UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

Development of the vision based navigation system for mobile robot

Autor:

De La Fuente García, Eva

Responsable de Intercambio en la UVa

Marta Herráez Sánchez

Universidad de destino

Vilnius Tech

Valladolid, Junio 2022

TFG REALIZADO EN PROGRAMA DE INTERCAMBIO

TÍTULO: Development of the vision based navigation system for mobile robot

ALUMNO: Eva de la Fuente García

FECHA: 7/6/2022

CENTRO: Faculty of Mechanics

UNIVERSIDAD: Vilnius Tech

TUTOR: Vytautas Bučinskas

Resumen en español (max 150 palabras)

Se ha desarrollado un sistema navegación basado en visión para un robot móvil que actuará en tuberías subterráneas. El sistema debe detectar la presencia de obstáculos en la trayectoria planificada tales como objetos o agujeros en el suelo. Si alguno de estos es encontrado e impide la continuación del seguimiento del camino planificado por el robot, el sistema presenta las instrucciones para esquivarlo y continuar con la trayectoria. El sistema de visión está formado por una unidad principal de procesamiento Raspberry Pi 4B junto con un rayo laser rojo y la Raspberry Pi Camera module v2. La programación se ha realizado en Python con la utilización de librerías de OpenCV. Se han estudiado distintos enfoques tales como la transformada de Hough, algoritmo RANSAC y contorno de componentes conexos. Además, se ha diseñado e impreso en 3D el soporte para el sistema de proyección de laser y adquisición de imagen.

Palabras clave: Sistema de navegación, Raspberry Pi, línea laser, blob análisis, Python.

Abstract (max 150 words)

A vision-based navigation system has been developed for a mobile robot that will operate in subway pipelines. The system must detect the presence of obstacles in the planned path such as objects or holes in the ground. If any of these are found and impede the continuation of the robot's planned path, the system presents instructions to avoid them and continue the trajectory. The vision system consists of a Raspberry Pi 4B main processing unit together with a red laser beam and the Raspberry Pi Camera module v2. Programming has been done in Python using OpenCV libraries. Different methods such as the Hough transform, RANSAC algorithm and blob analysis have been studied. In addition, the support for the laser projection and image acquisition system has been designed and 3D printed.

Keywords: Navigarion system, Raspberry Pi, laser line, blob analysis, Python.

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF MECHANICS

DEPARTMENT OF MECHATRONICS, ROBOTICS AND DIGITAL MANUFACTURING

Eva de la Fuente García

DEVELOPMENT OF THE VISION BASED NAVIGATION SYSTEM FOR MOBILE ROBOT

Final Bachelor’s Project

Study programme MECHATRONICS AND ROBOTICS,

Code 612H73002

Vilnius, 2022

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
FACULTY OF MECHANICS
DEPARTMENT OF MECHATRONICS, ROBOTICS AND DIGITAL MANUFACTURING

APPROVED BY
Head of Department

(Signature)
Vytautas Bučinskas
(Name, Surname)
2022 05 24
(Date)

Eva de la Fuente García

DEVELOPMENT OF THE VISION BASED NAVIGATION SYSTEM FOR MOBILE ROBOT

Final Bachelor's Project

Study programme MECHATRONICS AND ROBOTICS,
Code 612H73002

Supervisor: prof., dr., Vytautas Bučinskas (Title, Name, Surname) ________ (Signature) 2022 05 24 (Date)

Vilnius, 2022

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Eva de la Fuente García, 20213649

(Student name and surname, Student ID)

Mechanics Faculty

(Faculty)

Mechatronics and robotics, MRfuc-18

(Study programme, academic group)

COURSE PROJECT (WORK, INTEGRATED PROJECT)

DECLARATION OF ACADEMIC INTEGRITY

16 May 2022

(Date)

I confirm, that my course project (work, integrated project), which topic is

Development of the vision based navigation system for mobile robot

is written independently. The material presented in this project (work, integrated project) is not plagiarized. Quotations from other sources used directly or indirectly are indicated in the literature references.

The course project (work, integrated project) was inspected by:

Vytautas Bučinskas

There is no contribution of other persons to the course project (work, integrated project). I have not paid any statutory sums of money for this work.

EVA DE LA FUENTE GARCIA

Signature (Name and Surname)

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
FACULTY OF MECHANICS
DEPARTMENT OF MECHATRONICS ROBOTICS AND DIGITAL MANUFACTURING

Production and Manufacturing engineering study field

Mechatronics and robotics study programme,

State code 612H73002

APPROVED BY
Head of Department

(Signature)

Vytautas Bučinskas

(Name, Surname)

(Date)

TASK FOR BACHELOR THESIS

13 May 2022 No. 1
Vilnius

For student Eva de la Fuente García.

Bachelor Thesis title: Development of the vision based navigation system for mobile robot.

Approved on _____________________________.

The Final work has to be completed by ____________.

TASK FOR FINAL THESIS:

Initial data:

The vision-based navigation system should be suitable for use in a mobile robot. Such a device shall be capable of detecting all types of obstacles, including holes in the ground, along the robot's path. The system has been developed to be able to operate in a low-light environment as a pipeline. Throughout the project, the design of the support for the vision system is developed along with the necessary software for its proper operation.

Explanatory part:

1. Introduction. Analysis of analogical devices. Substantation of the taken technical decision.
2. Calculations needed for the design process.
3. Description of the design and working principle. Electric-block scheme. Algorithm of management of device.
4. Work safety. General provisions and requirements for safe working and environmental protection. Work safety of specific devices.
5. Evaluation of economic indicators of the designed or upgraded device.
6. Final conclusions and recommendations.
7. Literature reference list.

Drawings:

1. General drawing of the device (1 sheet A1); 2. Assembly drawing of the device (node) (1 sheet A1); 3. Algorithm of management of device (1 sheet A1); 4. The work drawing of the chosen part (0.25 sheet A1); 5. The work drawing of the chosen part (0.25 sheet A1); 6. Economic indicators (0.5 sheet A1).

Supervisor .... (Signature) prof. dr. Vytautas Bučinskas (Academic Title, Name, Surname)

Task accepted

....
(Student's signature)

Eva de la Fuente García
(Student's Name, Surname)

2022-05-13
(Date)

Vilnius Gediminas Technical University

Mechanics faculty

Mechatronics department

ISBN ISSN

Copies No. ………

Date ….-….-….

Mechatronics and Robotics study programme bachelor (master) thesis.

Title: **Development of the vision based navigation system for mobile robot**

Author **Eva de la Fuente García** Academic supervisor **Vytautas Bučinskas**

Thesis language

	Lithuanian
x	Foreign (English)

Annotation

A vision-based navigation system for a mobile robot has been developed. It will move in an environment where the lighting conditions are low, more specifically in the underground tunnels where pipelines of heating system are placed. The vision system must recognize if the planned trajectory is free of obstacles or not and in case it is not, plan another path for the robot to follow.

The system consists of a red linear laser together with the Raspberry Pi camera module 2. In this way, the laser beam will be projected on the ground presenting discontinuities in case there is an obstacle in the way. This projection will be captured by the camera and Raspberry Pi 4 model B will process the image obtaining 3 possible outputs: there is no obstacle and therefore the planned path is continued, there is an obstacle on the right side and it is necessary to turn left or there is an obstacle on the left side and therefore it is necessary to turn right.

Throughout the thesis, three different algorithms have been studied and tested to implement the navigation system described above. Finally it has been chosen the one that is not only able to recognize obstacles on the road but also to recognize holes in the ground. Another important reason for deciding on the final algorithm was the computational load required by each method.

Structure: introduction, owerview of analog contructions, calculation of project, description of the contruction and operational principle, work safety, economic calculation, conclusions and suggestions, references.

Thesis consist of: **52** p. text without appendixes, **44** pictures, **4** tables, **22** bibliographical entries.

Appendixes included.

Keywords: Navigarion system, mobile robot, obstacle avoidance, Raspberry Pi, laser line, Hough transform, RANSAC (Random sample consensus), blob analysis, Python.

CONTENT

LIST OF PICTURES

LIST OF TABLES

INTRODUCTION

Currently, the number of vehicles with autonomous navigation is increasing considerably, for example autonomous cars or AGVs (Autonomous Guided Vehicles) used in industrial environments. Therefore, numerous obstacle avoidance systems are being developed.

This thesis will present the design and implementation of a vision-based navigation system that will provide a solution for autonomously following a path along an obstacle-free path. It should be noted that the obstacles on the trajectory can be of any shape and size and can even be holes in the ground. In case an object is detected that impedes the passage of the robot, another route must be calculated to avoid it.

Throughout the project, different solutions will be studied to implement the most suitable system for the requirements. Nowadays, many methods have been developed to achieve a safe trajectory avoiding obstacles found on a route. Infrared and ultrasonic sensors along with laser, cameras and other kind of sensors are used for this purpose. It is important to know which technique is the most suitable for the application due to the fact that each one has its drawbacks. Ultrasonic and IR sensors are really sensitive to sound and light therefore, it could suppose problems on detecting objects on the trajectory because of the noise.

Other solution that can be found on this field is the utilization of a camera. Consecutives frames are taken in order to update and compare the background. This method has two main drawbacks: first of all, calculation cost is high because it has to process many images and do a lot of calculations for detecting the background. Secondly, it should be noted that the distance from camera to the obstacles cannot be calculate as this method works only with one camera.

Recently, other ways for estimating the distance to an object have been developed using devices as 2D-LiDAR which consist on a laser radar. The main disadvantages of this gadget are not only durability but cost because of the fact that it has a rotating mirror.

After studying all the options, it has been decided to use a line laser and a camera. On the one side the laser spreads a horizontal line while the other side the camera processes the image detecting an obstacle if the line is distorted. Moreover, with this method, light and noise independent results are obtained with a low computational load.

Although in this work it is only necessary to know the presence the obstacle in the robot path in order to avoid it, it is worth mentioning that it is also possible to know the following information doing some trigonometric calculations:

1. Object and holes presence in the trajectory.
2. Distance between robot and obstacle.
3. Geometric characteristics of the obstacle.

Therefore, this navigation system is the most suitable for the robot's working environment. It gives good results in real time with a not excessive computational load and it turns out to be a compact system easily attachable to any robot. Finally, an estimation of the production cost has been made in order to verify the feasibility of manufacturing.

1. OWERVIEW OF ANALOGIC CONSTRUCTIONS

1.1. Ultrasonic sensor

Ultrasonic sensors are able to measure the distance to an object within a range of 2cm to 400cm with a tolerance of 3mm. For this, the device emits a high frequency sound signal (over 20KHz) that will echo off an obstacle and will be perceived by the receiver of the ultrasonic sensor (Gowtham Rajmohan, 2016). The distance is calculated by tracking the time between the emission and reception of the sound wave. The distance values obtained will be calibrated to a range of 0v to 5v.

This small sensor consists of two cylindrical metal transducers: emitter and receiver. The first one converts the electrical signal into sound pulses while the second one receives the sound pulses and generates a signal whose width can be used to determine the distance travelled.

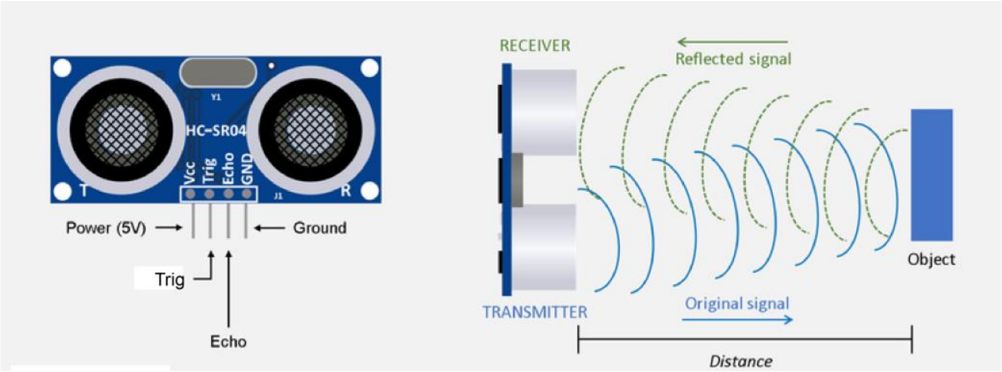


Fig. 1. Ultrasonic Sensor.

[https://www.teachwithict.com/hcsr045v.html]

As shown in figure 1.1, the component has 4 connection pins:

1. VCC: it is the sensor power supply at 5V.
2. Trig: generates a trigger pulse of at least 10 microseconds.
3. Echo: This pin switches from low level to high level at the end of the ultrasonic signal transmission and switches to low level when the echo signal is received.
4. Ground: 0V.

In this way, a trigger signal of at least 10 microseconds will be released to start the emission of an 8-pulse acoustic waveform. At the end of the transmission, the echo pin will change from 0 to 1 until the echo signal is received. The width of the pulse generated by this pin is proportional to the time it takes for the signal to be emitted until it is received by the device itself. This pulse is variable from 150 microseconds to 25 microseconds. The following image details its operation.

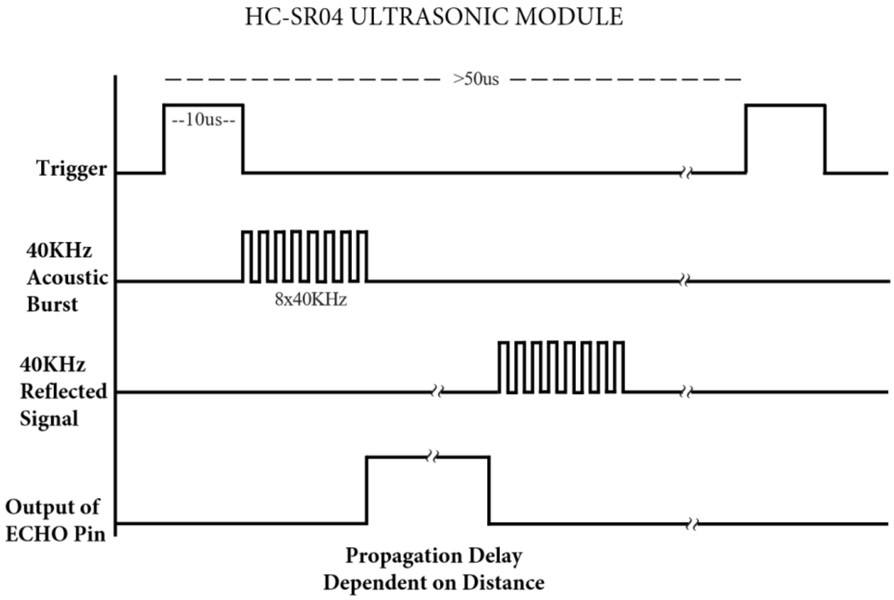


Fig. 2. Representation of trigger signal, acoustic bursts, reflected signal and output of echo pin.

Source: HC-SR04 User Guide.

The ultrasonic sensor is capable of detecting transparent objects as the wave can be reflected without any problem on non-opaque materials. In addition, it is not sensitive to fog, dirt or light. It should be noted that the effectiveness of the sensor is dependent on the effective angle of operation, approximately 30 degrees (Microcontroller Tips, 2019).

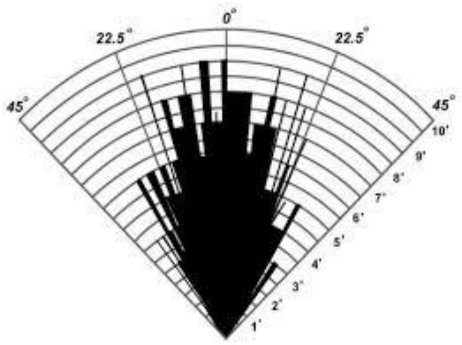


Fig. 3. Test of performance.

Source: HC-SR04 User Guide.

These types of sensors are limited because certain objects, due to their characteristics or position, do not reflect the wave in the direction of the sensor as shown in Figure 1.4. It can also be possible that the obstacle is too small to reflect the wave or that it is made of a material that completely absorbs the wave. Finally, it should be noted that a propagation medium is needed for the ultrasonic signal and therefore cannot work in a vacuum.

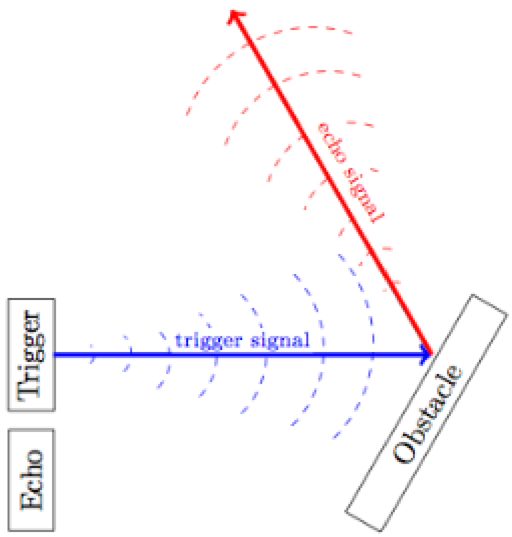


Fig. 4. Representation of ultrasonic signal deflected due to target object's position, resulting in error.
[https://forum.arduino.cc/t/problema-con-sensor-de-ultrasonido-hc-sr04/352252]

1.2. LiDAR (Light Detection and Ranging)

LiDAR (Light Detection and Ranging) also known as 3D scanning, is a device capable of measuring the distance from an emitter to an obstacle by using a pulsed light beam. As with the ultrasonic sensor, the device will measure the time it takes for the wave to propagate from emission to reception.

Fig. 5. 3D LiDAR sensors. Model MRS1000 Sick.
[https://www.sick.com/ag/en/detection-and-ranging-solutions/3d-lidar-sensors/mrs1000/c/g387152]

This type of sensor has the advantage of being able to analyse the entire environment covering 360º of its surroundings. On the other hand, obstacles that are lower than the height at which the laser is positioned cannot be detected. Therefore, the application for which the system is being developed must be taken into account in order to achieve a good positioning of the lidar.

The emitter sends out a series of infrared laser beams which reflect off objects in the environment generating a point cloud every scanning period that involves a 360-degree turnaround. Once the scan has been performed, each object is represented by a set of points. On the other hand, if there is no obstacle in the range of the scanner, no point is obtained. For each point in the cloud a lot of information can be obtained such as its position and the distance to it (Wei Chen, 2009).

It should be noted that an object is not given by a single point but by a set of them. Thus, it is not recommendable to use the point cloud generated by the scanner directly to generate a map of the environment where the robot is working. To obtain a map faithful to reality it is necessary to use some methods of clustering point cloud data.

The generated point cloud is processed by a computer to obtain a three-dimensional image of the environment in real time providing a map of the surrounding area. Since the lidar technology allows the creation of 3D models from the point cloud, in this map it will be possible to detect any kind of obstacle. Moreover, thanks to the fact that its wavelength is considerably smaller than that of the ultrasonic sensor, it is highly accurate in detecting small objects.

Finally, a grid map of the area to be analysed is constructed. This environment turns in small grids in order to be able to analyse and locate the objects with greater accuracy.

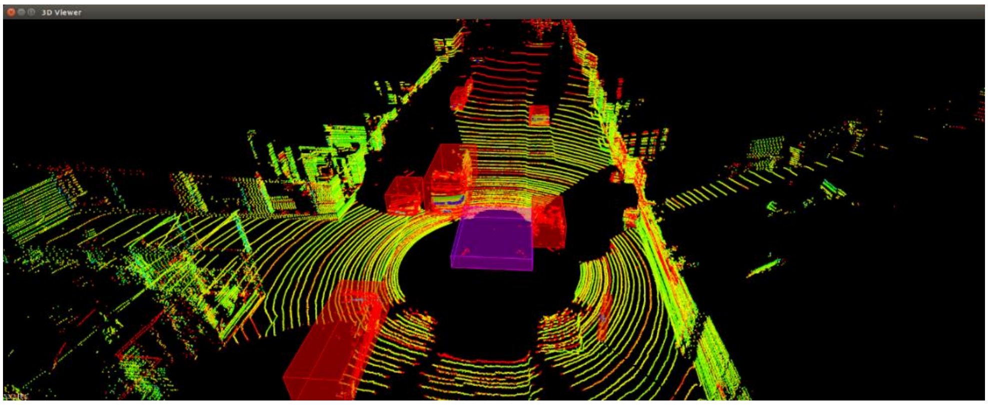


Fig. 6. LiDAR point cloud for object detection.
[https://edosoft.es/follow-the-lidar/]

On the other hand, this sensor is far more expensive than ultrasonic one. Moreover, it should be noted that the higher the desired accuracy, the more points to take from the environment and the greater the sampling frequency, the heavier the computational load. Another drawback with respect to the sensor described above on point 1.1, is that the light emitted by a LiDAR is harmful to the human eye due to the strong pulses.

1.3. Monocular camera and line-laser

This method consists in the configuration of a line laser and a camera. The laser beam will be projected on the horizon and it will be captured by the camera to be processed. In this way, the laser beam will be distorted in case of the presence of an obstacle.

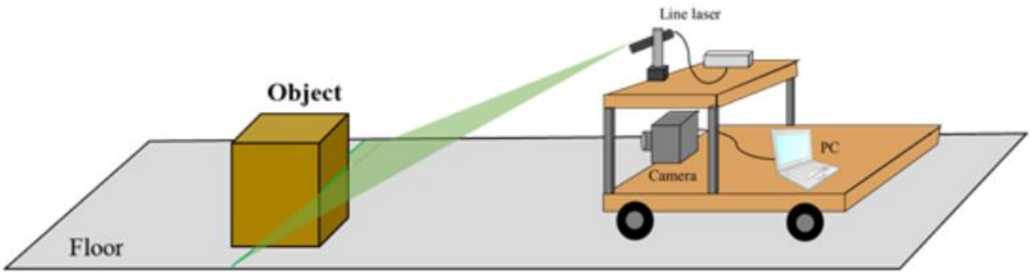
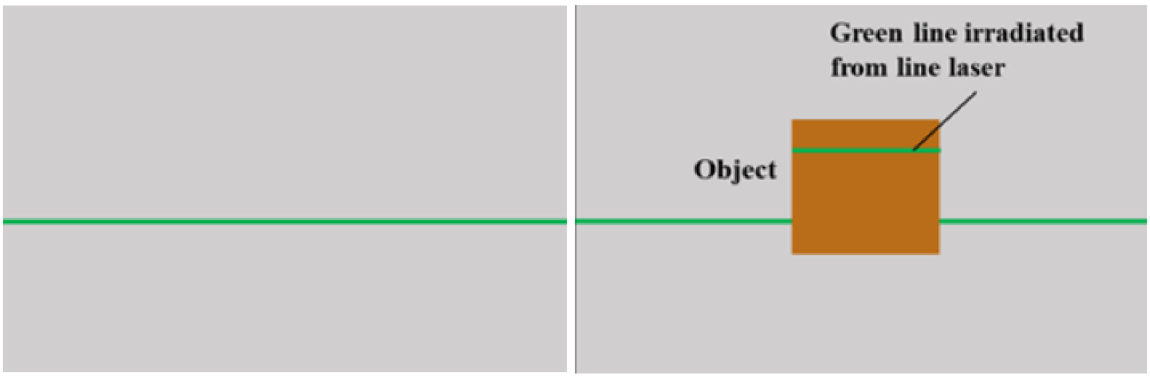


Fig. 7. System configuration.

Source: “A Method for Detection of Obstacle Using Line Laser and Camera”. Kyushu Institute of Technology

The distance between the laser and the camera will be fixed so that the dimensions of the object can be obtained by triangulation. In case of an obstacle the laser line will be cut off. On the contrary, if the path is free of objects, the line will not show any discontinuity. In this way, the camera will take the image of the environment in which the laser is reflected, this will be processed and it will be possible to discern if there is an obstacle or not (Miki Suetsugu, 2019).



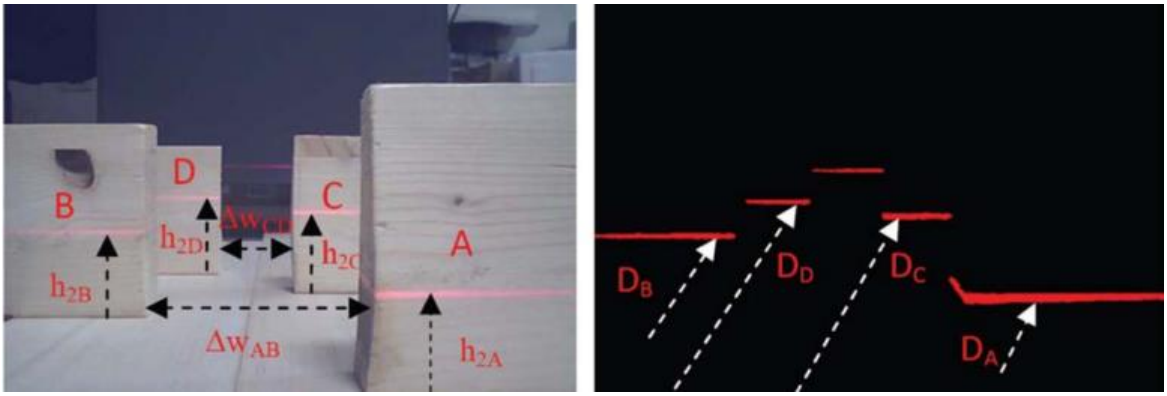
a) Laser without object.   b) Laser distortion with object.

Fig. 8. Image processing for laser line acquisition.

Source “A Method for Detection of Obstacle Using Line Laser and Camera”. Kyushu Institute of Technology

Since this method is based on the projection of a laser beam, in theory there is no limitation of the maximum measurable distance. However, some experiments have shown that beyond 600mm the laser beam is seen in a very faint form making the image incapable of being processed.

In contrast, it should be highlighted that this type of configuration not only allows the detection of the presence of multiple obstacles but it is also capable of measuring the distance to each of them at the same time. Some experiments have shown that the smaller the distance between the object and the robot, the higher the accuracy of the measurement (Guoqiang Fu, 2011).



a) Raw image. b) Image processed.

Fig. 9. Multiple object detection.

Source: Paolo Dario, "An Integrated Triangulation Laser Scanner for Obstacle Detection of Miniature Mobile Robots in Indoor Environment"

The three main factors influencing system performance are described below:

1. Ambient light.
2. Incidence angle of the laser beam.
3. Characteristics of the detected obstacle.

When the system is working in a dark environment, the laser beam is strongly reflected making it easier to separate the laser beam from the background. On the other hand, if the surroundings are brightly illuminated, the laser is perceived as weaker, making it more difficult to separate the laser from the background in the image. Therefore, the higher the ambient light the distance at which obstacles can be sensed decreases.

The second factor is the angle of incidence of the laser. The greater the angle of incidence of the laser, the more clearly it can be seen on the surface and the more accurate the measurements.

On the contrary, the smaller the angle of incidence, the system tends to look for objects farther away from the robot and the laser is seen in a more diffuse way, making it more difficult to process.

Last but not least, the properties of the detected obstacle also play a major role in the performance of the system. The strength with which the laser is perceived is dependent on the reflectivity of the surface on which it is projected. Therefore, not only the color of the obstacle but also the material of the obstacle is very influential. Very light colors have a reflectivity of more than 70% while dark colors have a reflectivity of less than 40%. This means that the darker the object, the more difficult it is for the system to detect it. Similarly, in materials such as polished aluminum the laser will be much more detectable than in other materials such as dark concrete (Miki Suetsugu, 2019).

1.4. Stereo vision

Stereo vision is a machine vision technique that can provide 3D measurements using two or more industrial cameras. The basis of stereo vision is similar to 3D perception in human vision and is based on the triangulation of rays from multiple viewpoints. In this way, a stereo vision system faithfully replicates how the eyes work to provide accurate, real-time depth perception. This is achieved by using two sensors at a set distance to triangulate similar pixels from both 2D planes.

The stereo vision model can be classified into three steps:

1. Selection of a point or small region in one of the images
2. Identification in the corresponding image of the other camera
3. Measurement of the disparity between the corresponding points.

These steps should be performed for all points that are considered necessary for a good interpretation of the environment. Depending on the requirements of the application, the depth can be calculated from the disparity found.

The difference that occurs when projecting the same three-dimensional point under the perspective of two or more cameras with different perspectives is called disparity. In the following figure the two points given by the coordinates (x_L, y_L) and (x_R, y_R) represent the one three-dimensional point captured by each of the cameras that compose the system. The location of this point on the vertical axis is the same in both images, while its horizontal location varies according to the perspective.

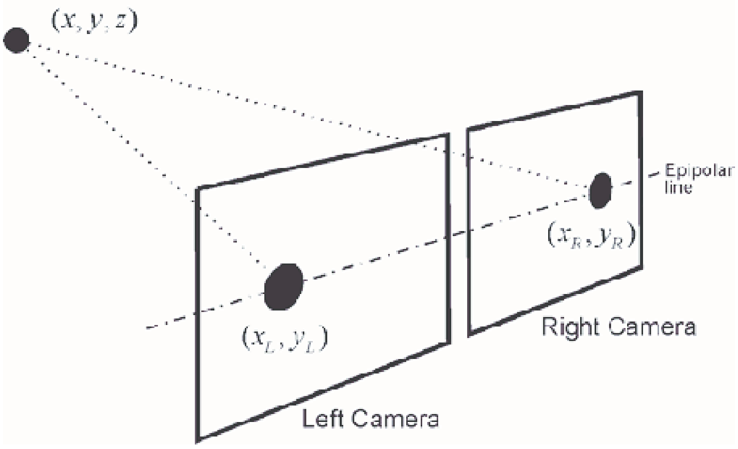


Fig. 10. Detection of a 3D point from the location of the 2D point on the images taken by the two cameras.

[https://www.researchgate.net/figure/Stereo-vision-principle-two-cameras-which-view-the-same-scene-detect-a-common-3D-point_fig1_221908788]

If the horizontal location of this point in the left view is x_L and in the right view is x_R, the disparity d that it presents is obtained by the difference of both values, as shown in the following equation.

$$d = x_L - x_R \tag{1}$$

However, knowing the disparity of a single point is not enough to have a good understanding of the three-dimensionality of the scene. It is necessary to generate a disparity map that can represent the whole, or a significant portion, of the observed environment.

There are two main types of stereo vision: active and passive. Active vision benefits from artificial illumination or a structured light source to assist visibility. In certain situations, this illumination is essential for the proper functioning of the system. On the other hand, there is passive vision which does not require any type of artificial illumination for correct operation. Therefore, only a connection, calibration and implementation of the cameras will be required (Tekkie Tak-Kei Chiu, 2009).

3D stereo vision can be CPU intensive along with significant speed loss when working in real time due to the required algorithms such as "Semi Global Matching" (SGM). This not only performs stereo matching using 2 cameras but also performs lens distortion compensation. This achieves stereo vision that works accurately and consistently over time.

As discussed above, passive 3D stereo camera systems can be implemented without the need for lasers / LEDs and can generally operate effectively in most ambient lighting conditions. This type of implementation can be much more affordable than other 3D vision technologies. In contrast, if the system is operated in low light or with untextured surfaces, stereo vision tends to perform poorly.

Another advantage of this sensing is that it does not impose restrictions on the range of motion on the target object, as other 3D profiling technologies do. This allows stereo vision to cope with long distances and moving objects.

Once calibrated, a stereo vision system can detect depth in real time, and when combined with the appropriate software to display the 3D image, it is possible to obtain color depth mapping for increased visibility as shown in the figure below.

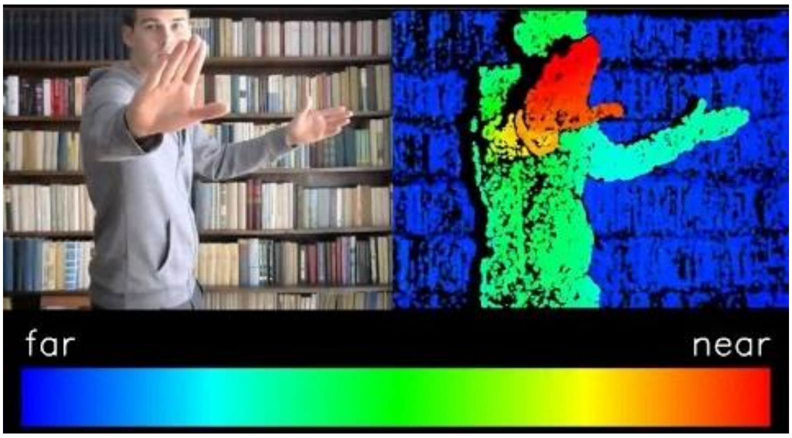


Fig. 11. Depth map generated from stereo vision.
[https://www.researchgate.net/figure/Depth-map-generated-from-stereovision_fig2_239618382]

1.5. Substantiation of the decision

After a study of all the possible solutions for the objective of this thesis, it has been decided to choose the proposal explained in point 1.3. Monocular camera and line-laser. This solution has many advantages over the others presented for the required application. First of all, it has a wide operating angle as opposed to an ultrasonic sensor. In addition, with this method it is possible to detect multiple objects at the same time up to a depth of 600mm without any problems.

Secondly, this method has no major restrictions with ambient light. As discussed above, a dark environment will increase the efficiency of the image processing in segmenting the laser beam from the background. However, this system is also capable of working in ambient light conditions with a smaller range of distances. For the application of this thesis this aspect will not suppose a problem.

Another aspect to take into account is that the material necessary to build the system does not have a high economic cost. It will require a camera, a laser beam and a computer capable of processing the image in real time. Specifically for this project we will use a Raspberry Pi 4 Computer Model B with 4Gb of RAM along with the Raspberry Pi Camera module 2. The support for the elements of the system have been printed in PLA with a 3D printer.

As far as image processing is concerned, it will only be necessary to observe whether the laser line has been cut. Thus, if there is only one line, it means that there is no obstacle present, while if there is more than one line, there is at least one object in the way. Unlike other solutions such as stereo vision that present a great complexity in software development along with a large computational burden by having to process and unify two frames, this solution is much more simplified in terms of software without presenting any drawbacks. By having only one camera, one image will only have to be processed in such a way that the laser beam is segmented from the background to be able to discern whether it has been cut by an object or not.

Finally, one aspect that has been critical to the decision of which method to implement has been the detection of holes. While systems such as an ultrasonic sensor or a LiDAR radar are not capable of detecting the presence of holes, this system is able to do so. Moreover, this detection does not imply an added complexity in the programming of the system.

It should be noted that although in this thesis it is only necessary to know whether or not there is an object on the path, with this system consisting of a laser and a camera it is possible to obtain additional information that other systems are unable to do. As long as the distance between the laser and the camera is fixed, it is possible to find the location of the obstacle with respect to the robot. In addition, other characteristics of the object such as its height and width can be calculated. In this way, the project is not limited at any time to be developed with more applications in the future with the need to replace the system.

2. CALCULATION OF PROJECT

In order to see the viability of the system designed in this thesis, SolidWorks 2021 software has been used to perform different studies such as stress and frequency studies. The whole system is designed in 3D with PLA (Polylactic acid) material which presents a good resistance for the objective to be achieved in this thesis.

2.1. Static study

The fastenings of the system have been defined together with the forces to which it is subjected have been defined for the purpose of the static study.

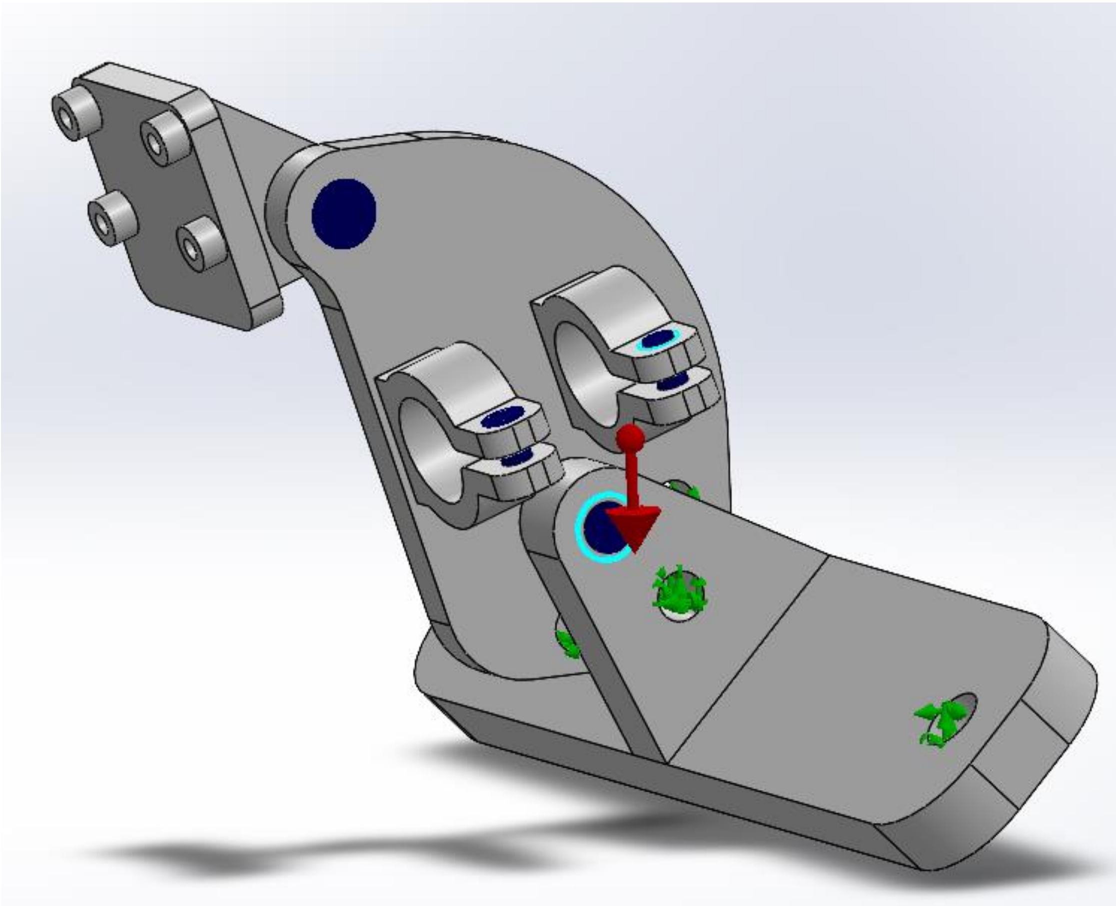


Fig. 12. Forces and fastenings of the system.

The green color shows the fixed fasteners. The two presented at the base of the support will be the union with the robot while the one presented on the projection of the base is the union with the rest of the system. On the other hand, the blue cylindrical shapes represent pin-type connectors as they are the ones that most closely resemble the function of the bolts located in that position.

In addition to the fastenings, the forces to which the system is subjected must be defined in order to make a study that is true to reality. First of all, the force of gravity has been defined, represented by a red line. Together with this we have defined the masses corresponding to the objects that interact with the system, the camera and the laser 3g and 13g respectively. These act as masses distributed on their corresponding supports.

2.1.1.- Stress study

Once the forces acting on the system have been defined together with the restraints, the stress study is performed.

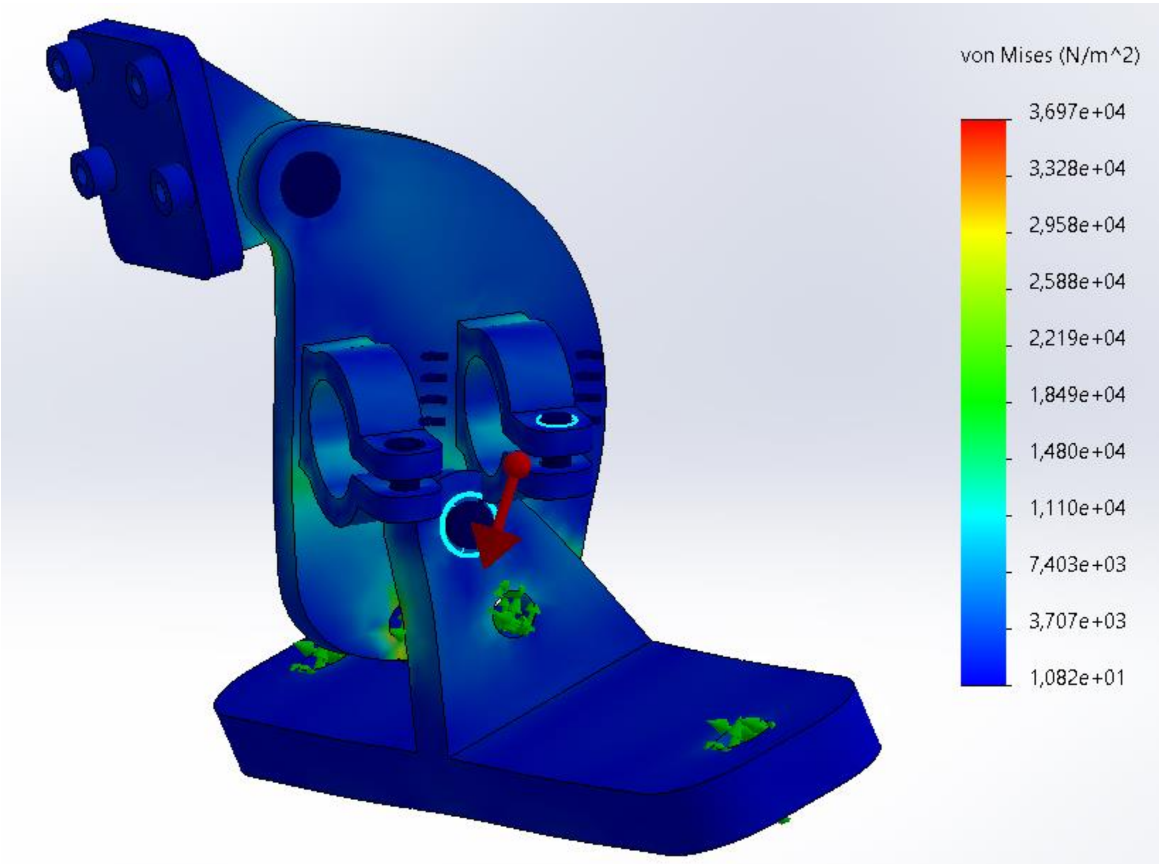


Fig. 13. Result obtained from stress study.

On the right side of the image, it can be seen the scale used for the study of stresses together with the criteria, in this case von Mises measured in N/m2. This study shows the force concentrated on a surface measured in m2. In general, a structure free of large stresses is obtained, except for the joint parts whose thickness is 5 mm. This is caused by the fact that these joints are in charge of supporting the weight of both the laser and the camera. For this purpose, bolts of metric 5 have been chosen.

2.1.1.- Displacement study

The following is the displacement study where red shows those parts of the system in which there will be a greater displacement versus the blue ones in which the displacement is minimal.

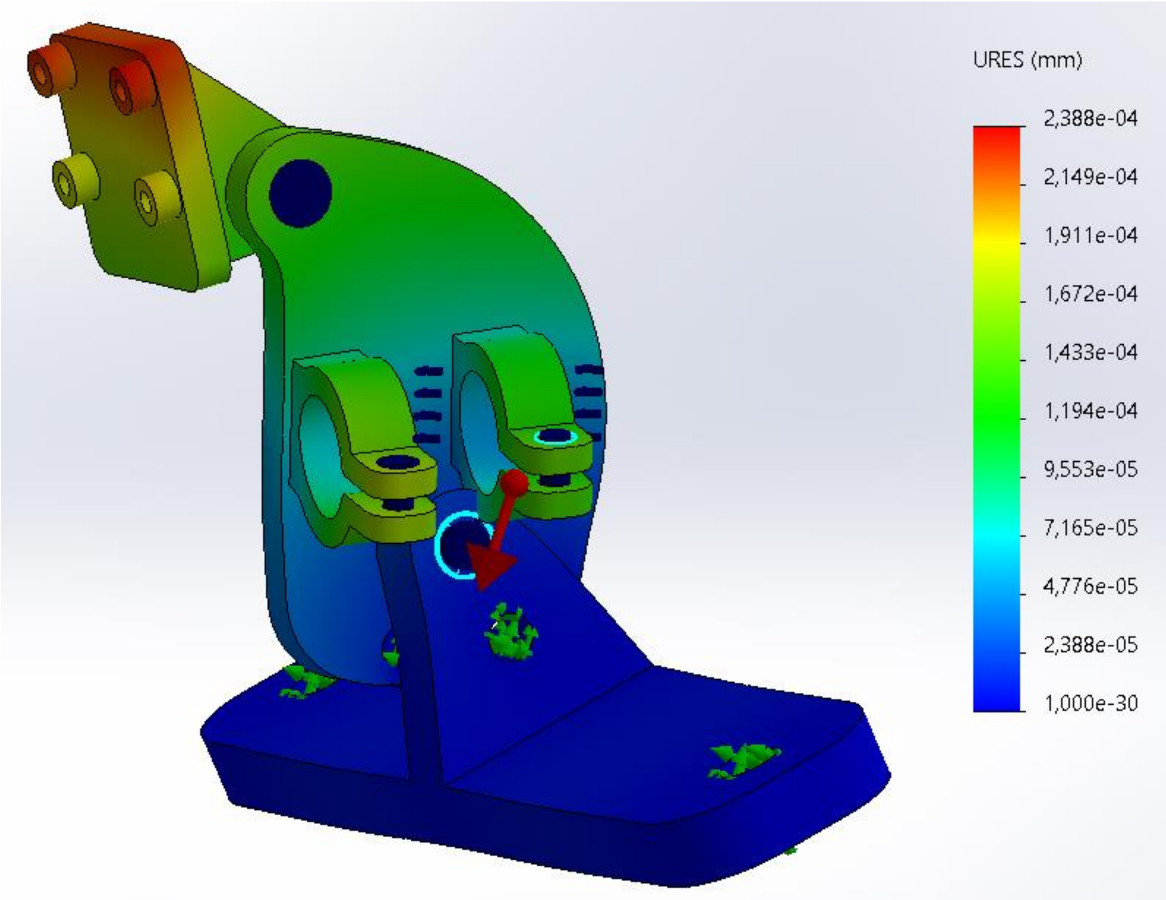


Fig. 14. Result obtained from displacement study.

It can be seen that the system will undergo less movement near the base supports. On the other hand, as the support gains height, the displacement is greater due to the fact that the fixings are located at the base of the structure. Although the displacement is maximum where the camera support is located, the magnitude is not significant as it is in the order of 10^-4 millimeters. For this reason, the displacement study is positive for the structure.

2.2. Frequency study

Once the study of the stresses and possible deformations of the system has been carried out and the validity of the system has been verified, the frequency study is performed. This study allows to calculate the natural frequencies in the system in order to avoid the phenomenon known as resonance. Resonance is an increase in amplitude that occurs when the frequency of a periodically applied force is equal or close to a natural frequency of the system in which it acts.

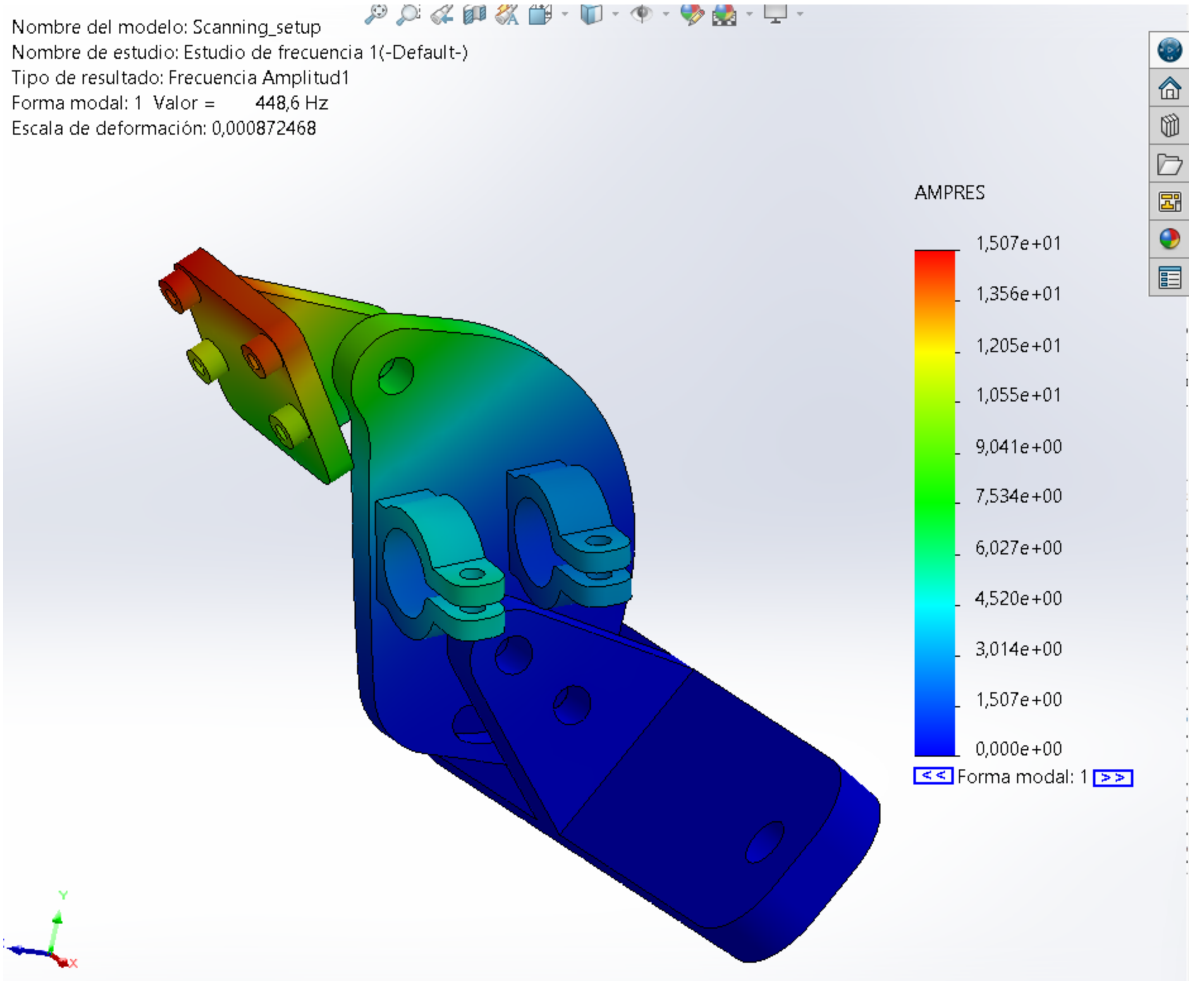


Fig. 15. Result obtained from frequency study.

It is important to perform this study when the components are interacting with some agent that generates some kind of vibration or loads that can excite the system. In the case of the design developed in this thesis, a natural frequency of 448 Hz has been obtained. In case it would be necessary to modify it due to surrounding conditions, some solutions would be to change the material or the geometry of the components.

3. DESCRIPTION OF THE CONSTRUCTION AND OPERATIONAL PRINCIPLE

3.1. Electric-block scheme

The robot in which the vision system is to be implemented consists of a switch that provides power to a 220VAC to 24VDC power supply. From this is fed the cooling system consisting of two fans together with the linear solenoid that transforms the electrical energy into mechanical energy or force. The emergency stop button that protects the whole system is connected to 24VDC and to the motors, in such a way that it will prevent the flow of current to the motors, stopping the movement in case it is activated.

The Raspberry is powered at 5DC through a 24VDC/5VDC converter. This in turn will feed 5V to the robot's position sensors which return a signal to the Raspberry of the position of the motors. The display control is also connected to the raspberry at 5V and can send and receive signals from the device. Finally, the camera is connected to the CSI (Camera Serial Interface) port.

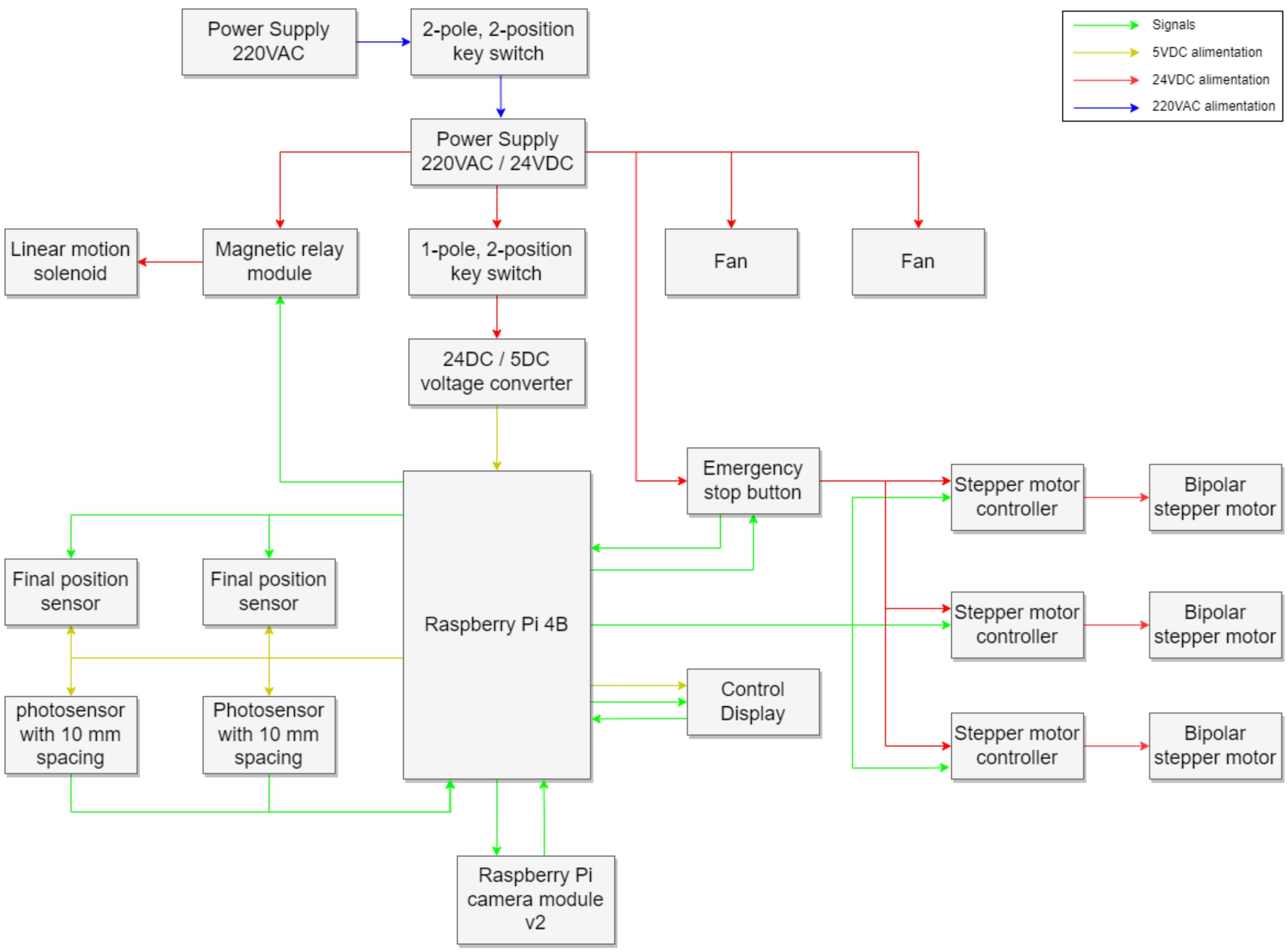


Fig. 16. Electric-block scheme.

The main component of the system is a Raspberry Pi 4B with a capacity of 4Gb of RAM. It is powered by the Broadcom BCM2711 chip along with a 1.5 GHz quad-core processor. In terms of connectivity, it is equipped with 802.11ac Wi-Fi / Bluetooth 5.0 and Gigabit Ethernet.

The Raspberry can be powered in two possible ways, but both at 5V. The first one is via USB-C with a current of 3A while the second way is via GPIO. In terms of video and sound, this device is equipped with 2 micro-HDMI ports that support 4K@60Hz displays via HDMI 2.0. It also includes a MIPI DSI display port, MIPI CSI camera port, 4-pole stereo output and composite video port. Finally, two USB 3.0 ports are available along with two USB 2.0 ports. In addition, there is a GPIO header which consists of 40 pins.

The camera module on the Raspberry Pi board is single channel, 8 megapixel and supports CSI-2 bus interface. It has a maximum frame rate capture of 30fps. This high-definition (HD) camera board connects to any Raspberry Pi or computer module, allowing it to create HD videos and photos. It uses Sony's IMX219PQ image sensor, which delivers high-speed, high-sensitivity video images. It also offers a reduction of image contamination, such as fixed pattern noise and speckle. To connect the camera module to Pi, a 15cm flat cable is connected to the module slots directly into the Pi camera's serial interface port (CSI). This device allows a fixed image resolution of 3280 x 2464 along with automatic black level calibration, automatic exposure control, white balance and band filter.

In the project, the Raspberry Pi 4B has been powered with the in-house adapter 220V AC to 5V DC using the UBS type C port. In addition, a 7-inch touch screen has been connected through one of the micro-HDMI ports to the HDMI port of the screen itself. With this device you can visualize the Raspberry operating system and perform actions on it with the help of peripherals such as a keyboard or a mouse. This is powered by one of the USB ports of the Raspberry to micro-USB port of the display. Finally, the Raspberry Pi camera module 2 has been connected to the MIPI CSI port designed for it.

3.2. Algorithm of management of device or node

The idea is to realize a vision-based navigation system, in such a way that it consists of a linear laser and a camera. The laser will project a line on the ground at all times while the camera will capture the image processing it and obtaining as output if the path followed is the same or not. The setup is shown below.

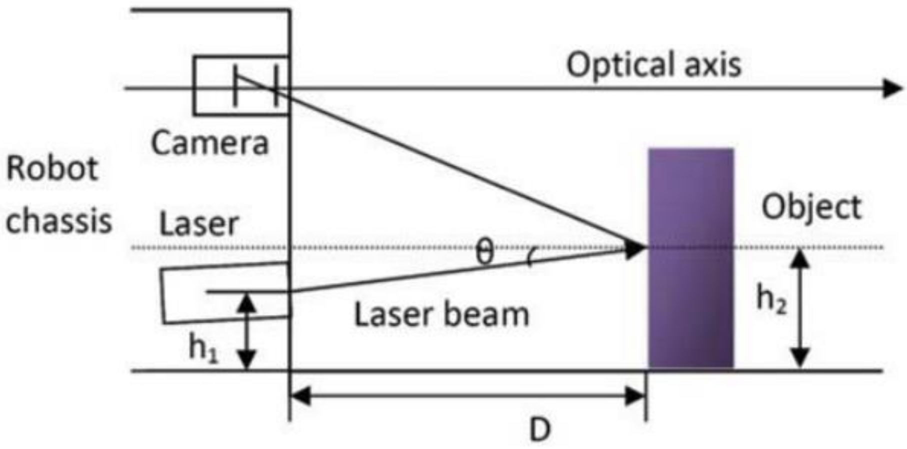


Fig. 17. Configuration of vision-based navigation system.
Source: Guoqiang Fu, Paolo Corradi, “An Integrated Triangulation Laser Scanner for Obstacle Detection of Miniature Mobile Robots in Indoor Environment.”

In this way, if there is no object present, the line projected on the ground will not have any kind of discontinuity. On the other hand, if there is an obstacle in the way, the laser projection will be cut off. Therefore, it is sufficient to analyze a strip of the image, corresponding to where the laser projection is located.

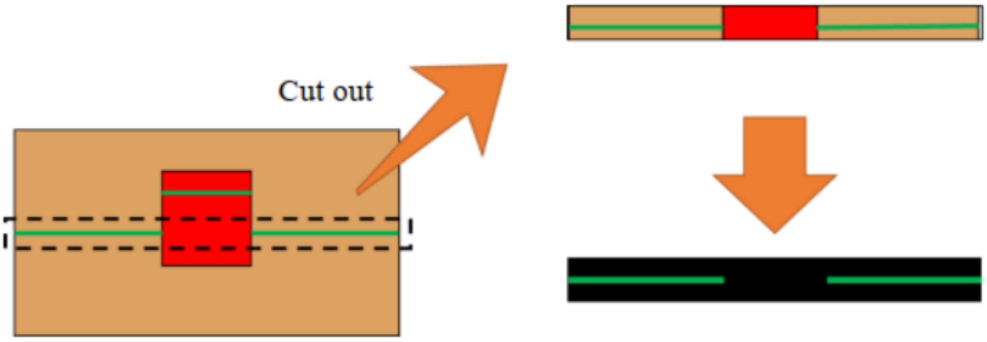


Fig. 18. Line segmentation from background of ground strip.
Source: Miki Suetsugu, Kanako Kinoshita, “Proposal of a Method for Obstacle Detection by the Use of Camera and Line Laser.”

The problem to be solved by the algorithm is first to segment the laser line, in this case red, with respect to the background. Subsequently, the search of the lines corresponding to the laser must be performed in order to detect if it has some kind of discontinuity. If so, it is assumed that the robot is facing an obstacle or hole and therefore must change its trajectory. Otherwise, the robot will be able to continue with the planned trajectory.

Nowadays there are many methods for detecting lines in an image that works pretty well in real time. Some of these are Hough Transform and RANSAC (RANdom SAmple Consensus). An algorithm for these two methods has been made in order to check the efficiency. The following is a brief theoretical explanation of the operation of both methods.

3.2.1.- Hough Transform

This process allows to find certain geometric shapes, which are of interest, within the image. In this thesis it is only necessary to detect straight lines, but it should be noted that it is a tool that can also detect curves in an image. Furthermore, this line detection system is capable of recognizing straight lines even if they present noise or are incomplete.

Before applying the Hough transform to an image it is necessary to first obtain the image of the contour points. For this purpose, one of the edge detection techniques is used.

The explicit equation of the line is the following where a is the slope and b the ordinate at the origin:

$$y = a \cdot x + b \qquad \textbf{(2)}$$

The objective is to find the different sets of aligned points that may exist in the image, or, in other words, the points in the image that satisfy the equation of the line for the different values of a and b.

For a given line, the parameters a and b are constants, where (x, y) are the variables. Suppose, however, that the parameters of the line are not known but the coordinates of one or more points on the line are known. In this case, the parameters can be considered as variables, while the coordinates will become constants.

The aim is to have the value of the parameters a and b as a function of the coordinates (xi, yi) of a given pixel, so the equation is:

$$b = -a \cdot x_i + y_i \qquad \textbf{(3)}$$

This transformation between the image plane (x - y coordinates) and the plane or parameter space (a - b coordinates) is called the Hough transform.

In this parameter space (also called Hough space), each point (x, y) of the image plane becomes a line of slope -x and ordinate in origin y, which represents the locus of all lines in the image plane passing through the point (x, y).

Consequently, the intersection in parameter space of two straight lines (x1, y1) and (x2, y2) gives rise to the pair (a0, b0) corresponding to the line that, in coordinate space, passes through the points (x1, y1) and (x2, y2), as illustrated in the figure. Thus, points aligned in the image plane are transformed in Hough space into straight lines that intersect at a single point (a0, b0).

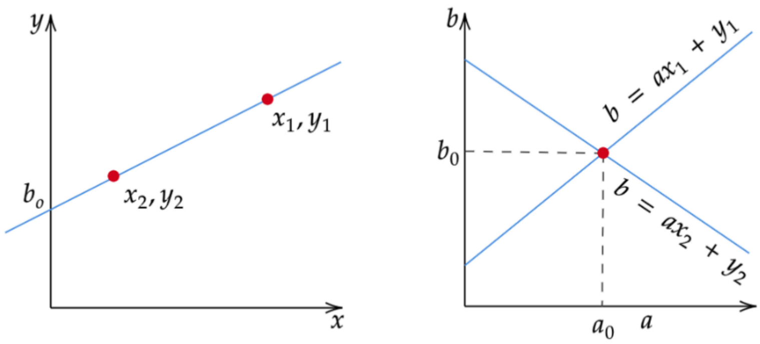


Fig. 19. Lines Detection with Hough Transform.
[https://towardsdatascience.com/lines-detection-with-hough-transform-84020b3b1549]

To apply the Hough transform it is necessary to discretize the parameter space in a series of cells called accumulation cells. All the possible values that the parameters a and b are discretized; in this way all the potential lines existing in the image will be searched, giving rise to what is called the accumulation matrix.

At the end of the process, the number of votes obtained in each cell indicates the number of points of the contour image that, except for discretization errors, satisfy the equation of the corresponding line.

Therefore, the cells with the highest accumulation of votes constitute the set of detected straight lines. For this purpose, a threshold is used, above which the line is said to exist. This threshold can be modified, so that one can be more or less restrictive when considering whether a cell is a straight line or not.

A problem that arises when using equation 4 to represent the lines is that both the slope and the ordinate to the origin can become infinite as the line becomes vertical. One way to solve this problem is to use the polar representation of the straight line.

$$\rho = x \cdot \cos\vartheta + y \cdot sen\,\vartheta \qquad \textbf{(4)}$$

where ρ is the distance from the line to the origin and θ is the angle that the normal forms with the x-axis:

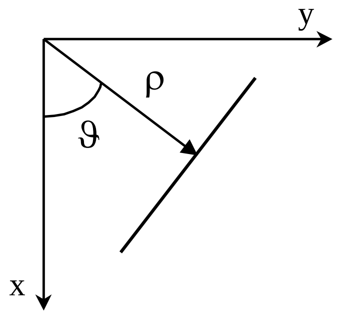
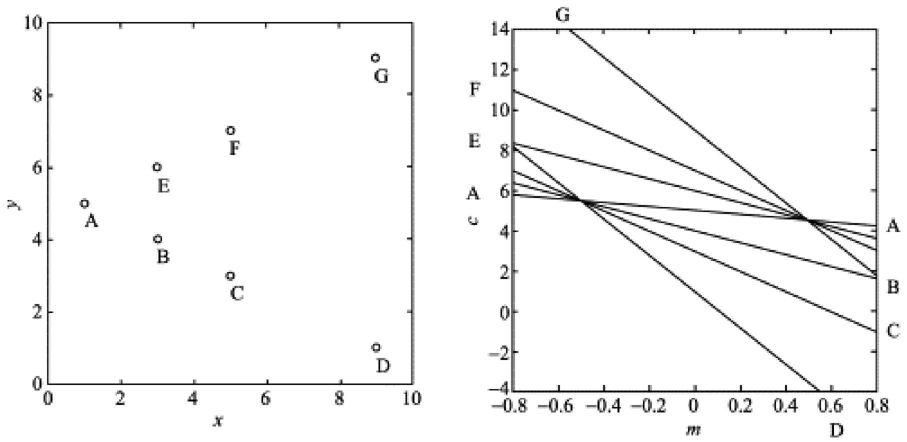


Fig. 20. Parameters of a line in polar coordinates.

Source: J. Hernández and R. Rodríguez, “Segmentación de Imágenes: Técnicas de Ajuste a Modelos”

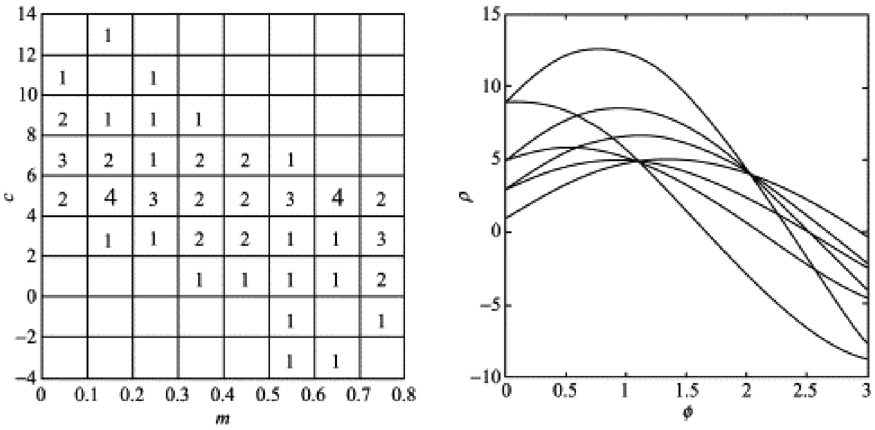
The way of constructing the accumulation matrix in the plane ρ-θ is similar to the one described above, the only difference being that, instead of straight lines, sinusoidal curves will be obtained. Thus, M collinear points belonging to a straight line as shown in equation 5, will give rise to M sinusoidal curves that will intersect, in parameter space, at the point (ρj ,θj).

$$\rho_j = x \cdot \cos\vartheta_j + y \cdot sen\,\vartheta_j \qquad \textbf{(5)}$$



(a) Image space.

(b) Parameter space corresponding to (m,c) parametrization.



(c) Accumulator array corresponding to (m,c) parameterization.

(d) Parameter space corresponding to {ρ,θ} parameterization.

Fig. 21. Hough Transformation process.

Source: Daniel C.I.Walsh, “The Importance Sampling Hough Transform”

Since the contour points used usually come from a contour detection method, their position may be affected by errors due to noise, spurious effects, illumination, etc.. Therefore, the cells with the highest accumulation will not appear in isolation in the parameter space, but will be found as part of clusters of cells with high votes. To control this in the call to the function that performs this transformation, it is possible to indicate the resolution of the concentration matrix of both ρ and θ.

If the orientation of the contours is known, information provided by the edge detection, for each contour point it is only necessary to vote in the box corresponding to that contour line defined by the point and orientation. Attention must be paid to the accuracy of the calculation of the gradient direction. Therefore, it is recommended to vote also in the boxes corresponding to +-20º deviations.

A drawback of the Hough transform is its computational cost since the number of operations is high. Although it is becoming less and less due to the development of computers.

3.2.2.- RANSAC

RANSAC is the abbreviation for "RANdom SAmple Consensus". It is a robust estimation algorithm that makes it possible to find a mathematical model from data contaminated with many outlier data.

For the use of this method, it is necessary to have previously performed the identification of contours with a suitable threshold of the image. Then, two random points of the image are selected and linearly adjusted by the least squares method to a straight line.

Once the parameters of the line made by the two randomly taken points have been obtained, a verification process begins. In this process it is checked if the mathematical model obtained is valid or not, and it is decided if it is necessary to obtain another one from another set of points. For this purpose, the number of points in the image that fit the line found with a tolerance *t* is determined. If the number of found inliers is the highest, it is saved as the current model. The algorithm will be repeated as many times as indicated by the defined *k* parameter (J. Hernández Hernández, 2008).

The algorithm will be repeated as many times as indicated by the defined k parameter. Finally, once the loop is finished, the model will be valid as long as d, the algorithm parameter, or more points match the model.

Therefore, the algorithm needs three parameters to control the estimation process of the model:

1. The maximum number of iterations **k** that the algorithm has to perform.
2. The tolerance **t** to determine when a pixel fits a model.
3. The number of inliers **d** that guarantees that a model is valid.

The algorithm is in principle programmed to stop after a certain number of iterations presenting the best model obtained but it can also be programmed to stop if the number of points close to the model is higher than a preset threshold, considering that the fit is already valid.

RANSAC is not a deterministic algorithm and therefore the model it provides is valid only with a given probability, and this probability increases as more iterations are carried out.

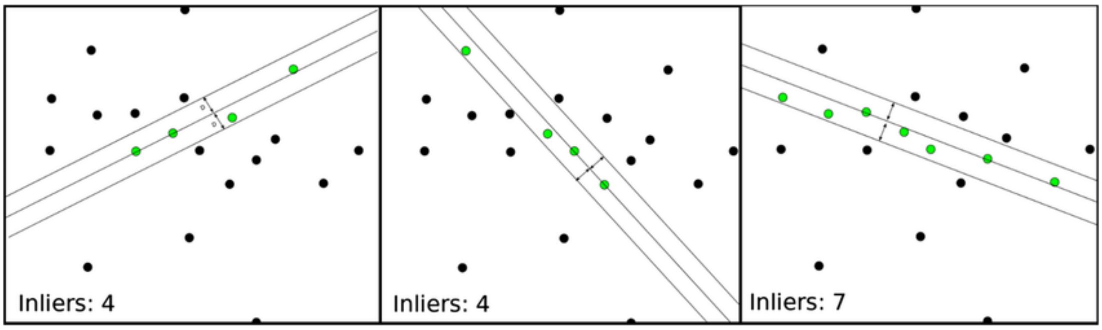


Fig. 22. RANSAC applied in a noisy dataset.

[https://www.researchgate.net/figure/An-example-of-line-fitting-to-a-noisy-dataset-using-RANSAC-Two-points-are-chosen-to_fig21_317964315]

Due to the fact that these methods are only capable of detecting straight lines, the navigation system would not be able to work on surfaces with irregularities in the ground. Not only that would be the problem, but also this system would not be able to detect the projection over a hole as it could take a curved shape. These limitations, together with the large computational load involved in executing these methods, have resulted in the search for another method for the navigation system. One of the possibilities studied is blob analysis.

3.2.4.- Blob analysis

Image labeling consists of locating connected components in a binary image. In order to determine their connectivity, some properties are analyzed looking for similarities. This technique locates different regions of an image making easier the segmentation task.

In OpenCV there are four possible ways to perform blob analysis:

1. cv2.connectedComponents
2. cv2.connectedComponentsWithStats
3. cv2.connectedComponentsWithAlgorithm
4. cv2.connectedComponentsWithStatsWithAlgorithm

The most common and the one that will be used in this thesis is the third one. The output is a matrix commonly called Stats whose length corresponds to the number of connected elements found. This matrix takes the form stats [label, COLUMN] where label identifies the component and column gives certain features of that component as where bounding box is, which size does it have and the centroid of it.

The flowchart is shown in Fig. 31 representing the algorithm explained above. It can be visualized in A1 format in the point referring to the drawings in the annexes at the end of this document.

Fig. 23. Flowchart of algorithm of management.

The system has as input the image captured by the Raspberry Pi camera 2. Then the first thing to do is the segmentation of the laser and the background for which the red channel of the RGB image will be taken since the laser is of that color. In this channel will appear red dots not coming from the laser beam which are not interesting to take into account, for this reason it has been decided to make a histogram to establish a threshold with which to discriminate the noise and the laser beam. Finally, a binarization is performed with the established threshold obtaining as output the binarized image.

It should be noted that it is not interesting to process the image completely because it is only relevant to study if there is any kind of obstacle or hole on the ground. For this reason, only a strip has been taken in the image to look for the presence of objects. In the same way, the width of the image is limited, since outside this band the presence of an obstacle would not imply the need to change trajectory. For example, if there is an obstacle on the far left and far right of the image, the robot is able to pass between the two without any problems.

Once the frame of interest of the image is obtained, the continuity of the laser line in the image is studied. For this purpose, each blob of the image has been labeled and colored in order to detect more quickly the number of laser beam sections. The image obtained is displayed on the screen together with the number of isolated objects recognized. The number of blobs will be high because the image contains noise that must be filtered out.

The filter used is a size filter defining the minimum and maximum area that the spots must have in order to be detected. For this, all the blobs of the image are traversed and the conditional sentence is established if the area meets with the thresholds. If it does, it implies that it is a laser section, therefore it is a spot of interest and it will be stored in a new array of filtered spots. In addition, a counter associated with the blobs of interest is incremented. On the other hand, if such a blob does not meet the established area thresholds, iterate to the next one until all the blobs have been covered. Finally, the number of laser fragments detected is displayed on the screen.

The next step is to go through the blob by blob again, counting the number of spots on the left and right side of the image. This information will be of great interest to recognize an obstacle-free path where the robot can deviate.

At this point the number of laser beam fragments in the image is accurately measured, which assumes that it is possible to discern the presence of an obstacle or not. The system will not only be able to do this, but in the event that the robot cannot continue its trajectory due to the presence of an obstacle in the path, it will give an answer as to where to turn.

A conditional statement is established that will differentiate whether there are discontinuities in the detected laser line or not. If there are no discontinuities it assumes that there are no obstacles in the path and that the robot can continue with the planned trajectory. On the contrary, in case the laser line presents discontinuities, it implies that there is at least one obstacle in the trajectory and that it is therefore essential to change the trajectory.

The direction of the new path that the robot will follow will be given by the difference of the line projected on the left side versus the one on the right side. Thus, if the amount of continuous laser beam on the left is longer than that on the right, it means that the path on the left contains fewer obstacles and irregularities in the path and vice versa.

Therefore, the system has three possible outputs: to continue with the planned trajectory and to recalculate a new path to follow by turning left or right. Finally, the image is re-acquired and the loop starts again, allowing a collision-free navigation of the mobile robot in real time.

3.3. Results.

The results obtained with the three algorithms explained above are shown below: Hough Transform, RANSAC and Blob analysis. The advantages and disadvantages of each of them will be studied.

3.3.1.- Results: Hough Transform and RANSAC

The detection of laser lines projected on the ground is similar for the RANSAC algorithm and Hough transform. In both algorithms a good laser detection is obtained as we can see in the following figures. The difference between the two methods is that on the one hand it is much easier to adjust the values of the RANSAC parameters with respect to Hough (D.A. Forsyth, 2016). On the other hand, the computation time of the first one is much longer than Hough because it must perform a larger number of operations.

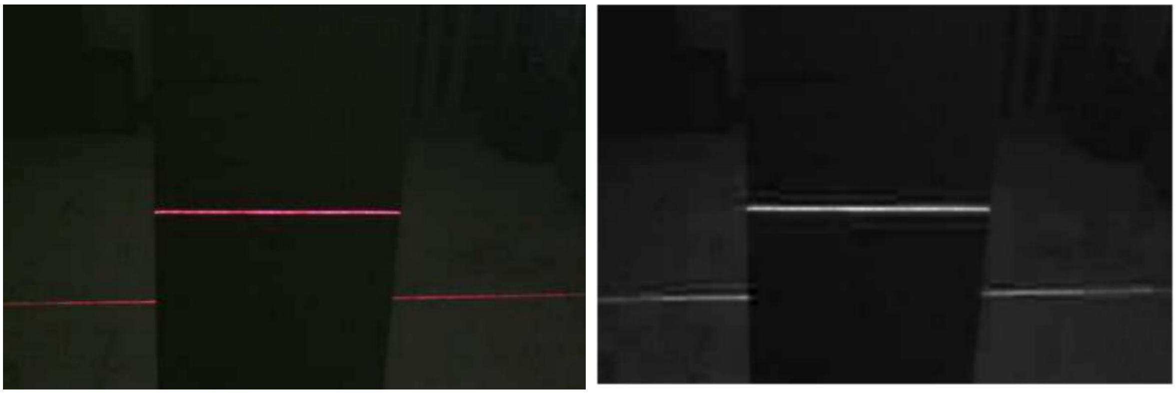


Fig. 24. Original image and image obtained with R channel.

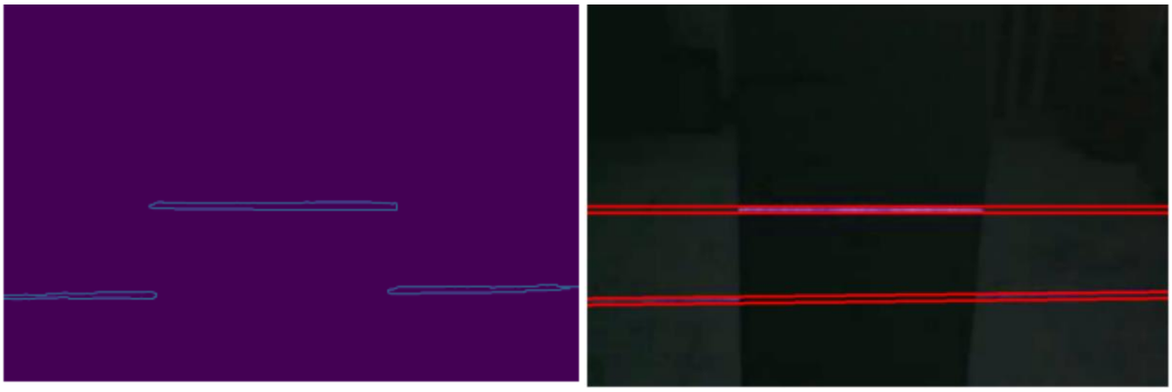


Fig. 25. Canny image and result image with lines detected.

```
In [14]: runfile('C:/Users/evafu/Documents/UVa/4Curso/Thesis_Python/v3/
nowi_Hough_lines.py', wdir='C:/Users/evafu/Documents/UVa/4Curso/
Thesis_Python/v3')
4 lines have been found
1 obstacles have been found
```

Fig. 26. Program Output.

First of all, the red channel of the RGB image has been taken since we are working with a red laser. Subsequently a contour detection has been performed together with the application of numerous filters to eliminate noise. Finally, the lines of the image are detected, in case there are more than two, it means that there is an object in the path of the robot.

As set out above, due to the fact that these methods are only capable of detecting straight lines so that the navigation system would not be able to work on surfaces with irregularities in the ground. Another problem that presents this method is that the system would not be able to detect the projection over a hole as it could take a curved shape. It has been decided to search for another method for the navigation system not only for these limitations but for the large computational load involved in executing these methods. Another possible method for detecting line laser is blob analysis.

3.3.2.- Results: Blob analysis

First of all, once the image has been acquired, the laser beam must be segmented from the background. To do this, the image can be processed in the RGB or HSV color space. In the case of RGB, it will be enough to take the channel of the same color as the laser, in this case red. HSV (Hue; Saturation; Value) is a cylindrical model that defines a color model in terms of its components. To carry out the segmentation with this color space, a pair of vectors of three values are defined to act as thresholds. Thus, values within the threshold will take the value 1 and those outside the threshold will take the value 0 [11].

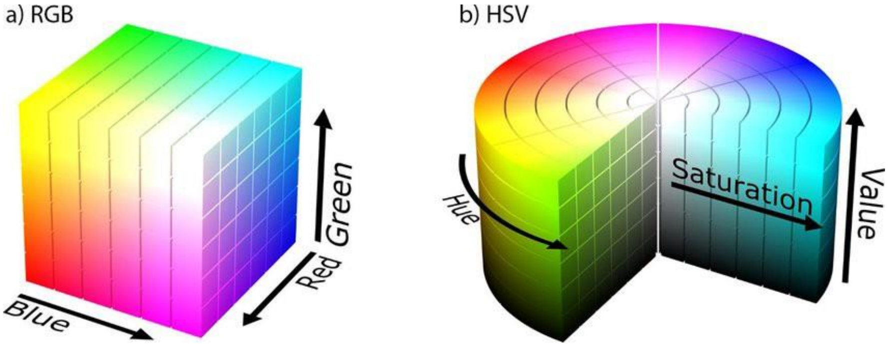


Fig. 27. RGB and HSV color space.
[https://www.researchgate.net/figure/Espacios-de-colores-RGB-y-HSV_fig1_351785444]

For the application developed in this thesis it has been decided to work with the red channel of the RGB color space since the environment will have a very low illumination highly contrasting with the laser beam.

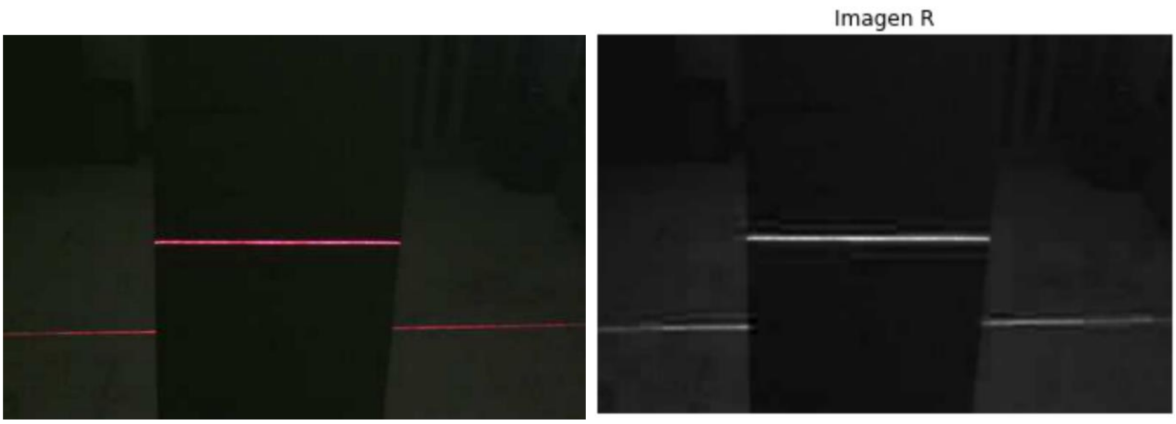


Fig. 28. Original image and image obtained with R channel.

Once the red channel of the image has been taken, a binarization is performed, for which it is necessary to establish a threshold. It has been decided to make a histogram of the image to see the weight and location of the pixels at high and low level, in this case we have taken the value 80.

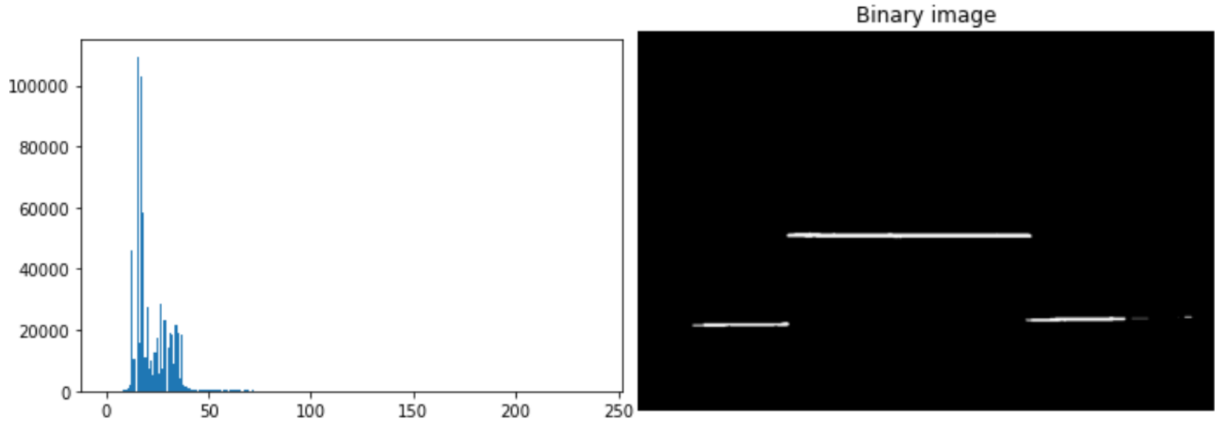


Fig. 29. Image histogram and binary image.

The laser and the camera will have a constant position along the robot's trajectory, which means that the laser line projected on the ground will always be in the same strip of the image. Ultimately this is the line of interest to process because if it has large discontinuities, it means that the robot is facing an obstacle or hole while otherwise the path will be clear. Therefore, and for faster processing, it has been decided to crop the image to the strip where the laser beam is expected to be projected.

The process of locating connected elements begins. The method will label more items than can be seen with the naked eye due to image noise. It should be noted that the background will also be labeled. In order to reduce the labels referring to image noise, a size threshold has been set. Thus, if the element to be labeled is smaller than the threshold, it will not be labeled. In the same way, it is interesting to eliminate the background label in order to completely segment the laser beam. Analogously, a maximum threshold has been set which means that if one of the labeled blobs is larger than the threshold, it will be eliminated. At the end of this process, laser beam segments are precisely obtained and projected on the ground.

Fig. 30. Final result: Blob analysis.

Finally, it is studied whether the robot can continue with the trajectory or must turn. For this purpose, the number of pixels corresponding to the laser projection on each side is calculated. For example, if the line on the left side of the image is almost complete, it means that the path on that side is free. If no more than the preset threshold number of low-level pixels corresponding to an obstacle is detected, the robot will continue with the its path.

```
In [75]: runfile('C:/Users/evafu/Documents/UVa/4Curso/Thesis_Python/v3/
nowi_blobs_discontinuty.py', wdir='C:/Users/evafu/Documents/UVa/4Curso/
Thesis_Python/v3')
Number of labels with background: 6
Number of labels after filter: 2
Turn right
```

Fig. 31. Program Output.

This method has many advantages over the two previously described RANSAC and Hough. First of all, it is not necessary that the robot is on a flat surface since it does not look for lines but for sets of points. Therefore, it is able to detect obstacles with any kind of shape and even holes in the ground. In addition, since it is not an iterative method as the previous ones, the computational load is significantly lower and together with the execution time.

4. WORK SAFETY

4.1. General provisions and requirements for safe working and environmental protection

Before starting the navigation system, it is necessary to perform a series of actions to ensure its proper functioning. First of all, it is necessary to make sure that all components are securely fixed to the support and that this in turn is attached in a stable manner to the robot. To do this, all the nuts and bolts that make up the assembly must be checked. Next, verify that the camera module is correctly connected to the Raspberry Pi.

Proceed to boot up the Raspberry Pi to start the calibration process. As mentioned above, this is a very benevolent navigation system because it can work in any environment as long as it is in low light conditions. Therefore, it is necessary to calibrate both the angle of the camera and the laser in such a way that the laser projects its beam on the ground and the camera is able to capture it in a certain strip of the image. At this point, the system is ready to be used.

Once the robot has performed the trajectory, the system must be properly disconnected. First, the Raspberry must be shut down from the graphical operating system. Once the red led on the board turns off, it is possible to disconnect the camera if desired. The laser should be turned off to save battery power and avoid possible incidents. In case the navigation system is not going to be used again on the robot, it is recommended to disassemble it with the help of a hexagonal wrench.

4.2. Work safety of Raspberry Pi 4B 4Gb

Raspberry Pi devices prove to be robust and safe as they were originally built for educational purposes for children. Despite this, the following precautions should be taken into account.

1. Connect the Raspberry to a suitable power supply. The device should be powered at a voltage of 5V dv. As it is model B, the operating current has the range of 700-1200mA. Make sure that the power supply used is approved and complies with the standards of your country. Do not power the Raspberry through the USB port as it is not designed for this purpose and may cause problems with the device.

2. Connect the power supply at the end. It is advisable to make all the connections required for the project to be carried out before connecting the Raspberry to the power supply.

This ensures that the user manipulates the contacts of the device in a safe way without possible discharges.

3. Turn off the device safely. Do not disconnect the Raspberry Pi from the power supply when it has not been turned off by the system. To do this, go to the top left where the Raspberry Pi logo is displayed and click Shutdown. You can also enter the system console and type the following command line:

   sudo shutdown -h now

   When the green light on the Raspberry is off it indicates that the power can be removed without causing problems. This ensures that the system and the data stored on the SD card are not corrupted and the system will be able to boot normally next time.

4. Add devices with the Raspberry Pi turned off. Before connecting devices such as LEDs, cameras, cards or boards, turn off and disconnect the Raspberry Pi from the power supply. Don't forget to check the wiring before booting the Raspberry Pi. If the wiring is wrong, it may cause irreparable damage to the device.

5. Caution with static electricity. The Raspberry is sensitive to this type of electricity so it is recommended to store the device in an anti-static bag and to use the device on an anti-static conductive floor.

6. Be careful with the SD card. The card is located in a slot of the Raspberry Pi that protrudes slightly in order to make it removable. Therefore, special care must be taken to ensure that it does not break as the device would not be able to boot in the absence of the card. It is advisable to remove the card and store it in a safe place when the Raspberry Pi is not being used.

7. Keep the Raspberry well ventilated. It is important to keep the device in a suitable temperature range for good device performance. Keep in mind that it can generate a large amount of heat when performing certain processing. In case you do not use the Raspberry Pi without a case or with the Coupé case, make sure that the ventilation of the device is correct. It should be noted that there are cases with fans adapted for the Raspberry Pi.

8. Pay special attention to the GPIO pins. It is easy to make a wrong pin connection. Use a guide that slips over the pins or check that you are connecting to the correct pin before booting the Raspberry Pi.

9. Resistors are necessary. Do not forget to connect resistors to components that require them, such as LEDs. This element is able to limit the current protecting the Raspberry. To do this, calculate the necessary resistor value with the required tolerance and look for the colours of the corresponding bands.

10. Use the Raspberry in suitable environments. Try to avoid using the device in environments with humidity or extreme temperatures. Under no circumstances subject the device to water or use it on conductive surfaces.

11. Use marked peripherals. All additional devices that you would like to connect to the Raspberry Pi such as keyboards, mice, monitors or others must be approved and therefore conform the relevant standards of each country.

4.3. Work safety of red linear - positioning laser (5mw)

Improper use of lasers can have harmful consequences for the body. The effects can range from slight damage to the skin to irreversible damage to the skin or eyes, in the form of different types of biological effects. Some preventive measures are as described below .

1. Do not point at people or animals at any time to avoid eye damage.
2. Do not use optical instruments such as prismatic or microscopes to view the laser beam. This is because these elements are capable of concentrating the light and can be extremely dangerous.
3. Do not project the laser beam onto surfaces capable of reflecting light such as mirrors or shiny surfaces. The reflected beam is as dangerous as the non-reflected beam.
4. Never leave the projection of this type of rays in a fixed and prolonged way, nor let it directly hit delicate materials susceptible to burns or flammability, such as fabrics, all types of fabrics, paper or cardboard, wood, etc.

5. ECONOMIC CALCULATION

The economic cost and profitability of the vision-based navigation system developed throughout this thesis are studied. For this purpose, a table has been made with all the elements that make up the system. It shows the identification of the elements together with the quantity necessary for the production of one unit and the price of the element.

Table 1. Material cost.

Number	Item	Quantity	Price per unit (Eur)	Sum (Eur)
1	Red linear - Positioning laser (5mw)	1	9,00	9,00
2	Raspberry Pi 4 Model B – 4 GB	1	68,43	68,43
3	Raspberry Pi Camera Module v2	1	41	41,00
4	Washer ISO 7089 (DIN 125) - M6	4	0,08	0,32
5	M6 Hexagon Nuts (DIN 934)	2	0,16	0,32
6	Plain Stainless Steel Hex, Hex Bolt, M6 x 20mm	2	0,29	0,58
7	Washer ISO 7089 - M5	6	0,08	0,48
8	M5 Wing Nuts (DIN 315 AF)	1	0,1	0,10
9	M5 Hexagon Nuts (DIN 934)	2	0,14	0,28
10	Plain Stainless Steel Hex, Hex Bolt, M5 x 20mm	3	0,37	1,11
11	Washer ISO 7089 (DIN 125) - M3	4	0,03	0,12
12	M3 Hexagon Nuts (DIN 934)	2	0,03	0,06
13	M3 x 12mm Full Thread Hexagon Bolts (DIN 933) - Stainless Steel (A2)	2	0,2	0,40
14	Vis CHC M2x12	4	0,04	0,16
15	M2 Hexagon Nuts (DIN 934)	4	0,03	0,12
	TOTAL (Eur)			**122,48**

The following is an analysis of the costs per unit produced. First of all, it is necessary to print three of the pieces that make up the system, for this a 3D printer capable of printing with the required precision will be necessary. It has been estimated a cost of 1500 €. This will have a payback time of 5 years and will be active during the working days of a year resulting in an amortization cost per unit of 15 cents. A roll of PLA (Polylactic Acid) filament is required for printing. It is possible to buy a 1Kg spool of this material in the market for 20€.

Another cost to take into account is the rent of the premises. This fixed cost has been estimated in the amount of 1500€ per month after researching the offer and demand of business premises in Lithuania. Currently the cost per megawatt hour in Vilnius is 0.15. Taking into account that a printer consumes on average 0,5KW the cost per hour will be 0,075.

Finally, an operator is needed to place the parts to be printed and to assemble them. It has been estimated that the preparation time needed to start printing, which may include small calibrations or adjustments of the printer, is 15 minutes. On the other hand, the operator will have another 15 minutes to perform the final assembly, since it is not very complex. Taking into account these times and the fact that the operator's salary is 15€/h, the cost associated with the worker's salary for each printed piece is 3.75€. Adding the assembly cost, the price per manufactured unit is €15 per assembled system.

Table 2. Variable and fix costs.

PLA cost (€/Kg)	**20**
Light cost (€/KWh)	0,15
Average consumption (KW)	0,5
Light cost per hour (€/h)	**0,075**
Operator hourly cost (€/h)	**15**
Preparation time for piece (h)	0,25
Postproduction time (h)	0,25
Cost operator per piece (€/piece)	**3,75**
Assembly cost (€/piece)	**3,75**
3D printer cost (€)	**1500**
Amortization time (years)	5
Active days per year (days)	250
Active hours per day (h)	8
Amortization cost (€/h)	**0,15**
Local renting (Eur)	**1500**
Time for produce 1 unit (h)	9
Units per month	17
Cost local per unit (Eur)	**88,23529**

Once the costs have been defined, the calculation of the cost associated with each part that makes up the system has been carried out. For this purpose, the weight of the part has been taken into account together with the estimated printing time. In this way it is possible to calculate the cost associated with PLA, electricity, labor and amortization of the machinery used. It should be noted that all the parts have been printed with 20% density, so this has not entailed any additional cost. As a final result, the cost of printing the three pieces that make up the system is 17.62€.

Table 3. Printed pieces cost.

Printed pieces	Base-Camera support joint	Camera support	Base-Robot joint	**TOTAL (Eur)**
Weight (g)	11,1	3,9	20,3	35,3
Printing time (h)	3	1	4,5	8,5
PLA Cost (Eur)	0,22	0,08	0,41	0,71
Electric cost (Eur)	0,23	0,08	0,34	0,64
Workforce cost (Eur)	3,75	3,75	3,75	11,25
Amortization cost (Eur)	0,45	0,15	0,68	1,28
Sum (Eur)	4,65	4,05	5,17	**13,87**
Assembly (Eur)				3,75
TOTAL SET				**17,62**

The total cost of manufacturing each navigation system is estimated. To do so, previous calculations will be used, resulting in a total cost of 228.33 €. To this it is necessary to add the profit that will be obtained by the sale of each unit. The selling price to the public has been set at 255 €, leaving a profit of 26.67 €, assuming about 10% benefit, per item. Below is a table specifying the costs associated with each unit along with the pie chart.

Table 4. Cost per unit produced.

Cost per unit	**Eur**
Material cost (Eur)	123,19
Electric cost (Eur)	0,64
Workforce cost (Eur)	15,00
Amortization cost (Eur)	1,28
Net profit (Eur)	26,67
Local renting (Eur)	88,24
Total cost without profit	**228,33**
Selling price	**255,00**

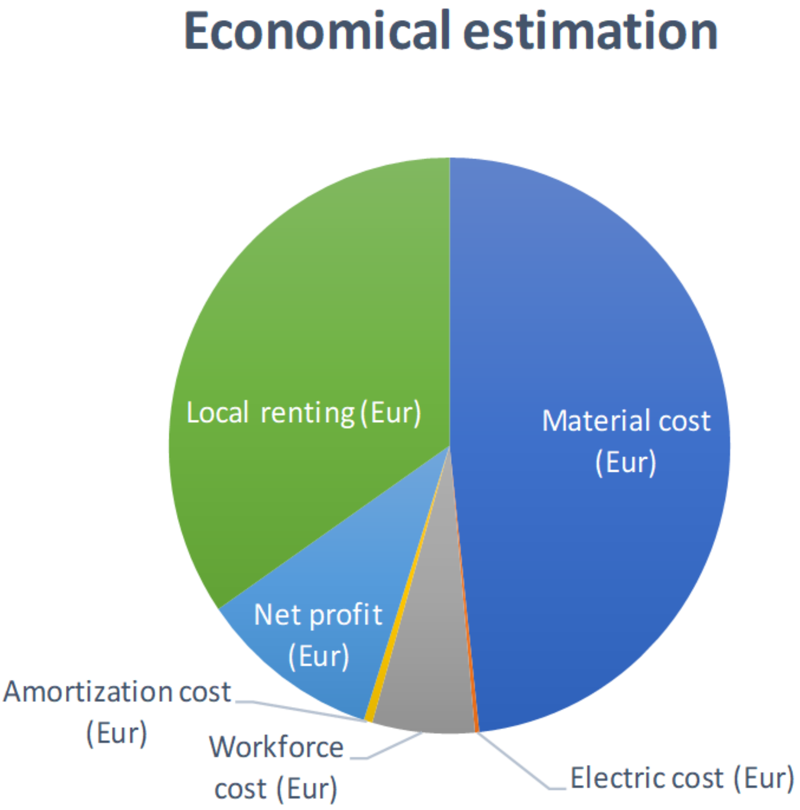


Fig. 32. Economical estimation per unit produced.

The payback period is then calculated. For this purpose, the profit obtained per unit sold has been equated to the cost generated by the production of the unit. The equations that model the profit and cost are as follows where *q* represents the quantity of units sold.

$$Cost\ (q) = 1500 + 228.33q \quad \textbf{(6)}$$

$$Profit\ (q) = 255q \quad \textbf{(7)}$$

Equating both equations result in the point of intersection. This point symbolizes the point at which profit begins to be made on the product manufactured (Dr. Audrius Dzikevičius, 2019). In this case, it will only be necessary to manufacture 57 units to start making a profit. These units represent an economic amount of both benefits and costs of 14535€.

$$Cost = Profit \quad \textbf{(8)}$$

$$1500 + 228.33q = 255q$$

$$q = 57\ units$$

$$Cost(57) = Profit(57) = 14535\ €$$

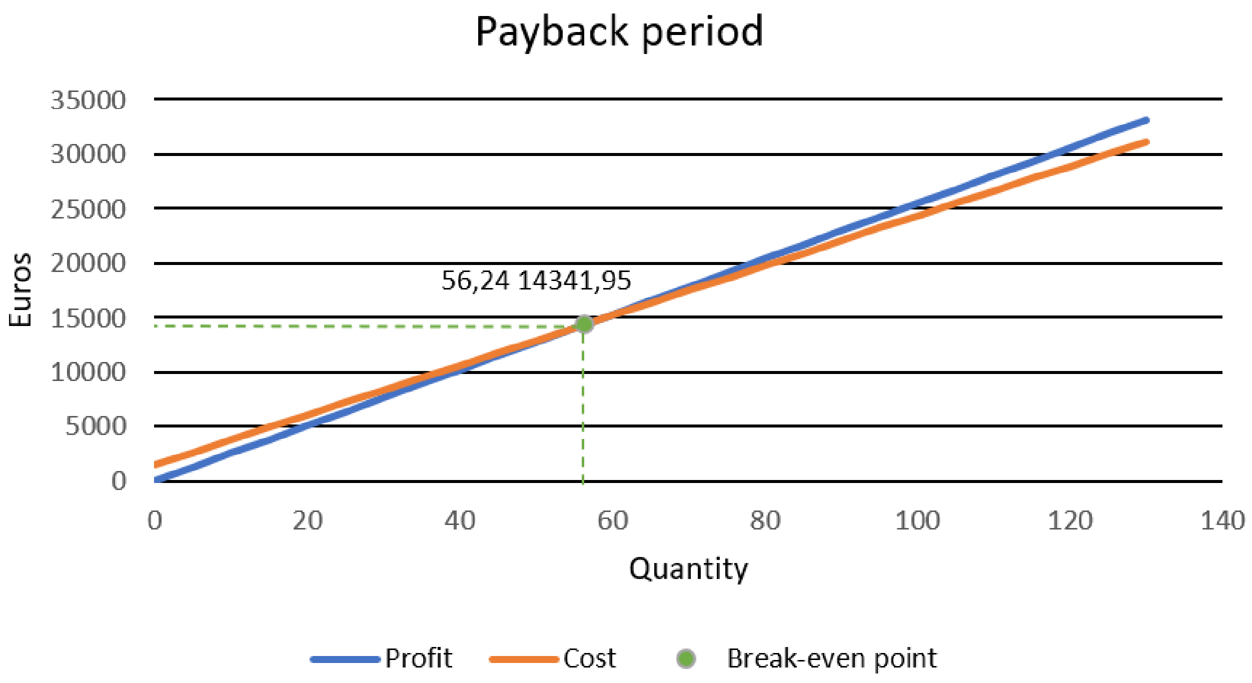


Fig. 33. Payback period.

This product has many advantages over other navigation systems that are currently available on the market. First of all, this system can be implemented on any robot that is going to work in low light conditions. It proves to be very flexible for numerous applications since both the inclination of the laser and the camera is adjustable for any kind of environment. The vision system is able to locate obstacles and holes not only on a flat floor but also on irregular or even circular floors such as a pipe.

It should also be noted that the durability of the system is high due to the low number of electronic components it requires being only a Raspberry, a camera and a laser, this is a great saving in the long term. In case it is necessary to replace any of the components it will not be a problem because all of them are easily accessible and removable.

CONCLUSIONS

1. Currently, the number of obstacle avoidance systems on the market is very large and continues to increase considerably due to the rise of autonomous vehicles. Some of them are sensors such as ultrasonic, infrared, LiDAR (Light Detection and Ranging), monocular camera and line-laser. Therefore, it is important to carry out an exhaustive study of the requirements in order to implement the system that best suits them. In the case of this thesis, a vision-based navigation system has been chosen.

2. The developed navigation system features a compact design, with easy maintenance, which will reduce future costs. Furthermore, it is able to react in real time without a high computational load, which is a prerequisite for the application. Therefore, the system is robust and cost-effective compared to other solutions available on the market.

3. Different design studies have been carried out in SolidWorks software such as stress study, displacement study and frequency study. A maximum stress in the system of the order of e-04 Nm2 has been obtained, which shows that the system is able to withstand the masses and forces applied on it without presenting problems. The displacement study is also positive since the maximum displacement is of the order of e-04 mm in the upper part of the chamber support. Finally, it has also been verified that the assembly does not resonate with the working environment. Therefore, it has been verified that the design is robust.

4. The electrical diagram of the system has been made showing the interaction of the different components of the robot. The vision system consists of a Raspberry Pi 4B together with the Raspberry Pi camera module v2 and a red linear laser.

5. Three different algorithms have been developed to identify the presence or absence of an obstacle in the robot's trajectory. The first two, Hough Transform and RANSAC, showed good results, however, being iterative methods, they were computationally intensive and slower. Being a real-time system, it is strictly necessary that the system reacts quickly. In addition, these two methods have great limitations of the robot's working space, as the surface on which the trajectory is performed must be completely flat and regular. This led to the search for a new method, blob analysis. This is able to work on irregular surfaces and can even detect the presence of holes in the ground. Another advantage is that the computational load is much lower than the previous ones and therefore the response is faster, being more convenient for a real time system.

6. Some important aspects to be taken into account when setting up and using the system have been detailed. These are relevant not only for the maintenance of the equipment but also for the safety of people in the proximity of the robot. In addition, some instructions for the correct operation of the system have been provided. In case these steps are not followed, the proper functioning of the system is not ensured.

7. A study of the costs involved in producing the designed article has been carried out. It has been found that this solution is significantly less expensive than others on the market. One of the most significant parameters is the payback period which establishes the number of units that must be sold in order to start making a profit, in this case it has turned out to be only 57 units.

LIST OF LITERATURE

1. “Application of artificial vision in production environments. Model adjustment and detection.” Master's Degree in Industrial Informatics. University of Valladolid.
2. Amruta Nikam, Akshata Doddamani, Divya Deshpande, Shrinivas Manjramkar. “Raspberry Pi based obstacle avoiding robot”. International Research Journal of Engineering and Technology (IRJET). Volume: 04 Issue: 02. Feb -2017.
3. D.A. Forsyth, I. Kokkinos. “Introduction to Computer Vision Hough Transform and RANSAC.” University of Virginia.
4. Detection of a specific color (blue here) using OpenCV with Python. December 2021. Detection of a specific color(blue here) using OpenCV with Python - GeeksforGeeks
5. Dr. Audrius Dzikevičius. Slides “Economic part”. Euroconsultant goup financial, investment and business consulting services. 2019.
6. Extracting a laser line in an image (using OpenCV). Stack Overflow. c++ - Extracting a laser line in an image (using OpenCV) - Stack Overflow
7. Fitri Utaminingrum, Hurriyatul Fitriyah, Randy Cahya Wihandika, M Ali Fauzi, Dahnial Syauqy, Rizal Maulana. “Fast Obstacle Distance Estimation using Laser Line Imaging Technique for Smart Wheelchair”. Faculty of Computer Science, Brawijaya University, Malang, Indonesia. 2016.
8. Gowtham Rajmohan, “Obstacle detection and avoidance methods for autonomous mobile robot”. June 2016.
9. Guoqiang Fu, Paolo Corradi, Arianna Menciassi, and Paolo Dario. “An Integrated Triangulation Laser Scanner for Obstacle Detection of Miniature Mobile Robots in Indoor Environment”. IEEE Xplore. August 2011.
10. J. Hernández Hernández, R. Rodríguez Vicente. “Implementation of a RANSAC algorithm for image segmentation.” September 2008.
11. Laser Pointer Safety. The Laser Institute. August 2017. https://www.lia.org/resources/laser-safety-information/laser-pointer-safety
12. Microcontroller Tips. “The working principle, applications and limitations of ultrasonic sensors”. August 2019. https://www.microcontrollertips.com/principle-applications-limitations-ultrasonic-sensors-faq/
13. Miki Suetsugu, Kanako Kinoshita, Shiyuan Yang, Siichi Serikawa. “A Method for Detection of Obstacle Using Line Laser and Camera”. Proceedings of the 7th IIAE International Conference on Industrial Application Engineering 2019.

14. Obstacle detection using Laser and image processing on LOGI-Bone. Dec 2014. Obstacle detection using Laser and image processing on LOGI-Bone - Blog - FPGA - element14 Community
15. Peter Neal. Obstacle detection using OpneCV. December 2014. https://bigfacerobotics.wordpress.com/2014/12/18/obstacle-detection-using-opencv/
16. R. Gowtham. "Obstacle Detection and Avoidance Methods for Autonomous Mobile Robot". International Journal of Science and Research (IJSR). June 2016.
17. Raspberry Pi Regulatory Compliance and Safety Information. Fanell. https://www.farnell.com/datasheets/1835724.pdf
18. Sam Westby Tech. How to Install OpenCV on a Raspberry Pi. (1188) How to Install OpenCV on a Raspberry Pi - YouTube
19. Tekkie Tak-Kei Chiu. "Stereo Vision Based Obstacle Avoidance in Indoor Environments". School of Mechanical and Manufacturing Engineering. University of New South Wales.
20. Todd Danko. Webcam Based DIY Laser Rangefinder. https://sites.google.com/site/todddanko/home/webcam_laser_ranger
21. Tzung-Han Lin, Chi-Yun Yang, and Wen-Pin Shih." Fall Prevention Shoes Using Camera-Based Line-Laser Obstacle Detection System." National Taiwan University. May 2017.
22. What is LiDAR and How Does it Work? GeoSLAM, March 2022. https://geoslam.com/what-is-lidarsurveying/#:~:text=LiDAR%20is%20an%20acronym%20for,maps%20of%20objects%20and%20environments.

THE ANNEXES

ANNEX 1: Installation of Raspbian on Raspberry Pi 4B

First of all, it was necessary to install a graphical operating system on the raspberry to be able to edit and run the image processing program. To do this, the Raspberry Pi 4B was first booted with the traditional MicroSD card. Then the following commands were executed to update the system:

sudo apt update

sudo apt full-upgrade

Once the update has been performed, it is recommended to restart the system. The reboot command is used for this purpose.

sudo reboot

Finally, the system is prepared to boot from a MicroSD card. To do this, enter the terminal settings and then select the "Advanced Options" option. A new menu will be displayed on the screen in which we will find the option A6 referred to "Boot Order". Again, another menu will be displayed in which you can configure the order of devices to boot the Raspberry Pi, including the MicroUSB and a USB in case of having any connected.

Many performance tests have shown that booting the system and using programs is much faster and smoother with a micro-SD than with a flash drive connected to a USB port. Using a flash drive in the Raspberry will cause the micro-computer to run much slower than using a memory card. In the development of this navigation system a fast image processing in real time is needed so it has been decided to boot the system in the fastest possible way, that is, by MicroSD.

sudo raspi-config

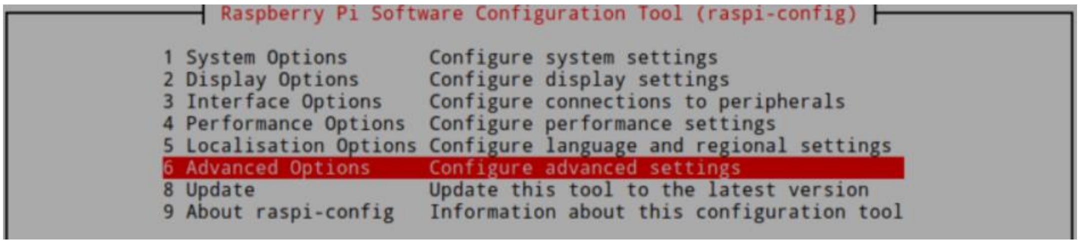


Fig. 34. Raspberry Pi software configuration tool.

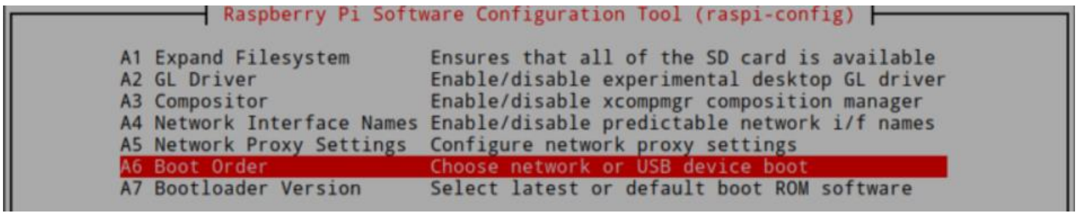


Fig. 35. Advanced Options menu.

Fig. 36. Boot Order menu.

Once the Raspberry Pi 4 is configured, it is possible to install the operating system on the SSD. To do this, we have used the Raspberry Pi Imager software that allows to install Raspbian in a simple way. For this, it is only necessary a MicroSD card formatted with the FAT32 file system and a minimum capacity of 16GB.

First of all, it was necessary to install this software on a laptop with SD card reader. Raspberry Pi Imager is available for Windows as well as for macOS and Ubuntu for free. Once installed, insert the SD card into the reader and launch the program.

Fig. 37. Raspberry Pi Imager software.

Now the first step indicated by the software is to choose the operating system. For this thesis it has been chosen the Debian 11 system called "Bullseye". This system is also the one recommended by Raspberry Pi Imager to be installed. Then choose the device where to install it, in this case on the MicroSD. Finally, click on the "Write" option to start writing the operating system to the selected memory. Afterwards a success message is received to mark the end of the installation process.

ANNEX 2: Installation of OpenCV on Raspberry Pi 4B

First of all, it is needed to check if it is being used all memory available of the system. This can be done accessing to the terminal and writing the following command.

df -h

If the space size shown at /dev/root is not close to memory card size, suppose that it is need to expand filesystem. In this case, go to raspi-config with *sudo raspi-config* command, then advanced, expand filesystem.

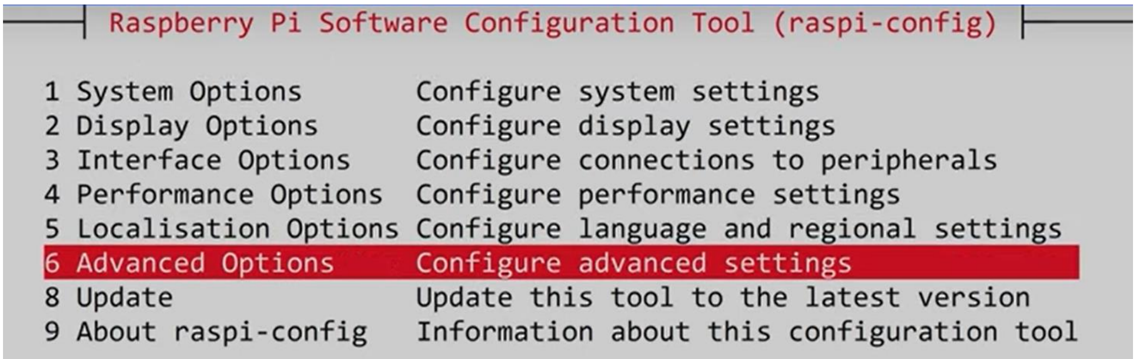


Fig. 38. Advanced Options menu.

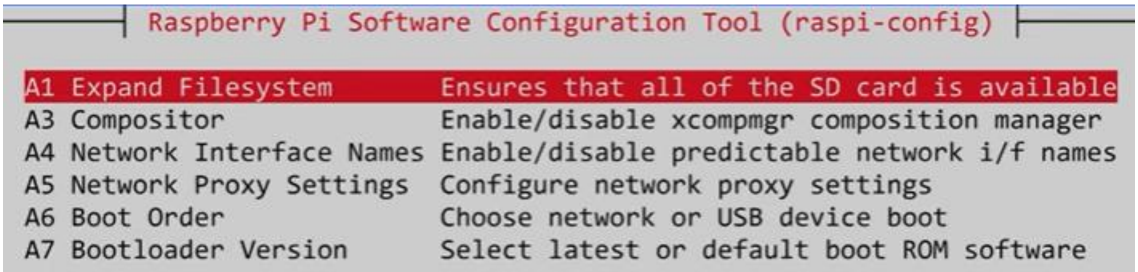


Fig. 39. Expand Filesystem.

In order for the changes to take effect, not only it is necessary to restart the Raspberry Pi but also to update and upgrade the hole system.

sudo reboot

sudo apt-get update && sudo apt-get upgrade

Next, a folder called project is created in which pip and virtualenv are installed. Once the installation has finished, a new virtual environment is created and activated called env.

mkdir project

cd project

python3 -m pip install virtualenv

python3 -m virtualenv env

source env/bin/activate

Now whenever you are working in the virtual environment, this environment will be indicated in brackets in the terminal as shown below.

Fig. 40. Activation and **e**nvironment workspace.

Once in the workspace previously created, in this case env, the installation of numerous packages necessary for the use of OpenCV and Python is done.

sudo apt install -y build-essential cmake pkg-config libjpeg-dev libtiff5-dev libpng-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libfontconfig1-dev libcairo2-dev libgdk-pixbuf2.0-dev libpango1.0-dev libgtk2.0-dev libgtk-3-dev libatlas-base-dev gfortran libhdf5-dev libhdf5-serial-dev libhdf5-103 libqt5gui5 libqt5webkit5 libqt5test5 python3-pyqt5 python3-dev

In addition, as this project is going to use Raspberry Cam, it is necessary to configure and install this device. For its installation we will use pip again.

pip install "picamera[array]"

It will not be enough with this alone but it is necessary to activate the camera from the device configuration. It will be necessary to go to the configuration as previously done, then *Interface Options*, *Legacy Camera Support* and enable it.

sudo raspi-config

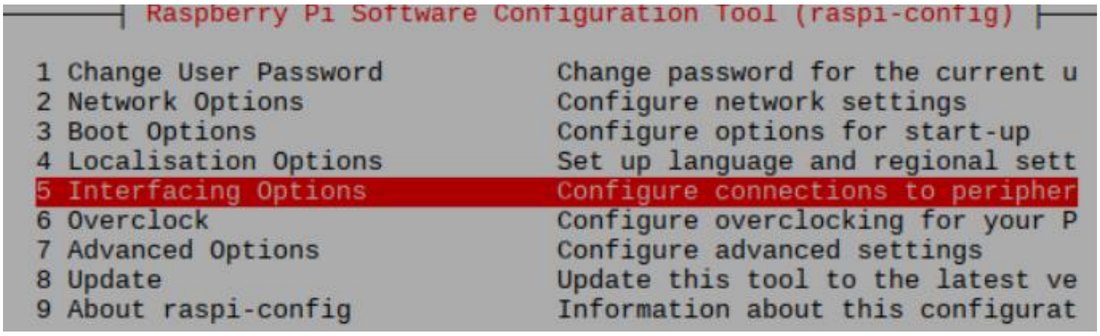


Fig. 41. Interfacing Options menu.

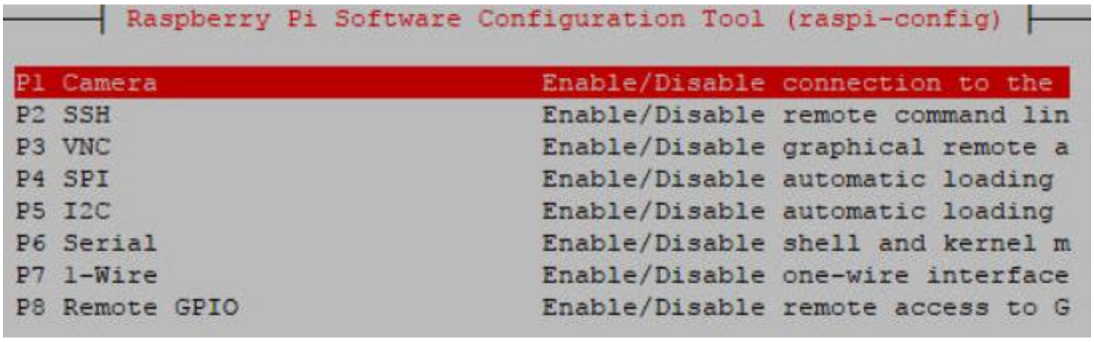


Fig. 42. Enable camera.

Finally, the system is ready for installing OpenCV through pip.

pip install opencv-contrib-python

This process will take about two and a half hours. Once the installation has finished, OpenCV is ready for being used. At the terminal is possible to open Python interpreter and check the OpenCV version importing cv2 and using *cv2.__version__* command. As a result, in this thesis the version installed has been ‘4.4.0’.

ANNEX 3: Code of algorithm.

```
"""
Created on Mon Apr 11 12:44:31 2022
Program that finds the laser segments projected on an image in real time.
Different filters are applied to remove noise and label only the laser lines in
the image. Finally, it is decided whether or not the robot is able to follow
the planned trajectory. In case it is not possible, it is decided in which
direction it should turn depending on the position of the obstacle.
@author: Eva de la Fuente García
"""

import sys
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math

def colorea_etiquetas(num_labels, img_labels):
    '''displays image of labels with random colors'''
    # Map component labels to hue val, 0-179 is the hue range in OpenCV
    label_hue = np.uint8(179*img_labels/np.max(img_labels))
    blank_ch = 255*np.ones_like(label_hue)
    labeled_img = cv2.merge([label_hue, blank_ch, blank_ch])
    # Converting cvt to BGR
    labeled_img = cv2.cvtColor(labeled_img, cv2.COLOR_HSV2RGB)
    # set bg label to black
    labeled_img[label_hue==0] = 0
    return labeled_img


if __name__ == "__main__":
    #%% LOAD image and take R channel
    img_BGR = cv2.imread('image9.png')

    if img_BGR is None:
        print('image does not found')
        sys.exit(1)

    R=img_BGR[:,:,2]
    plt.imshow(R,'gray')
    plt.title('Imagen R')
    plt.axis(False)
    plt.show()

    #%% HISTOGRAM
    # Trnasform image into vector with reshape(-1)
    plt.hist(R.reshape(-1), bins = 256)
    plt.show()

    #%% BINARY IMAGE
    THRESHOLD = 80
    img_bin = cv2.threshold(R, THRESHOLD, 255, cv2.THRESH_BINARY)[1]

    plt.imshow(img_bin,'gray')
    plt.title('Binary image')
```

```
    plt.axis(False)
    plt.show()

    #%%FLOOR STRIPE
    h,w,_=img_BGR.shape

    Y_MAX=math.trunc(h*0.9)
    Y_MIN=math.trunc(h*0.7)

    stripe_floor = img_bin[ Y_MIN : Y_MAX, :  ]

    #%% LABEL components
    num_labels, img_labels, stats, _ =
cv2.connectedComponentsWithStats(stripe_floor)
    print('Number of labels with background:', num_labels)
    #Showing Image after Component Labeling
    imgEtiq = colorea_etiquetas(num_labels, img_labels)
    plt.imshow(imgEtiq)
    plt.axis('off')
    plt.title("Image after Component Labeling")
    plt.show()

      #%% SIZE FILTER
    #Take area of stats

    MIN_SIZE = 100 #regions smaller than MIN_SIZE are removed
    MAX_SIZE = 0.1 * stripe_floor.size # regions bigger than MAX_SIZE are
removed
    #Pongo a cero todas las areas que no cumplan
    num_blobs_ok=0

    for i in range(0, num_labels):
        if stats[i, cv2.CC_STAT_AREA]>MIN_SIZE and stats[i,
cv2.CC_STAT_AREA]<MAX_SIZE:
            stats[num_blobs_ok, :]=stats[i, :]
            #print(stats[:, cv2.CC_STAT_AREA])
            num_blobs_ok=num_blobs_ok+1


    print('Number of labels after filter:', num_blobs_ok)

      #%% WIDTH
    line_left=0
    line_right=0

    #Decision for turning
    #Study each blob. If it is on the left side, increase line_left with its
width
    #At the end is possible to know where is the free way, at right or left
    for k in range(0, num_blobs_ok):
        if stats[k, cv2.CC_STAT_LEFT] < w/2:
            line_left=stats[k, cv2.CC_STAT_WIDTH]
        else:
            line_right=stats[k, cv2.CC_STAT_WIDTH]

    #Is there any big obstacle?
```

```
total_line = line_left + line_right

if w-total_line > w*0.3: #obstacle is bigger than 30% width, need to turn
    if line_left >= line_right:
        print('Turn left')
    else:
        print('Turn right')
else: #not a big obstacle
    print('Go forward')
```

ANNEX 4: Photos of the assembly on the robot

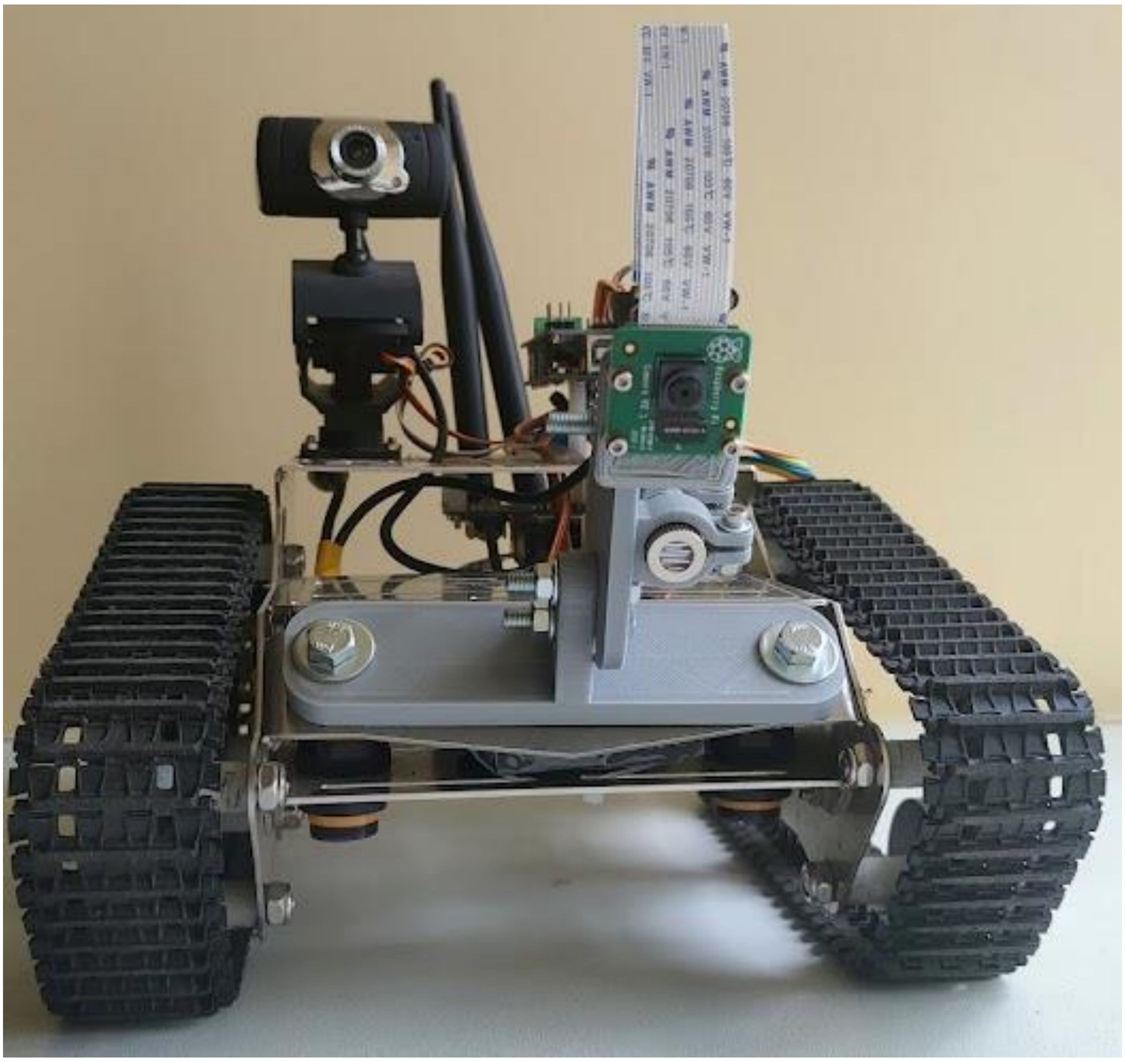


Fig. 43. Front view of the robot.

Fig. 44. Right view of the robot.

ANNEX 5: Bill of Materials.

Format	Area	Item No.	Mark	Tittle	Quantity	Note
				Document		
A1			MERS BM 22 01 01 00 00 GV	General View	1	
				Subassemblies		
A3		1	MERS BM 22 01 01 01 00 AD	Holder	1	
		2	MERS BM 22 01 01 02 00 AD	Laser support	1	
A3		3	MERS BM 22 01 01 03 00 AD	Camera support	1	
				Additional components		
		4	MERS BM 22 01 01 04 00	Red line laser 5mW	1	
		5	MERS BM 22 01 01 05 00	Raspberry Pi Camera Module 2	1	

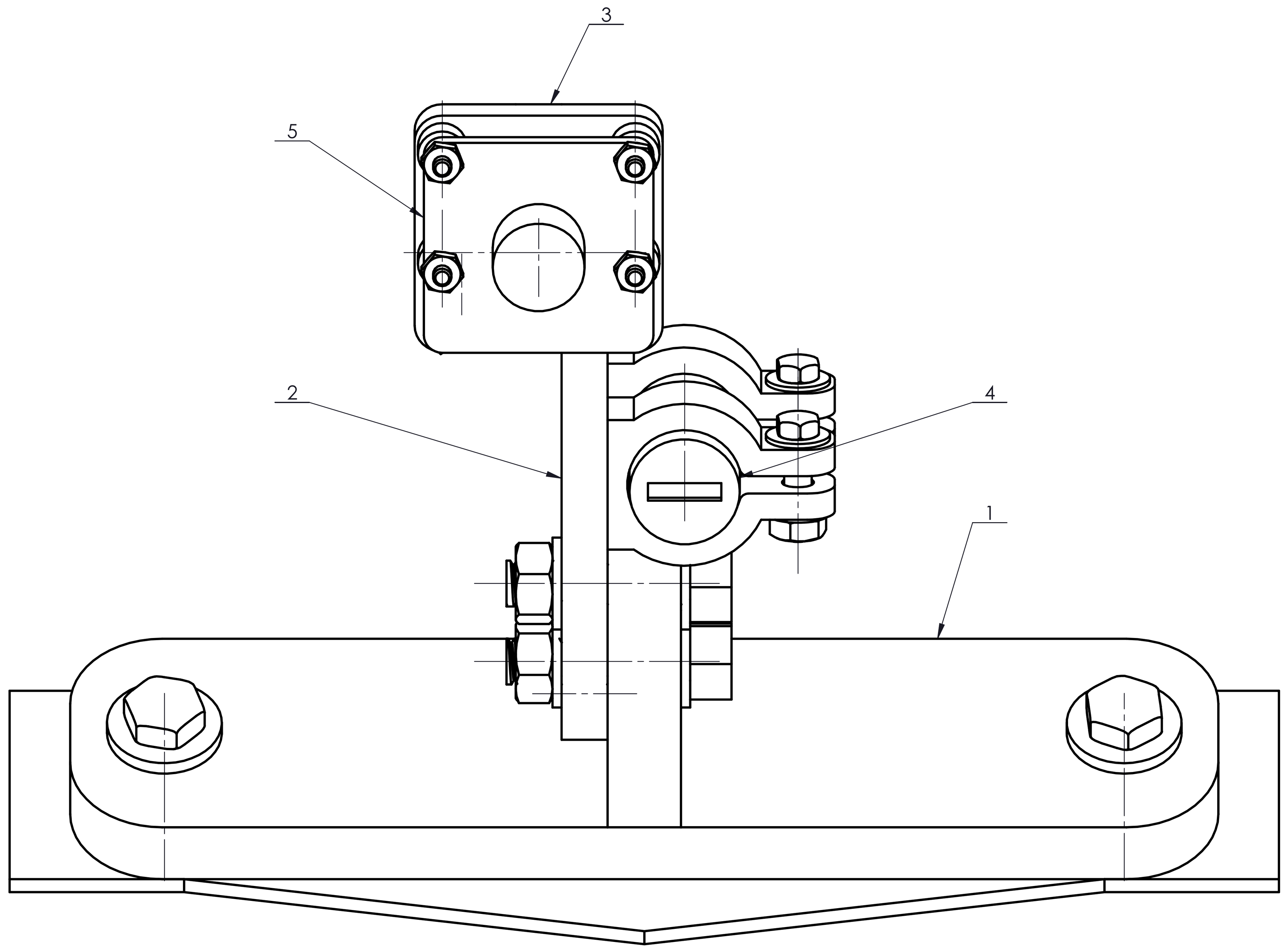
Responisible agency MERS dep.	Consultant	Document type Specification		Document Status Educational		
Owner VGTU MRfuc - 18	Compiled Eva de la Fuente García	Tittle Vision-based navigation system general view drawing	Mark MERS BM 22 01 01 00 00 GV			
	Checked		Rev A	Date 2022/05/12	Lang EN	Sheet 1

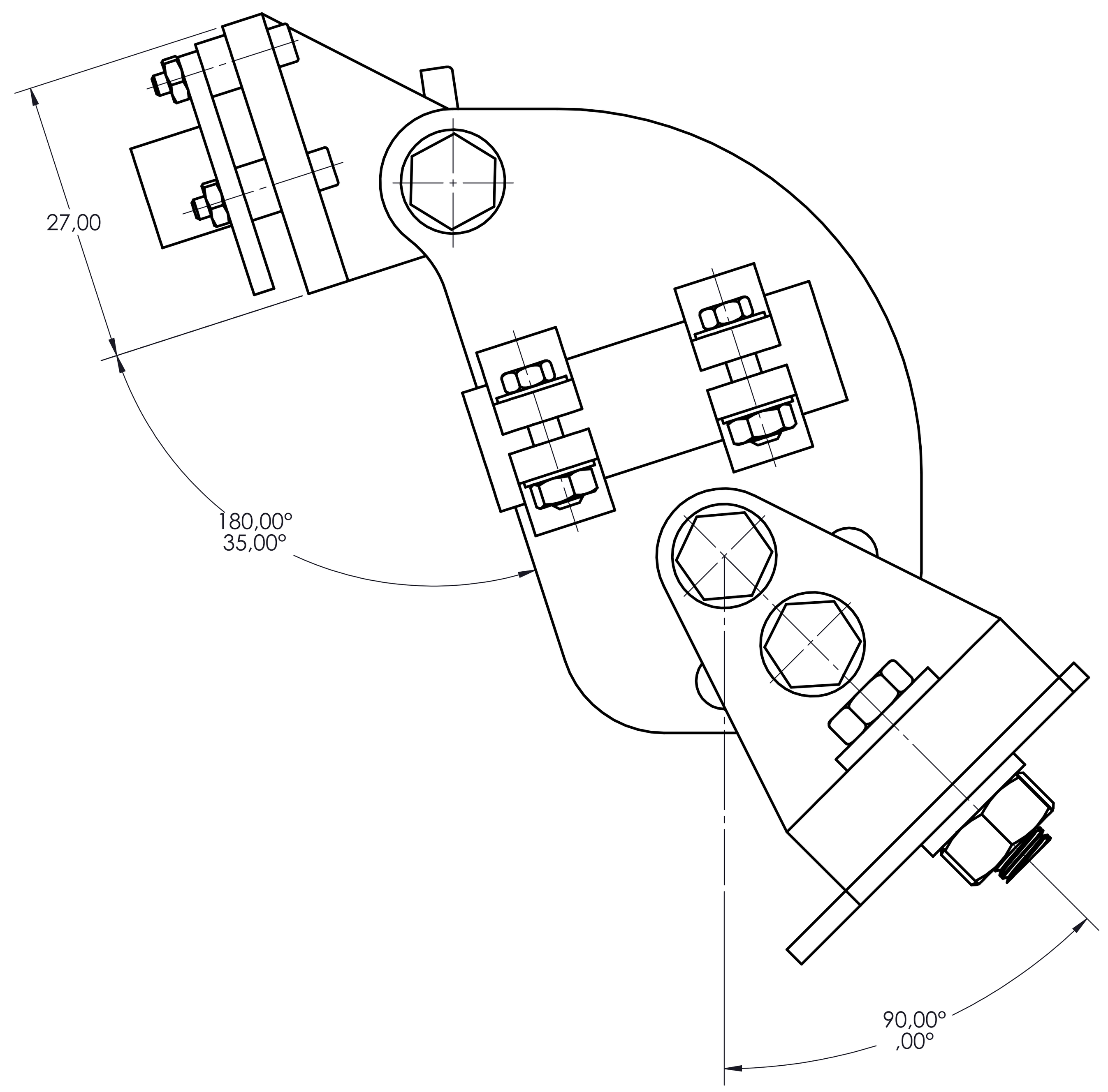
Format	Area	Item No.	Mark	Tittle	Quantity	Note
				<u>Document</u>		
A1			MERS BM 22 01 01 01 00 AD	Assembly drawing	1	
				<u>Parts</u>		
A3		1	MERS BM 22 01 01 01 01	Holder	1	
		2	MERS BM 22 01 01 01 02	Laser support	1	
A3		3	MERS BM 22 01 01 01 03	Camera support	1	
				<u>Standard components</u>		
		4	MERS BM 22 01 01 01 04	M3 Hexagon Nuts (DIN 934)	2	
		5	MERS BM 22 01 01 01 05	Washer ISO 7089 - M5	6	
		6	MERS BM 22 01 01 01 06	Hex Bolt M5 x 20mm (DIN 933)	3	
		7	MERS BM 22 01 01 01 07	M6 Hexagon Nuts (DIN 934)	2	
		8	MERS BM 22 01 01 01 08	Hex Bolt M6 x 20mm (DIN 933)	2	
		9	MERS BM 22 01 01 01 09	Washer ISO 7089 (DIN 125) - M6	4	
		10	MERS BM 22 01 01 01 10	M5 Hexagon Nuts (DIN 934)	2	
		11	MERS BM 22 01 01 01 11	M2 Hexagon Nuts (DIN 934)	4	
		12	MERS BM 22 01 01 01 12	Hex Bolt M2 x 12mm (DIN 933)	4	
		13	MERS BM 22 01 01 01 13	Hex Bolt M3 x 12mm (DIN 933)	2	
		14	MERS BM 22 01 01 01 14	Washer ISO 7089 (DIN 125) - M3	4	
		15	MERS BM 22 01 01 01 15	M5 Wing Nuts (DIN 315 AF)	1	

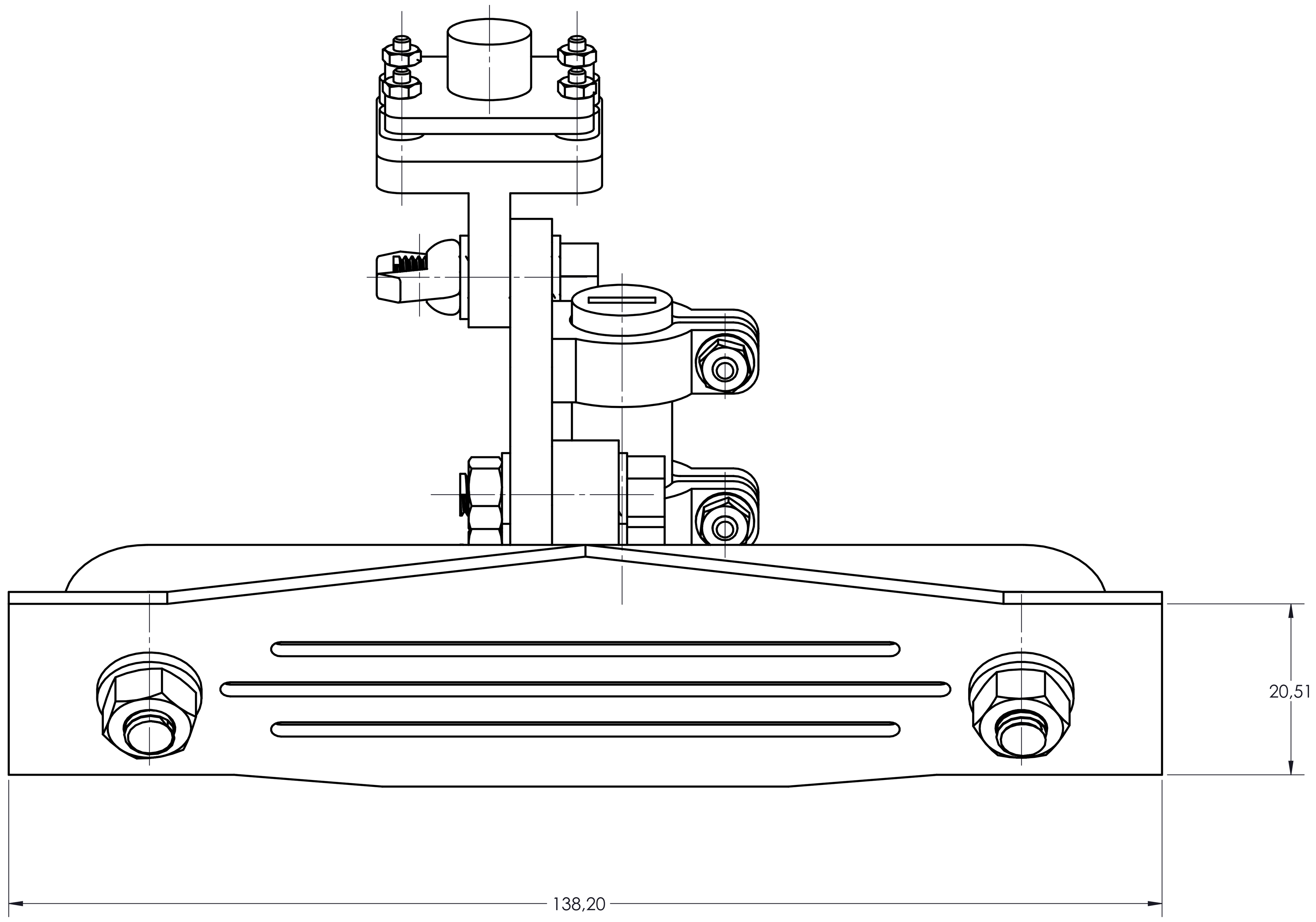
Responisible agency MERS dep.	*Consultant*	*Document type* ***Specification***			*Document Status* ***Educational***		
Owner *VGTU* *MRfuc - 18*	*Compiled* Eva de la Fuente García	*Tittle* Vision-based navigation system assembly drawing.	*Mark* MERS BM 22 01 01 01 00 AD				
	Checked		*Rev* A	*Date* 2022/05/12		*Lang* EN	*Sheet* 2

ANNEX 6: Drawings.

MERS BM 22 01 01 00 00 GV







Range of rotation of camera: 35º to 180º
Range of rotation of base: 0º to 90º

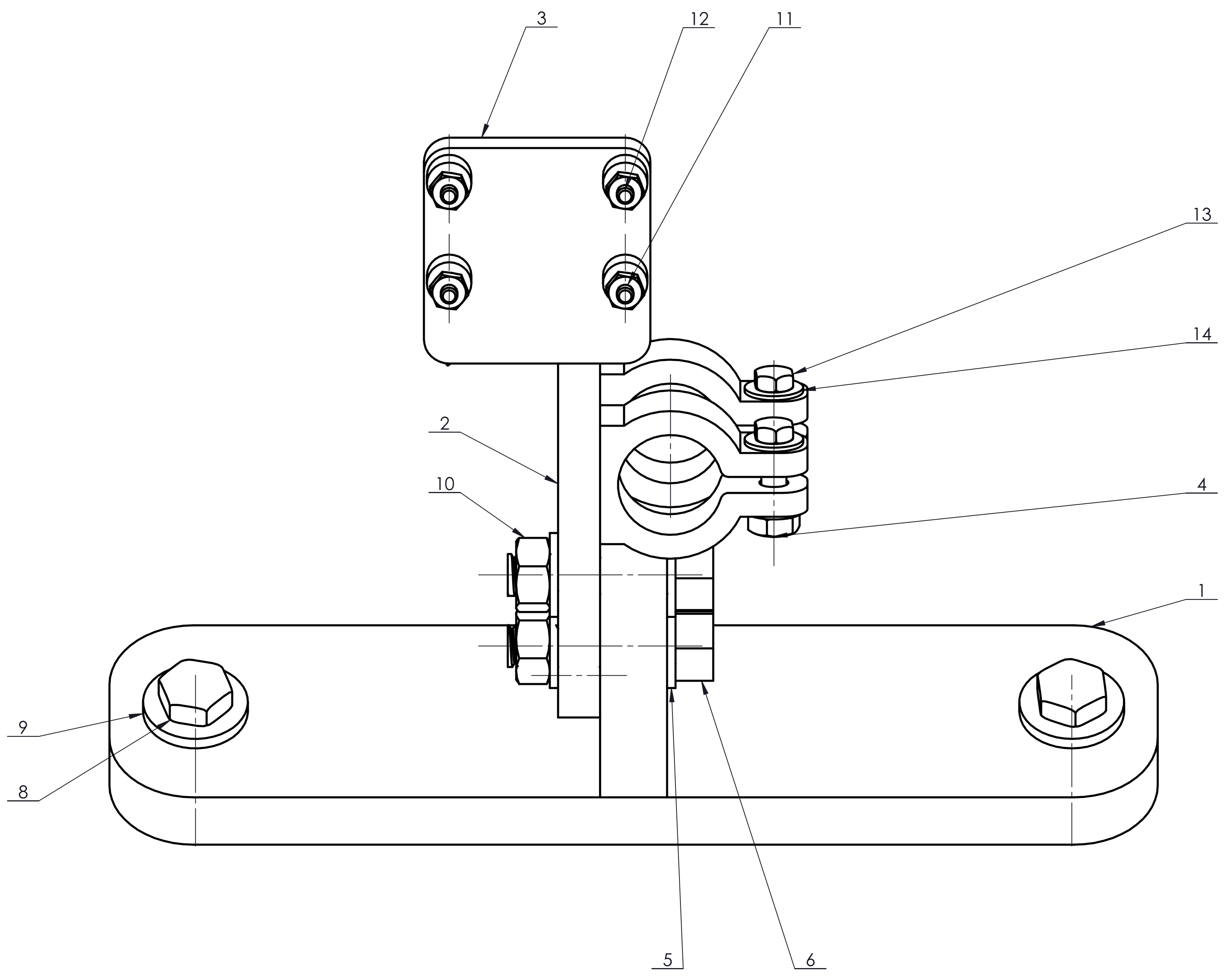
Image processing by Raspberry Pi 4B 4Gb.
Raspberry Pi 4B Voltage 5V DC

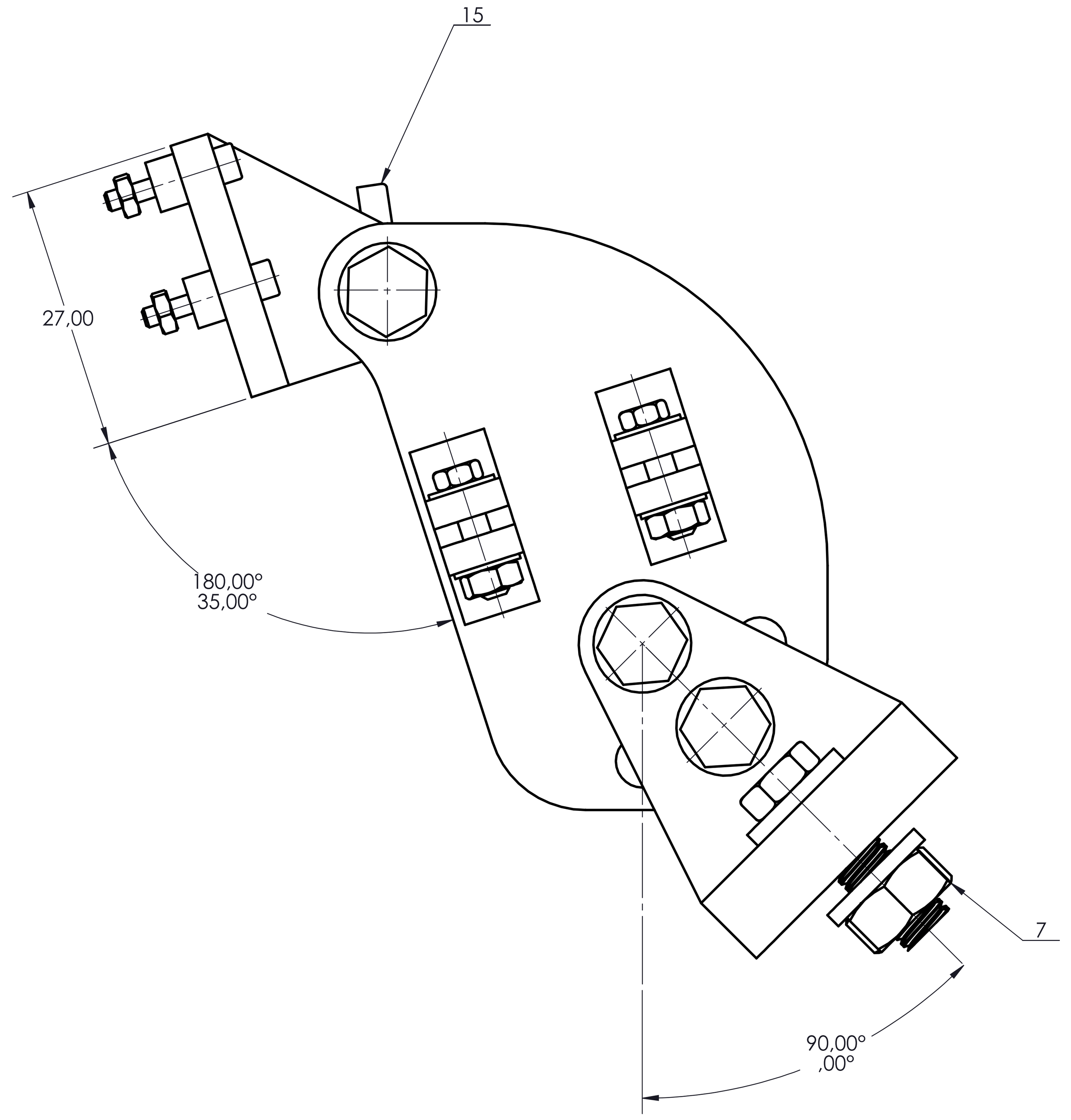
Camera: Raspberry Pi Camera Module 2.
Sensor resolution: 3280 × 2464 pixels.
Still resolution: 8 Megapixels.
Optical Size: 1/4".

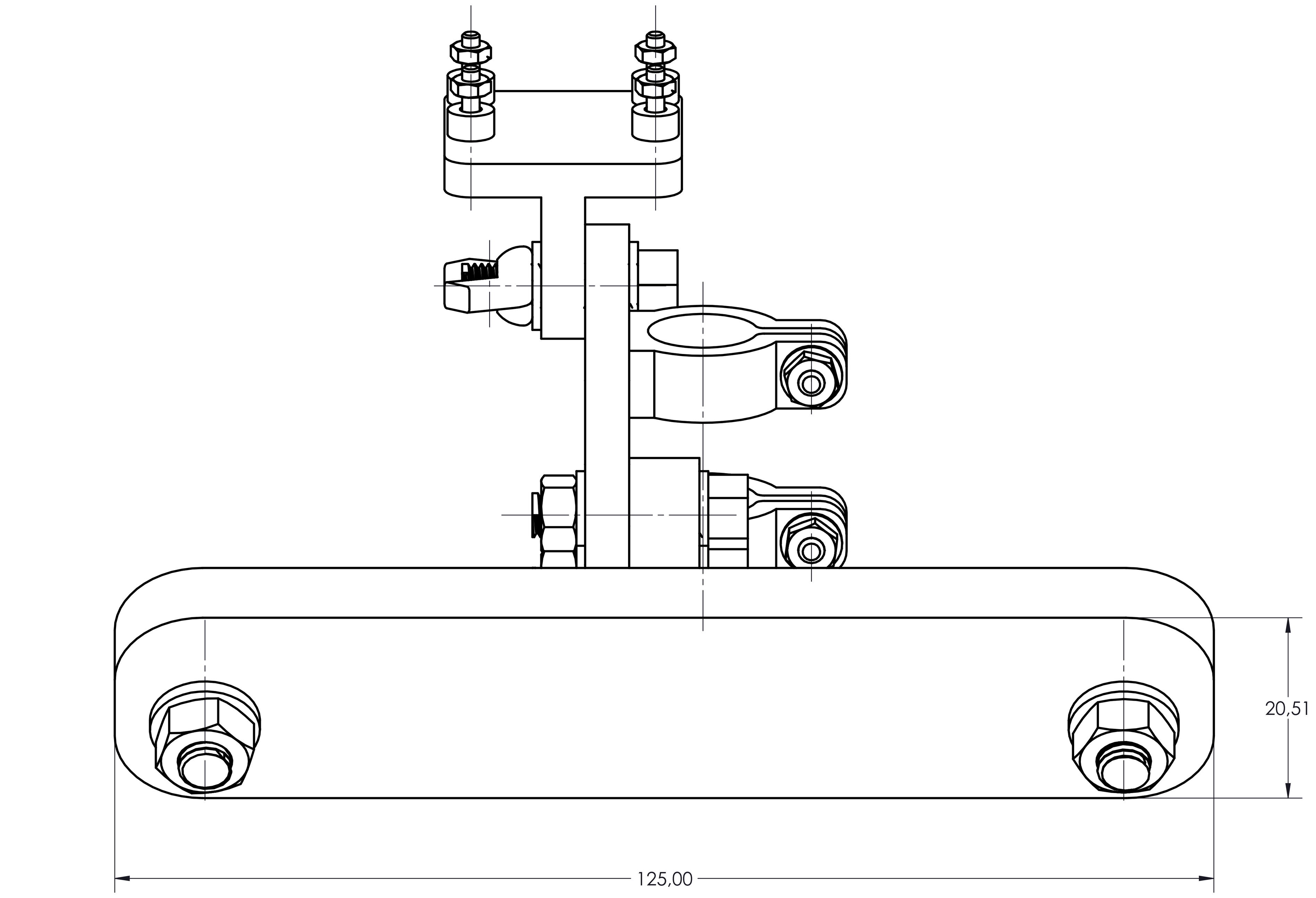
Laser: Red linear - Positioning laser.
Output power: 5mW.
Working voltage: 3 ~ 5V.
Working temperature: + 10 ~ + 40C

	Case No.	Additional information	Material	Scale 2:1
Responsible department Engineering	Consulted by	Type of document	Document status Educational	
Owner VGTU MRfuc-18	Drawn by Eva de la Fuente Garcia / Approved by	Title General View	Sign MERS BM 22 01 01 00 00 GV	Release A / Date 2022/04/21 / Language EN / Page 1

MERS BM 22 01 01 01 00 AD







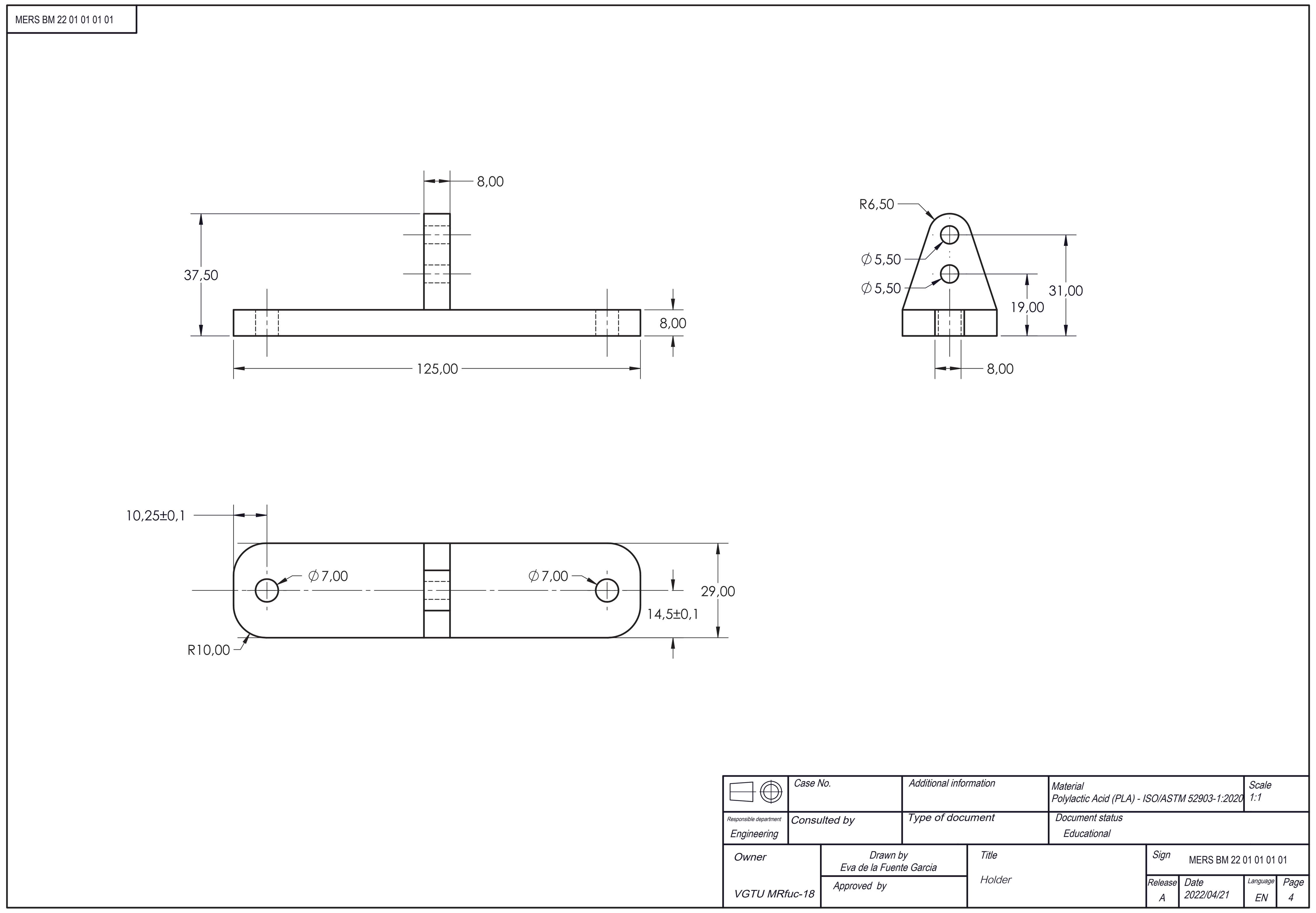
The camera must be fixed by the 4 metric bolts 2.
The angle of inclination of the camera must be chosen in such a way that it is able to capture the projection of the laser on the ground.

The laser must be inserted in its support and fixed with the two metric bolts 3. In the same way as the camera, the angle of inclination of the laser must be adjusted by means of the base-body union. This angle of inclination will be decided according to the distance at which you want to detect the objects.

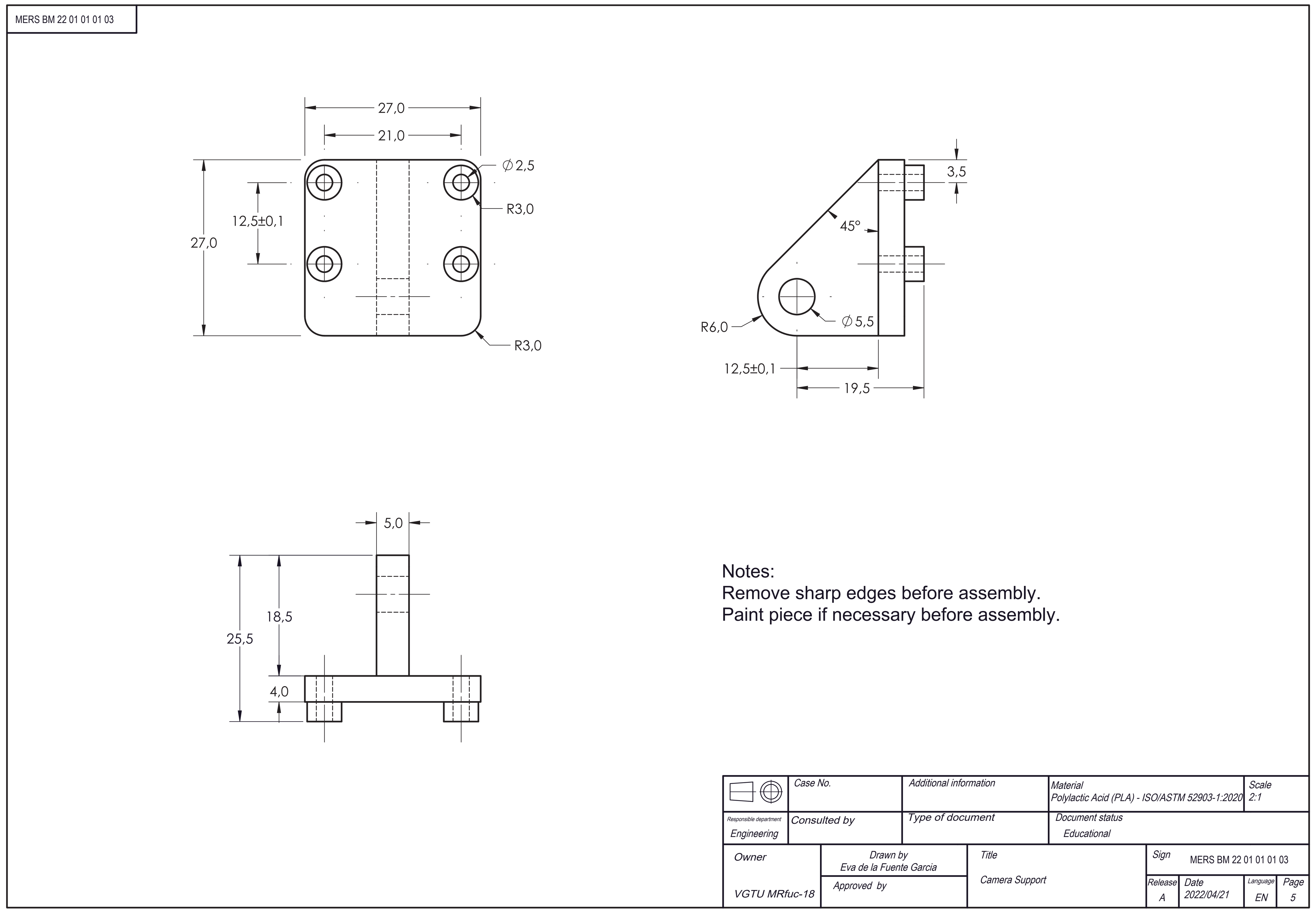
Robot base joint bolts: M6.
Bolts for base and body: M5.
Laser support bolts: M3.
Camera and camera support joint bolts: M2.

	Case No.	Additional information	Material	Scale 2:1
Responsible department Engineering	Consulted by	Type of document	Document status Educational	
Owner VGTU MRfuc-18	Drawn by Eva de la Fuente Garcia; Approved by	Title Assembly	Sign MERS BM 22 01 01 01 00 AD	Release A; Date 2022/04/21; Language EN; Page 2

Acquire data from camera
Segmentation laser beam and background
-Take red channel from RGB image
-Histogram for setting threashold
-Binarization
-Show binary image
Localizing laser projection on the ground
-Setting stripe of floor
Take laser projection width of interest
-Set and cut image side margins
Label blobs of image
Color each blob
Show blobs image
Print number of blobs
Size filter
-Define minimum and maximum area threashold
Go through all the labels
- cont = 0
cont == total of labels
Yes
No
Does the blob area meet the thresholds?
Yes
No
Save characteristics of the blob in an array.
Increase counter of filtered blobs
Increase cont
- cont ++
Print number of filtered blobs
Turning decision through estimation of concentration of laser beam at each side of image
- cont = 0
-left lines = 0
-right lines = 0
cont == filtered labels
Yes
No
Is blob on the left side of image?
Yes
No
Increase counter of left lines
- left lines ++
Increase counter of right lines
-right lines ++
Increase cont
- cont ++
Estimate total laser projection detected
- total lines = left lines + right lines
Are there big discontinuities?
Yes
No
Obstacle in the path
No obstacle in the path
Left line projection > Right line projection
Yes
No
No obstacle at left side. Turn left
No obstacle at right side. Turn right
Continue with trajectory
Case No.
Additional information
Material
Scale
Responsible department
Engineering
Consulted by
Type of document
Document status
Educational
Owner
VGTU MRfuc-18
Drawn by:
Eva de la Fuente García
Approved by
Tittle
Flowchart
Sign
Release
Date
10/05/2022
Len
EN
Page
3



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.