. This section presents functional results after putting together the solution's components previously mentioned.

The system functionality consists in choosing a point in the workspace of the instrument by using the visual feedback, then the system will pose the tip of the JAIMYTM in the desired position, Figure 23 shows the experiments when the system is positioning the tip of the instrument in the desired point, from the picture the error for the component x is -0.857 mm and the error for component y is 0.0889 mm , error less than 1 mm in both axis, all the values are in mm, the final velocity is 0 rpm, which means that the system has reached the steady state.

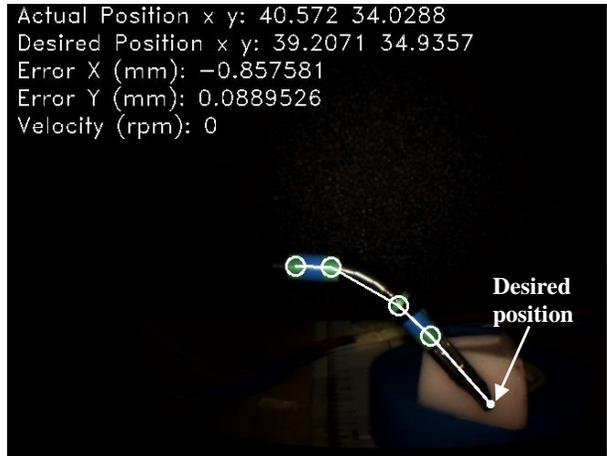


Figure 23. Automatic positioning of the tip of the JAIMY™ in the desired point.

Figure 24 shows the position response of the system for a given desired point, for this experiment the desired point is $(x_d, y_d) = (39.50, 34.14)$, the final position reached in steady state is $(x_a, y_a) = (39.27, 34.50)$, the error for the component x is 0.23 mm and the error for the component y is -0.36 mm , error less than 1 mm in both axis. It is important to mention that this response is being affected by the noise caused for the variation of the image amount frames, improving the image processing and tracking algorithm this noise can be reduced.

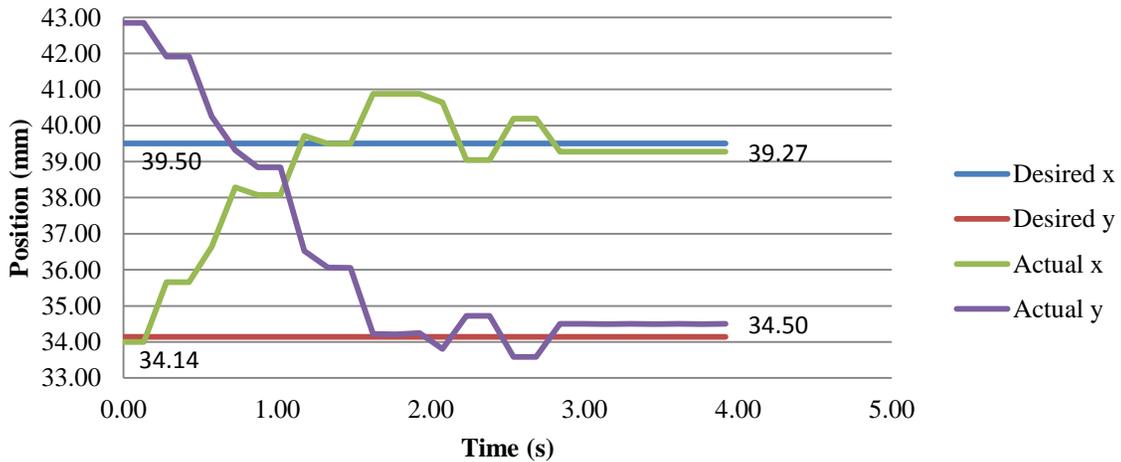


Figure 24. Position response of the system.

Figure 25 displays the velocity response for the same experiment in Figure 23, this graphic shows in blue the velocity demand value and the actual value in red. The velocity actual value is following the velocity demand value as expected. The velocity is increased in order to reach the desired position and then decreased as the error is being reduced.

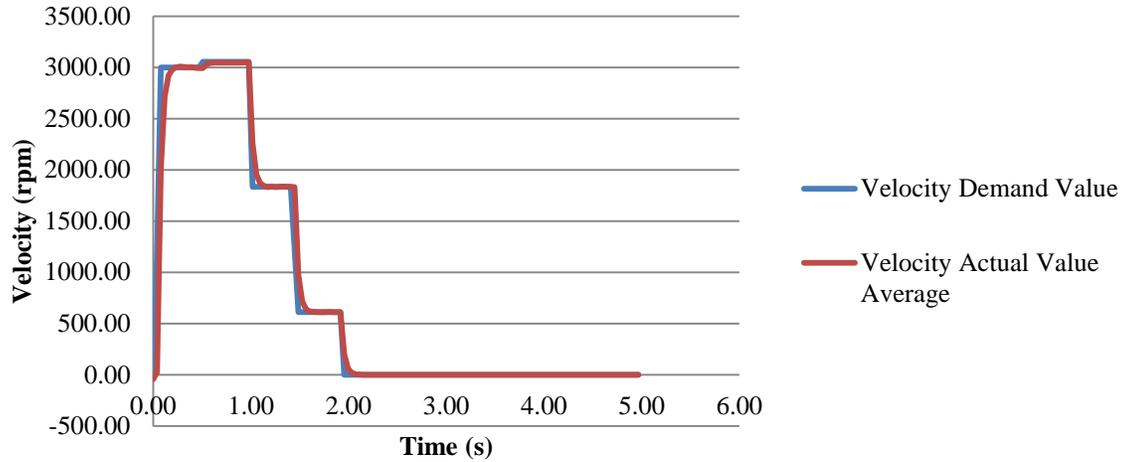


Figure 25. Velocity response of the system.

Figure 26 presents the disparity map and 3D information for different block sizes of the same image. This information is used by the software that implements the tracking and visual control algorithms. The disparity map represents the θ_m differences between the stereoscopic images, then by reprojecting the disparity map to 3D space the 3D map information is obtained, images b) and d) are the disparity maps, images c) and e) the corresponding 3D map information. It can be observed that the smaller is the block size the more difficult is to find the disparities and more details can be observed, therefore the higher is the block size the easier is to find the disparities and less details can be observed. Hence a trade-off between the detail of the image and the smoothness of the disparity must be established. For this project, it is experimentally determined that a block size between 17 and 29 works properly.

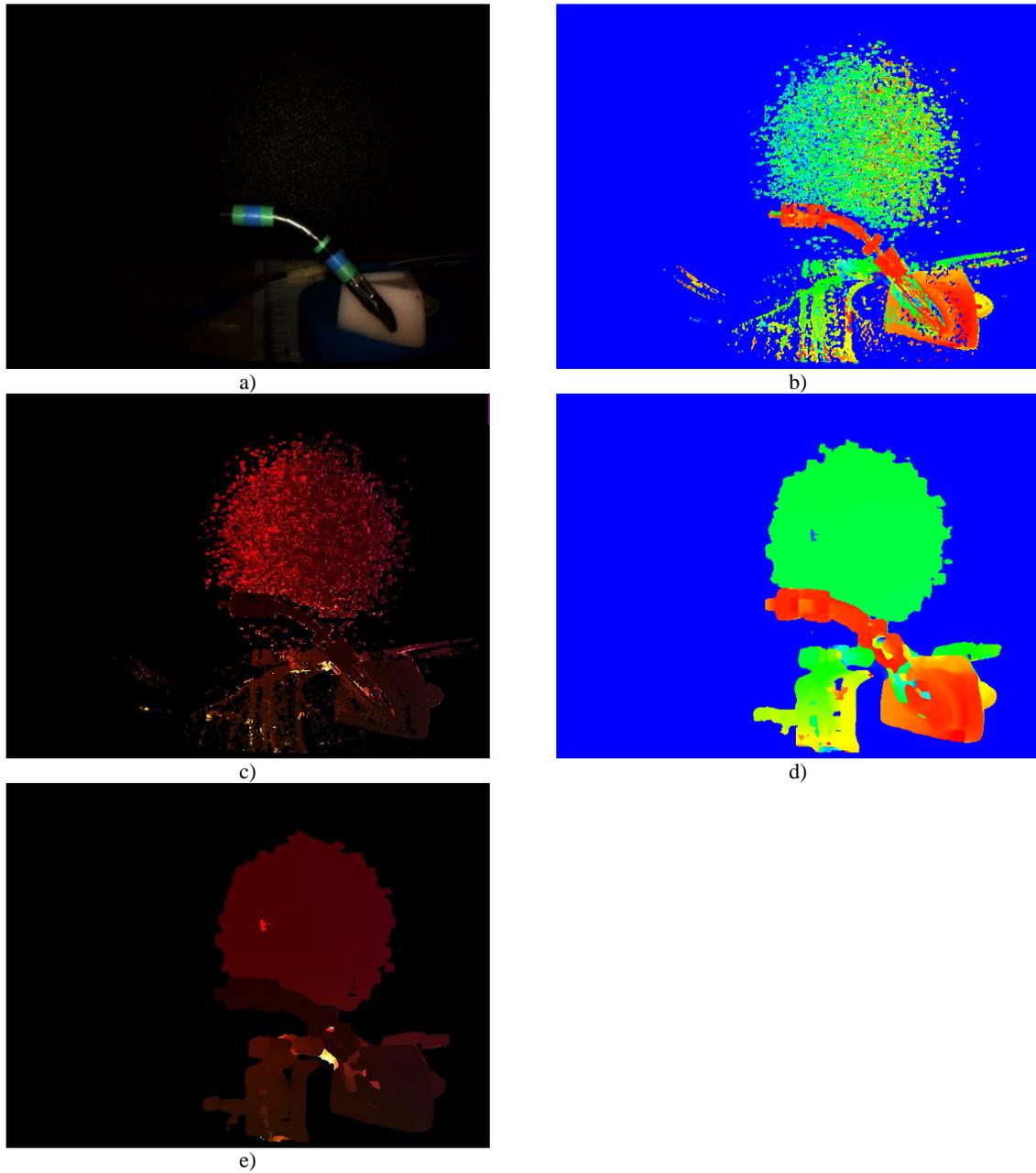


Figure 26. Disparity map and 3D information for different blocksize of disparity, a) Original image, b) Disparity map blocksize = 3, c) 3D map blocksize = 3, d) Disparity map blocksize = 17, e) 3D map blocksize = 17.

Conclusions and Perspectives

In this thesis, the development of a system with the capability of assisting surgeons to pose the end-effector of the JAIMYTM instrument is presented as the main objective; it is done by using stereoscopic vision feedback. As part of the work flow to reach the general objective, specific objectives in Introduction section are proposed, concluding the following:

- For the specific objective of determining the kinematic model of the instrument, it is determined and validated in a closed loop simulation; refer to sections 3.1-3.2.
- Missing parameter radius of the pulley identification is successfully estimated.
- For the objective of using stereoscopic vision as feedback to estimate the actual position of the instrument and the point of interest, it is developed according to Figure 11, with a unit proportional gain.
- The position controller of the system provides a response with an error less than 1 mm in steady state based on the data presented in Figure 23 and Figure 24.

The general and specific objectives have been reached during the project development, the probe of concept of its functionality has been successfully validated, however there is more work to do in order to improve the system and make it feasible in real surgical environment, it is translated as next steps for futures works, such as the implementation of a profile to reach the desired position to help avoiding high velocity steps at the moment of positioning the tip of the instrument, besides of improving the controller position response by adjusting the controller parameters instead of using only a unit proportional gain, also knowing the camera parameters is possible to add augmented reality to provide more information to be displayed to improve assistance during the surgical procedure. Another important improvement would be taking advantage of the new JAIMYTM's version with bidirectional flexion motion, also improving the algorithms to track the instrument and 3D reconstruction can reduce the noise affecting the position controller, a measurement of the accuracy of the system is also needed with the appropriate instrumentation since the scope of this stage of the project was focus on functional results not in accuracy.

The project scope and recommendation has been presented in order to improve the system, others considerations would come out according to the work to be continued in the future.

Bibliography

- [1] LIRMM, “DEXTER / Teams / Research / Homepage / Lirmm.fr / - LIRMM,” *www.lirmm.fr*, 2014. [Online]. Available: http://www.lirmm.fr/lirmm_eng/research/teams/dexter. [Accessed: 23-Nov-2015].
- [2] Endocontrol, “JAIMY™ - Endocontrol,” 2015. [Online]. Available: <http://www.endocontrol-medical.com/produits/jaimy-oui/>. [Accessed: 24-Nov-2015].
- [3] F. Tendick, S. S. Sastry, R. S. Fearing, and M. Cohn, “Applications of micromechatronics in minimally invasive surgery,” *IEEE/ASME Trans. Mechatronics*, vol. 3, no. 1, pp. 34–42, Mar. 1998.
- [4] A. Krupa, J. Gangloff, C. Doignon, M. F. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux, “Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing,” *IEEE Trans. Robot. Autom.*, vol. 19, no. 5, pp. 842–853, Oct. 2003.
- [5] P. Hynes, G. I. Dodds, and A. J. Wilkinson, “Uncalibrated Visual-Servoing of a Dual-Arm Robot for MIS Suturing,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, 2006, pp. 420–425.
- [6] B. C. Becker, C. R. Valdivieso, J. Biswas, L. A. Lobes, and C. N. Riviere, “Active guidance for laser retinal surgery with a handheld instrument.,” *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2009, pp. 5587–90, Jan. 2009.
- [7] B. C. Becker, R. A. Maclachlan, L. A. Lobes, G. D. Hager, and C. N. Riviere, “Vision-Based Control of a Handheld Surgical Micromanipulator with Virtual Fixtures.,” *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 674–683, Feb. 2013.
- [8] A. Agustinos, R. Wolf, J. A. Long, P. Cinquin, and S. Voros, “Visual servoing of a robotic endoscope holder based on surgical instrument tracking,” in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2014, pp. 13–18.
- [9] R. Reilink, S. Stramigioli, and S. Misra, “3D position estimation of flexible instruments: marker-less and marker-based methods.,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 8, no. 3, pp. 407–17, May 2013.
- [10] P. Cabras, D. Goyard, F. Nageotte, P. Zanne, and C. Doignon, “Comparison of methods for estimating the position of actuated instruments in flexible endoscopic surgery,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3522–3528.
- [11] Katsuhiko Ogata, *Modern Control Engineering, 5/E*, 5th ed. Pearson, 2010.
- [12] F. Monasterio-Huelin and Á. Gutiérrez, “Modelo lineal de un motor de corriente continua.,” *Robolabo*, 2012. [Online]. Available: http://www.robolabo.etsit.upm.es/asignaturas/seco/apuntes/motor_dc.pdf. [Accessed: 01-Dec-2015].
- [13] OpenCV, “Camera Calibration and 3D Reconstruction — OpenCV 2.4.12.0 documentation.” [Online]. Available: [http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=3d reconstruction#cv2.reprojectImageTo3D](http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=3d%20reconstruction#cv2.reprojectImageTo3D). [Accessed: 02-Dec-2015].

- [14] “maxon motor - Online Shop.” [Online]. Available: http://www.maxonmotor.com/maxon/view/product/control/Positionierung/411146?etcc_med=1column&etcc_cmp=comp_00003AC8&etcc_cu=onsite&etcc_var=%5bcm%5d%23en%23_d_&etcc_plc=product_overview_controller_epos. [Accessed: 04-Dec-2015].
- [15] “FAULHABER.” [Online]. Available: https://fmcc.faulhaber.com/details/overview/PGR_1049_13818/PGR_13818_13813/en/GLOBAL/#. [Accessed: 04-Dec-2015].
- [16] G. Bradski and A. Kaehler, *Learning OpenCV - O’Reilly Media*, First Edit. O’Reilly Media, 2008.
- [17] “Basler scout sca640-70fc - Area Scan Camera.” [Online]. Available: <http://www.baslerweb.com/en/products/area-scan-cameras/scout/sca640-70fc>. [Accessed: 07-Dec-2015].
- [18] “Qt.” [Online]. Available: <https://www.qt.io/>. [Accessed: 07-Dec-2015].
- [19] “OpenCV | OpenCV.” [Online]. Available: <http://opencv.org/>. [Accessed: 07-Dec-2015].
- [20] “Debian -- El sistema operativo universal.” [Online]. Available: <https://www.debian.org/>. [Accessed: 07-Dec-2015].
- [21] “IgH EtherCAT Master for Linux.” [Online]. Available: <http://www.etherlab.org/en/ethercat/index.php>. [Accessed: 07-Dec-2015].
- [22] “Back Projection — OpenCV 2.4.12.0 documentation.” [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/back_projection/back_projection.html. [Accessed: 09-Dec-2015].
- [23] “OpenCV: Meanshift and Camshift.” [Online]. Available: http://docs.opencv.org/master/db/df8/tutorial_py_meanshift.html#gsc.tab=0. [Accessed: 09-Dec-2015].
- [24] “OpenCV: cv::cuda::StereoBM Class Reference.” [Online]. Available: http://docs.opencv.org/master/db/d8a/classcv_1_1cuda_1_1StereoBM.html#gsc.tab=0. [Accessed: 09-Dec-2015].
- [25] Maxon motor, “EPOS3 EtherCAT Position Controllers, Firmware Specification,” 2014. [Online]. Available: http://www.maxonmotor.com/medias/sys_master/root/8815100592158/EPOS3-EtherCAT-Firmware-Specification-En.pdf. [Accessed: 11-Dec-2015].
- [26] “IgH EtherCAT Master 1.52 Documentation.” [Online]. Available: <http://www.etherlab.org/download/ethercat/ethercat-1.5.2.pdf>. [Accessed: 17-Dec-2015].
- [27] “Quadro 4000 – Workstation graphics card for 3D design, styling, visualization, CAD, and more.” [Online]. Available: <http://www.nvidia.com/object/product-quadro-4000-us.html>. [Accessed: 16-Dec-2015].
- [28] Endocontrol, *JAiMY - SPY Manuel Utilisateur*. EndoControl, 2013.
- [29] Maxon motor, “EPOS3 70/10 EtherCAT Positioning Controller Hardware Reference,” 2014. [Online]. Available: http://www.maxonmotor.es/medias/sys_master/root/8815047180318/411146-Hardware-Reference-En.pdf. [Accessed: 17-Dec-2015].

Command	LowByte of Controlword [binary]	State Transition
Shutdown	0xxx x110	2, 6, 8
Switch On	0xxx x111	11 3
Switch On & Enable Operation	0xxx 1111	3, 4 ^{*1)}
Disable Voltage	0xxx xx0x	7, 9, 10, 12
Quickstop	0xxx x01x	7, 10, 11
Disable Operation	0xxx 0111	5
Enable Operation	0xxx 1111	4, 16
Fault reset	0xxx xxxx → 1xxx xxxx	15

Remark:

*1) Automatic transition to “Enable Operation” after executing “Switched On” functionality.

Table 3. Device control commands, taken from [25].

A.3 Software components

- **Debian**
Operating system
<https://www.debian.org/>
- **Camwire**
Generic Firewire camera communication in linux
<http://kauri.auck.irl.cri.nz/~johanns/camwire/>
- **Ethercat version: 1.52**
Linux driver
<http://www.etherlab.org/en/ethercat/index.php>

Suggestion of configuration for Ethercat driver installation:

```
./configure --with-linux-dir=/usr/src/linux --enable-8139too=no --enable-cycles=yes --with-e1000e-kernel=yes
```

Check if the NIC is compatible with the Ethercat driver, look for the linux kernel that support your hardware:

<http://www.etherlab.org/en/ethercat/hardware.php>

Refer to [26] for further details about installation.

- **QT Libraries**
 - Download and install QtCreator.
<https://www.qt.io/download-open-source/>
 - Install opengl libraries for qt
\$sudo apt-get install libqt5opengl5-dev
- **OpenCV**
<http://opencv.org/>
- **CUDA Toolkit**
Download nvidia drivers and CUDA toolkit for the graphic card to be used
<https://developer.nvidia.com/cuda-toolkit-65>

A.4 Graphic card information

GPU Specs:	
NVIDIA Quadro GPU	Quadro 4000
CUDA Cores	256
Form Factor	4.376" H x 9.50" L / Single Slot
Gigaflops (Single Precision)	486.4
Gigaflops (Double Precision)	243.2
GPU Memory Specs:	
Total Frame Buffer	2 GB GDDR5
Memory Interface	256-bit
Memory Bandwidth (GB/sec)	89.6 GB/s
Software	
Linux NVIDIA Driver versión	340.52
Cuda Toolkit version	6.5

Table 4. Graphic card specifications used in the project, refer to [27].

A.5 Connections

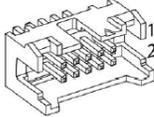
JAIMY™, Sub-d connector 15 wires 	1: + flexion motor 2: - flexion motor 3: channel A flexion encoder 4: channel B flexion encoder 5: channel Z flexion encoder 6: + rotation motor 7: - rotation motor 8: channel A rotation encoder 9: channel B rotation encoder 10: channel Z rotation encoder 11: GND flexion encoder 12: GND rotation encoder 13: +2.5V 14: setpoint rotation 15: setpoint flexion
EPOS3 Encoder connector J4 	1: do not connect 2: +5 VDC / 100mA 3: GND 4: do not connect 5: Channel \bar{A} 6: Channel A 7: Channel \bar{B} 8: Channel B 9: Channel \bar{Z} 10: Channel Z
EPOS3 Motor connection J2 	1: Motor (+M) 2: Motor (-M) 3: do not connect 4: Motor shield
EPOS3 Power supply connector 	1 Ground of supply voltage 2 +Vcc. Power supply voltage

Table 5. JAIMY™ and EPOS3 connections, refer to [28], [29].