Master thesis

Industrial Engineering

Improvement of OWPT system by image recognition and machine learning

MASTER THESIS

Author: Sergio Pérez
Director: Noboru Yamada

Date: September 2022

国立大学法人
長岡技術科学大学
Nagaoka University of Technology

ETSEIB

Nagaoka University of Technology

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

This Page Intentionally Left Blank

ABSTRACT

New technologies for the future in reference to wireless energy transfer (WPT) highlight the optical path (OWPT). While its efficiency is relatively low compared to other types of WPT, the large distances that this type of system can support, and the safety of the devices compared to others (microwaves or large magnetic fields), make it attractive to develop and perfect.

One of the big problems with that is the aim towards a target, whether fixed or mobile. The efficiency of power transmission is directly proportional to how good the target recognition is and the system's ability to aim correctly. To solve this, this work develops the implementation of machine learning and image recognition through Yolov5 programs, OpenCV and Python programming.

The system consists of a structure where the transmitter is located, a halogen lamp in which it focuses directly on the recognition of the receiver, a toy car and later a radio control car. The system recognizes the car and radio control car thanks to a camera attached to halogen lamp. The halogen lamp moves on two axes thanks to two stepper motors controlled by an Arduino attached at the top of the structure.

Being moving objects, the comparison between several trajectory prediction systems is incorporated: Lagrange interpolations, Kalman filter and the absence of any predictive method.

The result is that the accuracy of the system depends largely on the speed of the receiver and that the base program developed follows up in a hurry and with less noise, which is tried to achieve through the Lagrange interpolations and Kalman filter systems, giving a slightly better result the latter.

On the other hand, the recognition of the receiver is perfected depending on the environment and the receiver in question. The efficiency of this recognition is proportional to the quality of the neural network's training to locate the receptor.

Summary

This Page Intentionally Left Blank

1. Forewords

Currently, all equipment requires electric power transmission for its operation. It is true that there is a lot of variety counting the types of wiring, the way to collect the energy or the batteries used, but all these methods have several problems to solve. If you solve the problem of energy transfer, society will be able to develop in new aspects. Methods that use energy transfer through light beams are being considered promising technologies. The potential for long-distance energy transfer without any electromagnetic interference is what makes this technology attractive.

These new technologies of energy transfer in an optical way, will be used in the future for a wide variety of applications, such as electric vehicles, robots, drones or energy transmission. Although the technical and scientific knowledge for this type of technology has been known for years, there are no practical or commercial systems, so research in this field is very necessary for the correct development of technology that could solve many of today's problems. Through research in materials, systems, subsystems, applications and the safety and standardization of processes and regulations is how this technology could be promoted.

2. Introduction

This project is the continuation of a previous project where the structure, receiver, and control system were stipulated. This second part seeks to improve the tracking system through Yolov5 software and prediction models.

Throughout the project it will be contemplated how to start improving the different elements that were already present and how most of the efforts are dedicated to achieving a prediction system with the available hardware elements. This will be achieved by experimenting with different filters and algorithms, coupled with a deeper understanding of the most optimal way to train a recognition system.

While it is true that it is a project that uses OWPT technology as a basis, it does not focus on it. The focus is on the tracking and prediction system, a key part of wireless energy transfer by the optical method. This is because although the emitter of the project is a halogen focus, it is known that there are more efficient ways to transmit energy, such as infrared or in the most demanding case at the tracking level, a laser.

In short, this work uses the tools of Yolov5, with open software in Python programming to precisely direct a hardware controlled by Arduino and stepper motors assembled to a halogen focus. The main point is the improvement of the software and achieve a neater and more accurate follow-up.

3. Structure

This project takes up a previous project in which a structure was already available that held the halogen bulb, the Arduino circuitry and the camera.

The main elements of this structure are made up of standardized aluminum profiles of different sections and lengths, joined by specific corners for each section. To all this, it is added a custom assembly of laser cut sheet metal that acts as a support and guide for the halogen bulb.

3.1. Initial structure

This initial structure consisted of two frames of standardized aluminum profiles joined together, with a support at the top with the halogen bulb outside the area comprised by the frames.

A structural analysis at first glance suggests that placing the main weight, which is also a moving element, away from the center of mass of the structure, makes everything vibrate more than desired, since the joints between the profiles were not perfectly rigid. These vibrations were translated into a camera image with more noise and less sharp, which caused measurement errors, which when corrected caused in turn more movement than necessary and therefore more vibrations.

On the other hand, the initial design had a really large base, which made it difficult to transport because of the dimensions and weight and made the whole preparation of the experiments more uncomfortable.

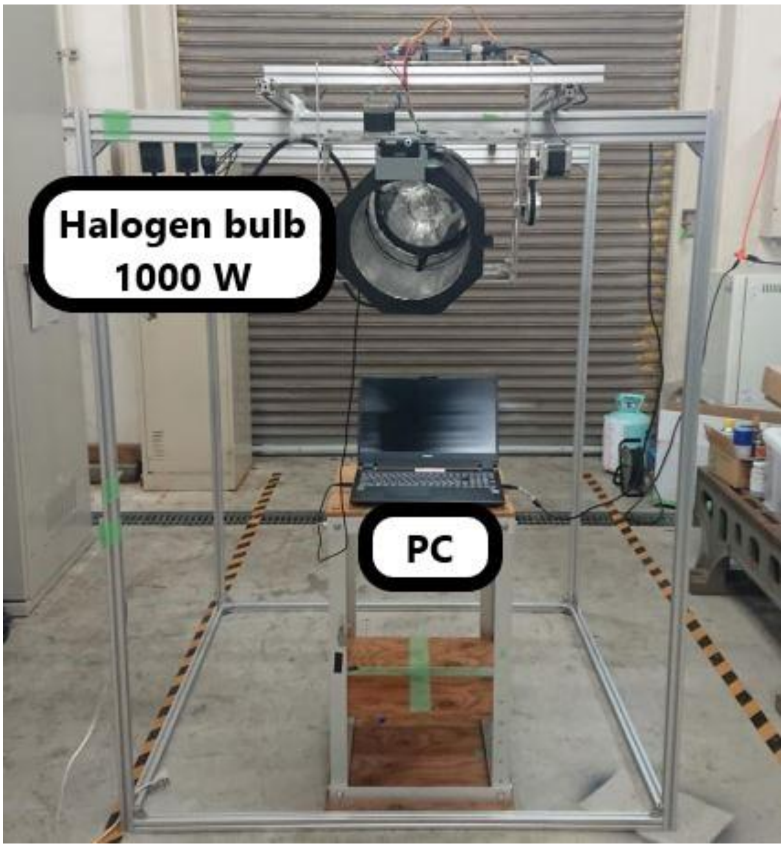


Image1: Original structure

3.2. Halogen lamp frame and elements

The halogen bulb of 1000W is mounted on pieces of galvanized steel sheet of 3 mm thickness that allow its rotational movement in 2 axes, as can be seen in image number 2.

Image2: Halogen bulb frame detail

On the halogen bulb has been placed a support designed in 3D for the correct annexation of the camera in charge of the recognition of the receiver.

Image3: Camera frame

The camera is an Elecom ucam-dle300tbk model of 30fps Full HD 1920×1080. It is a good enough camera for our application, although it is also true that an improvement in fps and vision quality could make a difference in operation in case the whole system is perfected enough.

3.3. Modifications

The main modifications of the structure previously described have been the following:

- Reduce the base of the entire structure achieving less weight and a more manageable size.
- Add crossbars and braces to the structure at key points to ensure greater rigidity despite the anchors between the aluminum profiles that make up the structure.
- Place the whole of the HL inside the structure achieving more stability and resistance to vibrations by being more vertically centered with the center of mass.

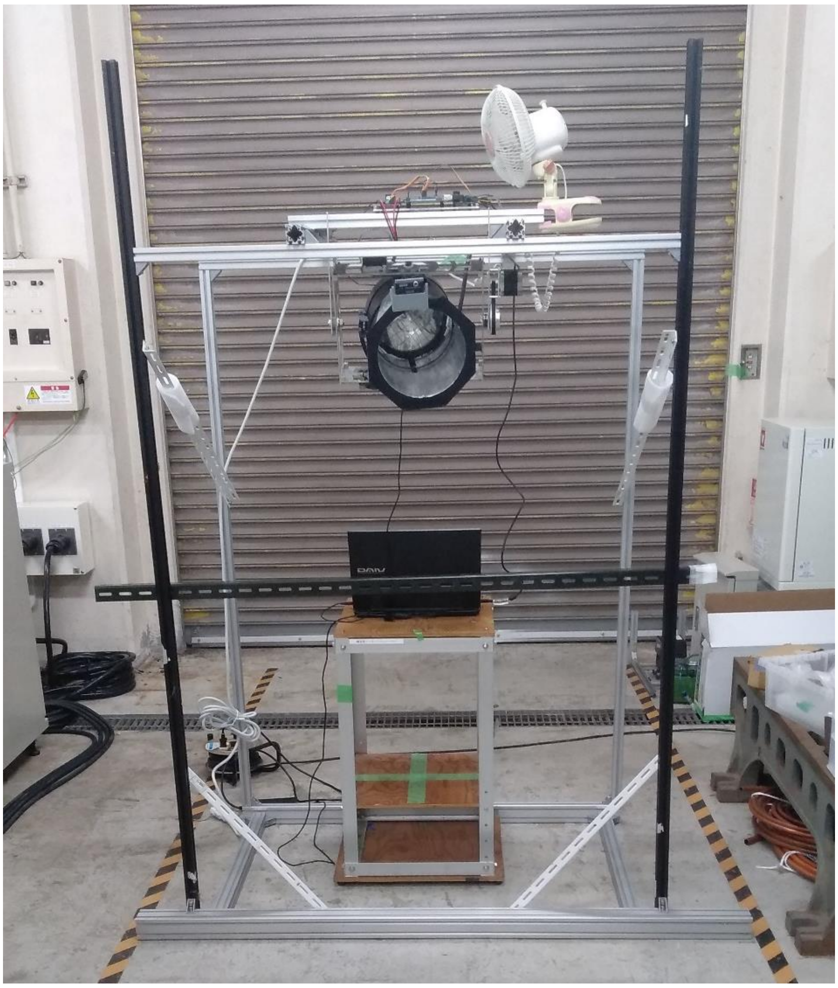


Image4: New structure

4. Arduino system

Arduino is an open source, affordable and simple platform that can perform the control of the stepper motors we need through a board with a programmable microcontroller.

4.1. Elements

For our circuit we will need the motors, controllers for these motors and an Arduino board. Next, the models used are defined.

4.1.1. L298N driver

This dual bidirectional motor driver is based on the L298 Dual H-Bridge Motor Driver. The circuit allows you to easily and independently control two motors of up to 2A each in both directions. Although in our case since the engines are 2.8A as we will see later, we will use two units.

It is ideal for robotic applications and very suitable for connection to a microcontroller since it requires very little programming line for its control.

This board is equipped with LED power indicators, on-board +5V regulator and protection diodes.

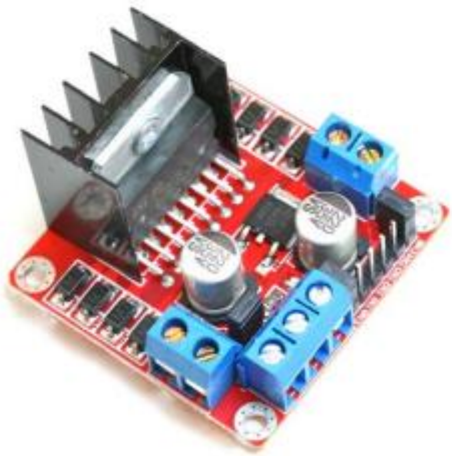


Image 5: L298N Dual H-Bridge Motor Driver

4.1.2. Arduino mega 2560

The Arduino Mega 2560 is one of the most complete Arduino models, a microcontroller board based on the ATmega2560. It contains everything necessary to support the microcontroller; simply connect to a computer with a USB cable or feed it with an AC-to-DC adapter or battery.

Image6: Arduino mega 2560

We will screw it to thetop art of the structure with separators to ensure proper cooling, as with the previous element, the drivers of the engines.

4.1.3. Stepper motors

Will use the model RS PRO Hybrid, Permanent Magnet Stepper Motor, 2.5 V, Unipolar, 1.26Nm Torque, 0.9°, 6.35mm Shaft.

St

Image 7: RS PRO Hybrid, Permanent Magnet Stepper Motor

4.2. Circuit

The circuit used is the original already present of the first part of the project. It is screwed to the top of the halogen focus assembly where the motors for movement are on boosters for cooling.

The Arduino connects through different ports to the two controllers that in turn connect the motors and allow their control in a safe way. The connection used is the one commonly represented in datasheets and multiple projects with stepper motors in Arduino using L298N drivers.

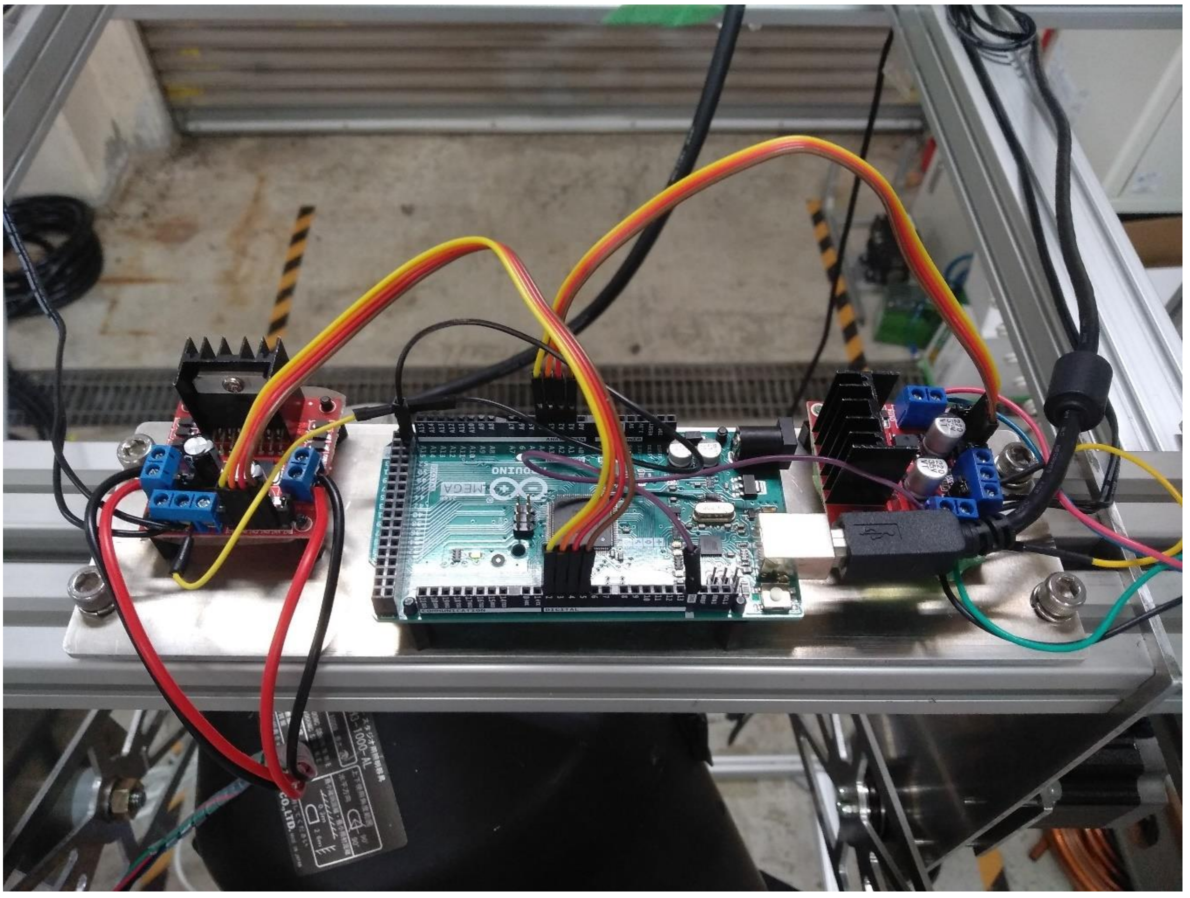


Image8: Circuit

Programming and feeding the system is done via the USB cable directly to the PC throughout the experiment. This allows changes to be made to the program easily and not using external batteries, giving more versatility to experimentation. It is also where the Arduino program receives the data calculated by Python as we will see later.

4.3. Program

The operation depends on the values read and calculated through the Python program that we will see later. These values are transformed to be sent to Arduino and processed.

The program used in Arduino can be divided into two parts. The first prepares the libraries, variables and parameters for their correct operation. The second is the main program that, when executed, will reproduce the movements we need.

The first part is the one shown in image 9. The image shows the code and comments explaining the use of each important line and its values

```
#include <Stepper.h>

const int stepsPerRevolution = 1200; // change this to fit the number of steps per revolution
// In our case, the stepper have 400 steps per revolution, but, due the reduction in the belt
// that connects the motor with the halogen bulb, the result is 1200 instead of 400

// initialize the stepper library on pins
Stepper myStepper1(stepsPerRevolution, 2,3,4,5);
Stepper myStepper2(stepsPerRevolution, A1,A2,A3,A4);

int Mvn_x,Mvn_y,stepper_no,stepper_angle; // initialize variables

int const stepper_count=2; // number of steppers that we use for the recognition

void setup() {

  Serial.begin(9600); // set the port

  myStepper1.setSpeed(10); // set the speed at 10 rpm
  myStepper2.setSpeed(10); // set the speed at 10 rpm

}
```

Image 9: First part of Arduino program

For the second part of the program present in image 10, the main program of operation is represented.

As can be seen in image 10, the value received from the Python program that we will see later, must be transformed so that it can be transferred, so once it reaches the Arduino program it must be transformed again. To this must be added that in the movement of the X axis (motor 201) the focusing distance does not increase, but by the position of 0 on the screen in the upper left, it must be multiplied by -1 so that it is directed in the right direction.

Pero on the Y axis (motor 202) the focusing distance if it changes with its movement, and is adjusted by increasing that value by 50%, so the data is multiplied by 1.5.

```
void loop(){
  if(Serial.available() > (2*stepper_count-1)){  // wait for connection of the a motor
    for(int i=0; i < stepper_count; i++){
      stepper_no = Serial.read();
        if(stepper_no == 201){                   // if the motor is the 201, X axis
          stepper_angle = Serial.read();
          Mvn_x = ((2*(stepper_angle))-200)*(-1);// The data recived have to be transformed*
           myStepper1.step(Mvn_x);               // The motor moves

        }
        if(stepper_no == 202){                   // if the motor is the 202, Y axis
          stepper_angle = Serial.read();
          Mvn_y = ((2*stepper_angle)-200)*(1.5);// The data recived have to be transformed*
          myStepper2.step(Mvn_y);               // The motor moves
        }
      }
   while (Serial.available() > 0){
     char t = Serial.read();
  }
    }
  }
```

Image 10: Second part of Arduino program

5. Receivers

The receivers of OWPT systems have a wide range ranging from fixed devices (mobile phones in a charging area, laptops, lamp in the living room, etc.) to mobile devices of greater or lesser speed (drones, cars, etc.).

The receivers are the element that must receive energy from the source, in our case HL. Normally these elements would have a small battery or capacitors to ensure operation in case they do not receive power for a few seconds, but in our case, we dispense with them to see clearly when this happens.

Our recipients during this project have been two; a steering-free toy car, running freely through a closed circuit and a radio control car with the same theoretical energy demand as the aforementioned receiver.

Both receivers have been fitted with solar cells on top in a structure designed and 3D printed by our ABS laboratory.

5.1. Solar cells

The solar panels are provided consisting of a wafer of monocrystalline photovoltaic material with marks for its partition into 4 pieces to later weld them in the desired arrangement. Each part of the wafer provides about 0.6/0.7 volts per unit, with a total of about 2 or 3 volts once welded in series. If welded in series a higher voltage is achieved, which is increased depending on the number of pieces welded in this way, while, if welded in parallel, more intensity is achieved.

The initial wafer is of a surface of about 15x15cm with chamfered edges and with the central indentations to weld one plate to another.

Image11: Complete photovoltaic cell

5.1.1. Manufacturing

We start by carefully splitting the wafer into the four predetermined pieces. It is prepared and welded in series with a strip of lead-free copper and helped with resin applied with brush for operation.

Once welded, the non-evaporated resin is removed and the set is carefully placed between two pieces of ethylene vynil acetate (EVA) and this between two anti-heat papers (such as baking paper) and placed in the vacuum machine that also applies heat and the following steps are performed in order:

1. The air vacuum is made on both sides of the machine and heated to 150º for 4 minutes
2. Vacuum is eliminated and kept at 150º for 9 minutes
3. Vacuum is applied again on both sides minus one atmosphere 4 minutes
4. The vacuum is applied only at the bottom, also less than an atmosphere 4 minutes

Once this procedure is finished, it must be allowed to cool for a few minutes and remove the baking paper very carefully, exerting a force as parallel as possible to the surface of the now molten assembly, since, if we apply a normal force to the surface, we run the risk of separating the EVA layer from the photovoltaic plate.

With this procedure we create 3 different sets with the same configuration to which we will perform performance tests later to determine which of all is the most efficient for use in this experiment.

5.1.2. Performance

First of all, it is checked that none of the cells have been broken during the previous procedure, since it is a very fragile material.

This is observed by applying current through the photovoltaic panel, which causes it to emit light in the infrared spectrum that we can capture thanks to a camera.

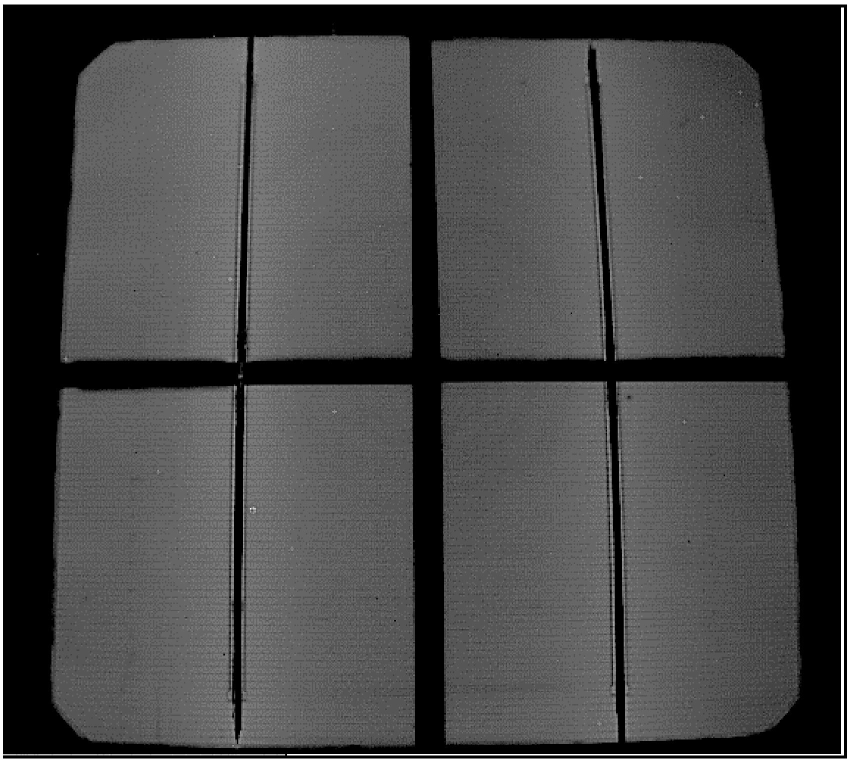


Image 12: Photovoltaic plate emitting infrared light showing that there are no cracks

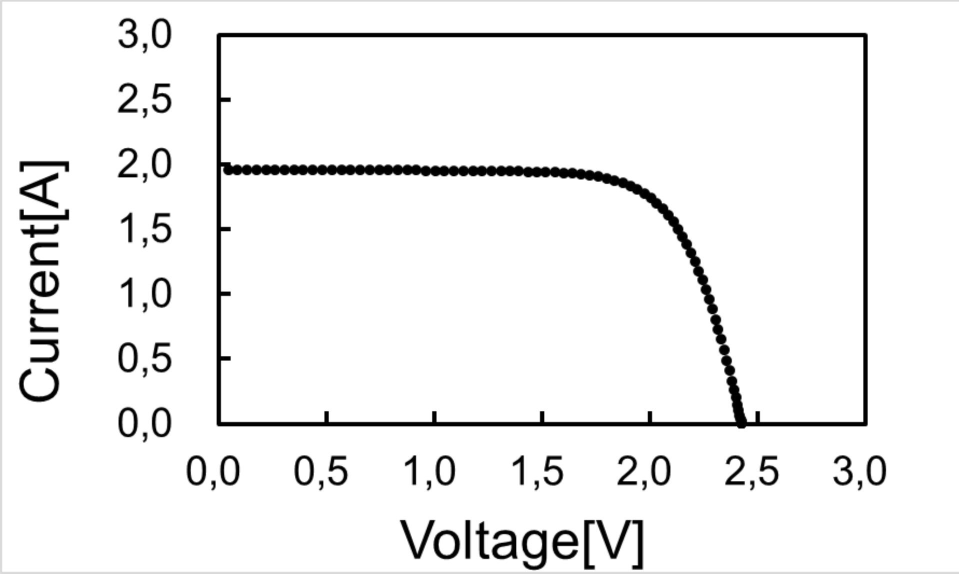
Once verified that the plates do not contain any dark part, that is, rotated in the procedure with IR, the standard performance test is carried out to generate the I/V curve, efficiency and fill factor of a solar panel.

To achieve these results, the test is carried out according to the procedure established in standards such as IEC 61215.

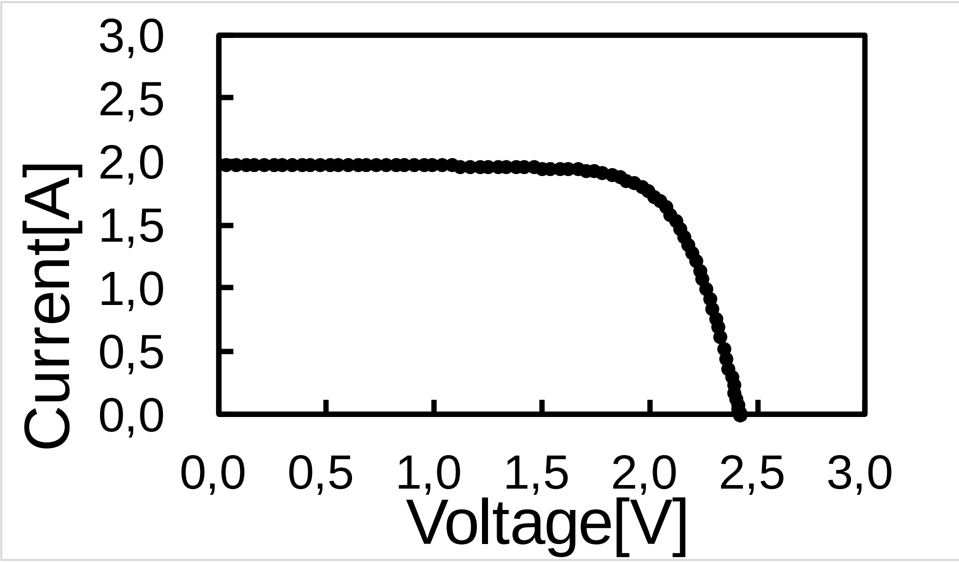
It consists of increasing the resistance of the photovoltaic panel from 0 to infinity while it is exposed to an irradiance of 1000 W/m2 and measuring the current and voltage. From this table are located the maximum, minimum and optimal values of both current, voltage and intensity.

Image 13:Solar panel testing machine

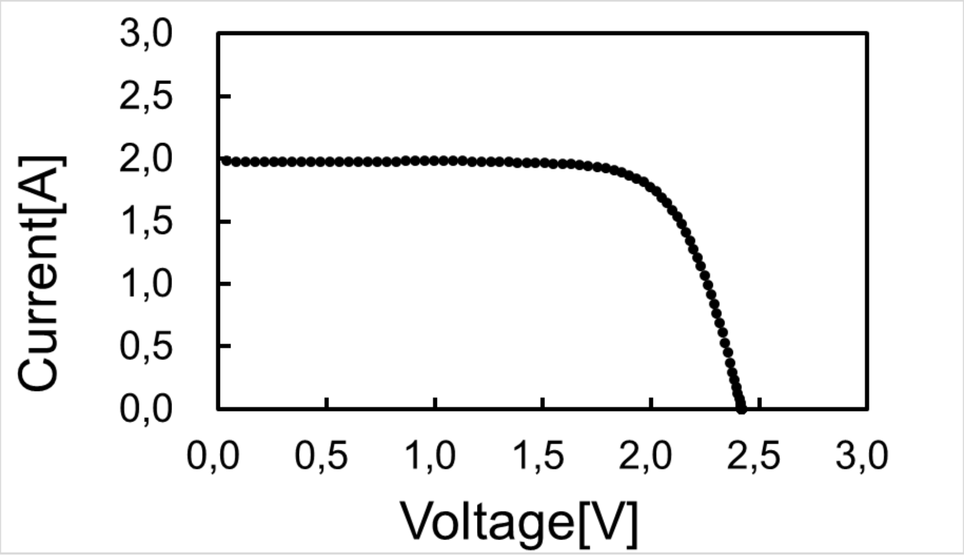
5.1.3. Results



Graphic 1: Photovoltaic cell assembly 1



Graphic2: Photovoltaic cell assembly 2



Graphic3: Photovoltaic cell assembly 3

Parameter		1st	2nd	3rd
Voc	**[V]**	2.42	2.42	2.42
Isc	**[A]**	1.96	1.98	1.97
Pmax	**[W]**	3.51	3.55	3.56
Vmax	**[V]**	1.94	1.96	1.93
Imax	**[A]**	1.81	1.81	1.84
efficiency	**[%]**	14.70	14.85	14.90
f.f.	**[--]**	0.74	0.74	0.74

Table 1: Performance values

5.1.4. Conclusions

Based on the above data and knowing that what we need in our experiment is a solar panel that is as close as possible to the values given by two standard batteries, the solar panel number 2 would be the most indicated, although it does not present too many differences between the other two plates.

5.2. Toy car

The first receiver used in this experiment is a toy car that runs along a pre-built circuit in the form of a loop and powered by two standard batteries that we replaced with the previously decided solar panel.

Image14: Toy car (redesigned)

5.2.1. Solar cells frame

This toy car was used in the first part of this experiment, and the first structure used was a single piece extruded vertically by 3D printing with four very thin supports and round section.

The fact of printing it in one piece, made the printing layers on the legs, were arranged in the most fragile way possible and because of a small blow it was broken before starting with this second part of the project.

To solve this, the design was changed in 5 pieces assembled with M3 screws and H7h6 lace tolerance:

- Base part where the solar panel will be added

- Four individual legs

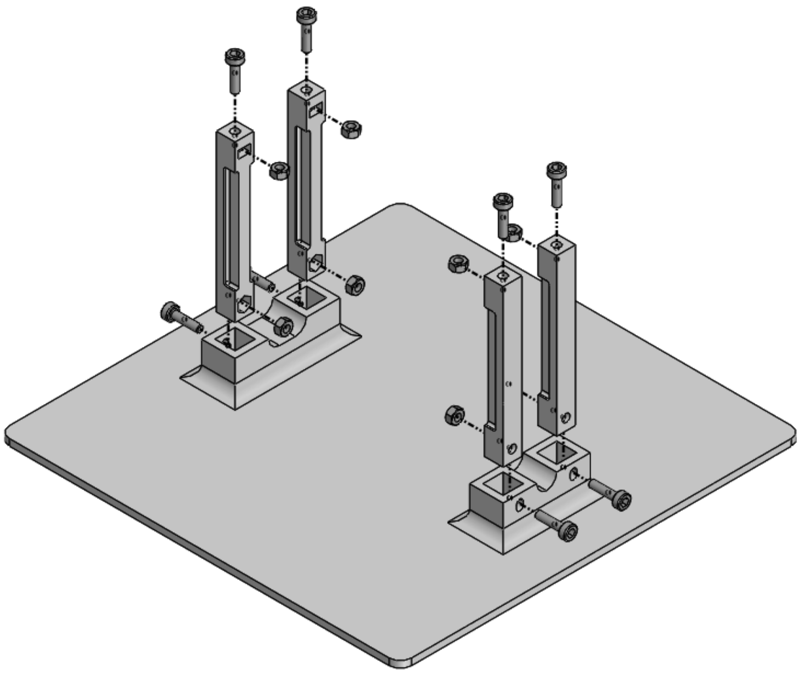


Image15: Toy car structure

With this new design the printing time was reduced and the resistance in each support was increased by its shape and by the printing in the optimal layer direction to withstand the stresses to which it will be subjected.

5.2.2. Problems detected

This toy car, being confined within a closed circuit, always made the same trajectory.

This adversely affected several points:

- The training necessary for the recognition of the receiver by the camera that we will see in the following sections, requires variety in the images to be processed in order to create a model with a high recognition accuracy. This system focuses on specialization, that is, the same receiver always doing the same thing, and a real implementation of this system would not be like that.

- Structurally, the mechanical tracking system always makes the same movements, or very similar. This prevents us from seeing if there is a problem with certain movements or in certain areas.

- The friction caused by the toy car with the walls in the corners of the circuit absorbed a lot of energy in certain cases, and a correct configuration of the circuit so that this did not happen was necessary. But even in the best position, the car crashed and caused it to brake losing inertia.

Image 16: Toy car loop circuit

5.3. Radio control car

Once the experiment with the previous receiver was carried out, we realized that a free trajectory of the receiver, instead of within a few lanes, would give us better data to improve recognition by artificial intelligence.

In addition, if a future implementation of this system is sought, this configuration would be more realistic and we could see if certain trajectories cause certain failures, system dead spots, etc.

For this we get a radio control car powered by two standard batteries that has two motors: one for steering and one for translation.

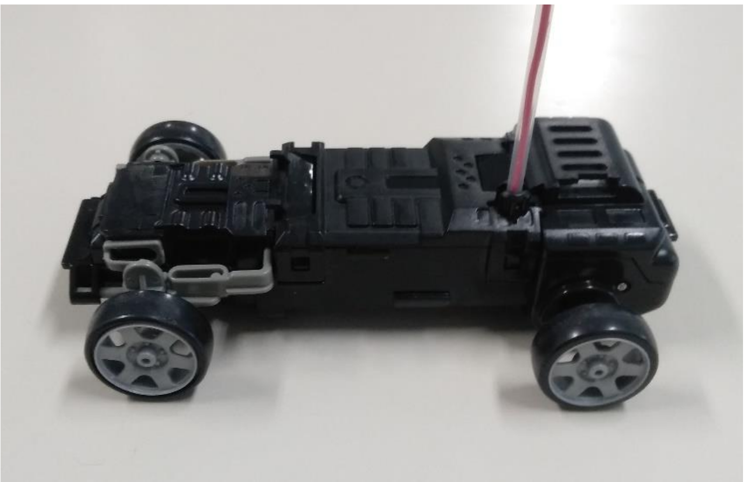


Image17: Radio control car

5.3.1. Solar cells frame

For the design of the solar cell support in this receiver, it was decided to reuse the previously used base and thus act more efficiently.

He only needed the design of the four support columns, embedded in a system of the sandwich type to the radio control car. All assembled with M3 screws and H7h6 sockets again.

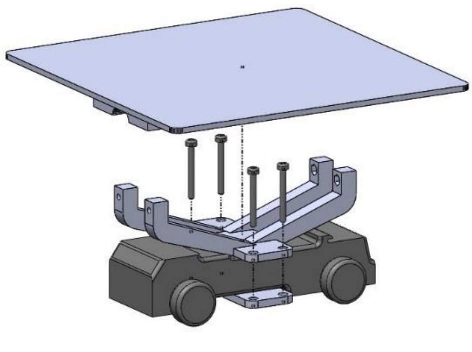


Image 18: Design of the support for the radio control car and photo of the current assembly

5.3.2. Problems detected

The first problem detected is that the energy transferred to the radio control car is not sufficient to drive the two motors available (steering and translation) at the same time at the maximum performance. This caused that at the time of changing direction the speed of the radio control car was drastically decreased, making the tracking system easier.

To fix that the steering system was blocked for experiments, making the radio control car circle at maximum speed.

6. Machine learning and image detection: YOLOv5

YOLO (You Only Look Once) is an open-source algorithm capable of detecting objects in real time. It does this through a single convolutional neural network that divides the image into different regions, where it predicts the pictures where the object would be located, and the probabilities that the object will be found in each region. In this way it learns generalities of the object and is able to detect it even with new inputs, not present in the training data set. [1],[6]

Yolov5 has several configurations of neural networks, with different nodes and degrees of accuracy represented in image 19.

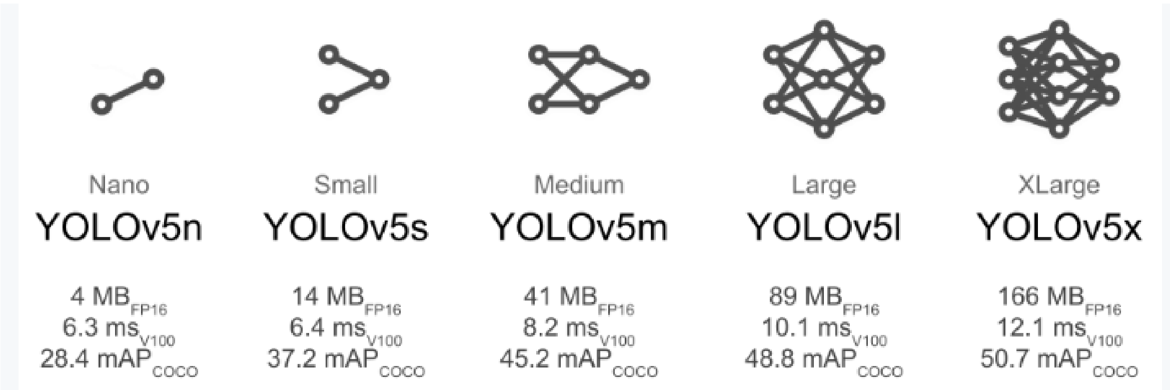


Image19: Yolov5 different neural models

In our case we will use in YOLOv5s, since it will be enough and does not require too much processing power.

6.1. Training Custom dataset

In order for Yolov5 to recognize the object, it is usually backed up by huge training databases of common objects, such as cars or road signs. In our case we want to recognize an object different from those present in those databases, so we must perform a personalized training.

6.1.1. Traces

To create our database for the recognition of our receiver, first of all we make a video of the receiver in motion in the environment in which we will perform the experiment. From that video, we usually get about 3000 frames and in each frame the receiver must be labeled correctly.

The program used for this purpose is Vott.

Vott allows the convenient and fast tagging of any video and the subsequent export of this data to different formats to make the subsequent training of the convolutional neural network easier. In the first cases we used the Vott version 1.7.2, which allowed the correct format for Yolov5.

Later we use 2.2.0, which does not allow export to that format, so we use Roboflow to obtain the correct format, thus discovering new possibilities in the management of tagged images.

6.1.2. Roboflow

Roboflow is an online platform with an immense community training for neural networks an endless number of different objects.

The platform allows you to tag your images and videos (just like Vott), but offers some especially attractive services such as:

- Preprocess images: Converts images to sizes and formats easier to process, so training is faster and more efficient.
- Data amplification: Through the process of rotating, changing luminosity or blurring the images, for example, it manages to multiply the number of images used in training and therefore its effectiveness.
- Own training: It allows you to train your neural network directly on the platform. The only problem is that it has a trial version of up to 50000 images and some limitations in the free version, so we use another method for training.
- Format conversion: You can upload your trained images in any format to the platform and download them in another format in the trial version without any limit.

Thanks to the conversion of formats, we can get the necessary one for Yolov5 and do the training. [2]

6.2. Training

Once we have the data of the images correctly labeled in the correct format, it must be used to train the convolutional neural network.

For this during this project two different methods have been used: through the Google Colaboratory tool and through programming locally on our computer.

For training you have to set three different parameters:

- Batch: It is the parameter that defines the number of packets in which the total set of tagged images used for training is divided.

- Epoch: It is a parameter that defines the number of times the learning algorithm will work across the entire training dataset.

  Each epoch can have one or more batches of the total data set, and each of them passes the number of times stipulated by the training.

- Img: It is the square size to which the images are scaled to perform the training in a more efficient way. At completion, the scale is reset for the original resolution. [1]

6.2.1. Collab

Using this Google tool, the free version, we could perform training to generate the "weight" file necessary for implementation in the system.

The method starts with the integration of Google Colaboratory with Google Drive. In this way we have access to the previously prepared files with the tagged images.

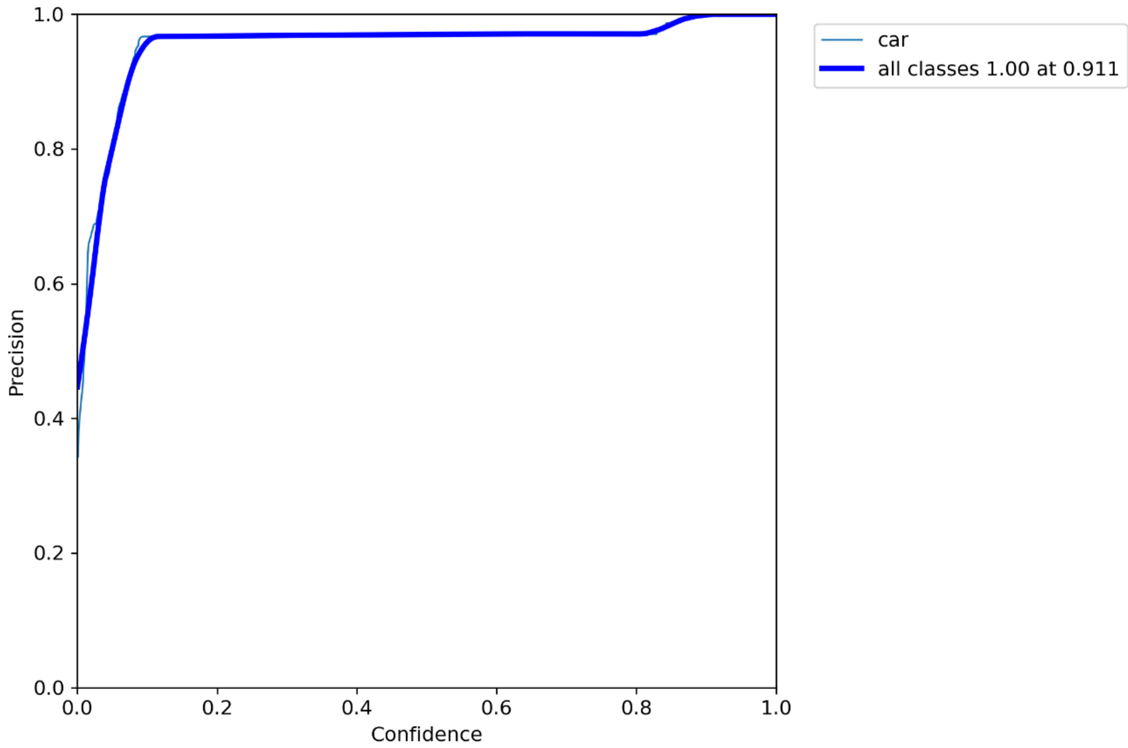
Then, a clone of the repository present on the Yolov5 github website is made and we install the necessary libraries.

With everything ready, you are ready to perform the training command, but you must stipulate the aforementioned data of image size, batch and epochs to use. In our case they will be the following:

- Img: 640
- Batch:32
- Epochs:60

At the end, a test is carried out with a video previously recorded for this purpose and the learning curves are generated.

One of the problems encountered with this method is that, if the workout has too many images and takes more than 4 hours, it stops automatically. This meant that I had to look for a new method without the time constraint, so prepare the laptop for local training.



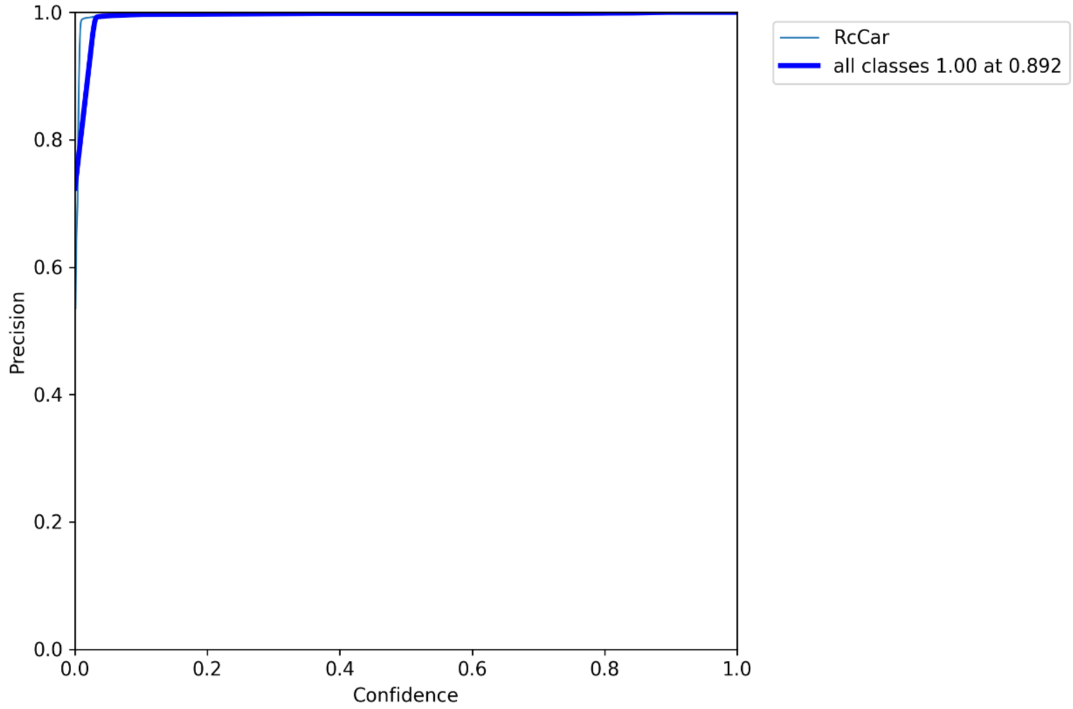
Graphic 4: Precision-Confidence Colab training

6.2.2. Local training using Roboflow

We went on to create a new video and extract from it about 3000 images that would later become about 12000 images. This video showed the radio control car running free and at higher speeds than the previous receiver. The improvement in the variety of images, both in distance, light and position, added to the increase in the number of images should mean an improvement in the recognition of the receiver.

This data size was not supported by Colaboratory, since the training lasted about 6 hours, so we created the training environment on the lab pc.

The results of this improvement can be seen in section 10, but ultimately, we went from errors of 12% to less than 1% in recognition.



Graphic5: Precision-Confidence local training

7. Python

The main programming language used is Python, although we also integrate it with Arduino for engine control.

Python is a powerful, simple and very versatile language for this kind of project. The large amount of information and open-source projects present on the network makes it easy to learn and solve the possible problems that arise throughout a project.

It has multiple libraries with specific purposes that adapt to our needs. As for example OpenCV (open computer vision), specialized as its name suggests in artificial vision, which together with the Yolov5 project of convolutional neural network will allow us to recognize the receiver in the camera and mark it. [3]

We will also use other libraries such as Pytorch, specialized in machine learning or classic mathematical libraries such as numpy.

To have all the necessary libraries and programs installed and updated we have a requirements file provided by Yolov5 that makes the necessary versions for the correct functioning of the program be installed directly on our computer.

7.1. Anaconda environment

Python typically uses modules, libraries, or packages that are not part of the standard library. Applications needs a specific version of a library, and that sometimes results in needing an already obsolete version.

This may mean that it is not possible for a Python installation to meet the requirements of each application. It may happen that two different programs or processes require the same module with different versions, making it impossible to implement them. This is where the virtual environment comes in.

This virtual environment is responsible for ordering, managing, notifying of updates and storing everything necessary for the virtual training of the program we need to use. Avoid conflicts between versions of the different modules. [4]

The chosen environment is one of the largest distributors of Python and other programming languages oriented to scientific computing: Anaconda.

7.2. Main program

In this section we explain the procedural operation of the main program written in Python. Step by step the important and conceptual processes that the program performs to make the entire system work.

The main operation is to square the central point of the camera image with the central point of the receiver's recognition rectangle.

1. First of all, the main program detects by Yolov5 and OpenCV the receiver in question and frames it within a green rectangle from which it extracts the central point and stores it in a list.
2. Determines the distance between the center point of the rectangle and the center point of the image read by the camera and stores it in a list.
3. A filter is applied taking into account those distances or values read in the previous points to predict the position of the receiver and that value is used in the next frame.
4. That value is modified to be formatted as compatible with transferring to Arduino and sent.
5. Arduino translates that value to the real one again and moves the engines accordingly.

8. Filters

By implementing the entire system and program explained in the previous sections, we wanted to increase the accuracy of the set without having to change elements. The options were mathematical filters to predict the position of the receiver in the following moments and thus advance the system to the movements, increasing the accuracy.

By predicting the position of the receiver in the following instants, the entire system can be anticipated. Although usually this anticipation is not totally accurate, it does cause the entire system to be positioned at a point closer to the correct position than if it were not using the prediction system.

8.1. Lagrange interpolation

The first position prediction filter used was Lagrange interpolation.

From the position of the receiver in the previous three frames, the position it will have in the next one is interpolated. In this way it was intended to soften the lines resulting from the reading of the distance from the central point of the camera, to the central point of the receiver. This will be seen more clearly in paragraph 10.

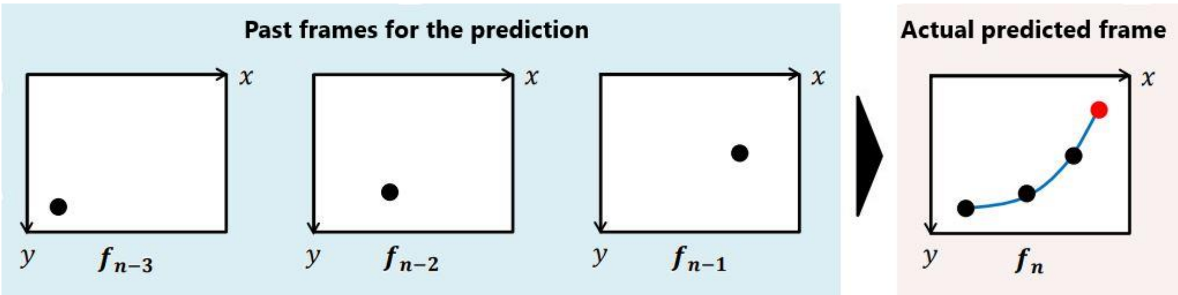


Image20: Lagrange interpolation frame representation

8.1.1. Lagrange main program

For the use of Lagrange interpolation in the predictive system in Python, the function to be used in the main program is first defined. The equation used is defined for the variable X and Y independently and is as follows:

$$L(x) = \left(\frac{x - x_2}{x_1 - x_2} * \frac{x - x_3}{x_1 - x_3}\right) + \left(\frac{x - x_1}{x_2 - x_1} * \frac{x - x_3}{x_2 - x_3}\right) + \left(\frac{x - x_1}{x_3 - x_1} * \frac{x - x_2}{x_3 - x_2}\right)$$

Equation1: Lagrange interpolation

Where $x_{1,}\ x_{2,}\ x_3$ they are respectively the first, second and third frames above.

The program predicts where the current frame will be and moves to it from the 4th program cycle, to have the necessary data for the calculation.

8.2. Kalman filter

The Kalman filter is a method for estimating non-observable state variables from observable variables that may contain some measurement error.

It is an algorithm that requires two types of equations: those that relate state variables to observable variables and those that determine the temporal structure of state variables.

It estimates state variables using its own dynamic. But it improves this to each cycle, using the information of the observable variables, it improves the prediction of the previous stages. In this way, the estimation of state variables uses all the information available up to that point, not as in the Lagrange interpolation described above. [7]

Luckily there are libraries and ways to integrate this into Python without writing all the equations in code. In our case we use the program made by Dr. Rahmad Sadli [5]

8.2.1. Kalman main program

The Kalman filter calculation program consists of a separate file from the main program where the necessary equations are defined based on some parameters necessary for its operation that we will give in the main program.

The calculation starts from the second frame, to have precedents for the calculation and needs the following defined variables:

- dt: sampling time (time for 1 cycle)
- u_x: acceleration in x-direction
- u_y: acceleration in y-direction
- std_acc: process noise magnitude
- x_std_meas: standard deviation of the measurement in x-direction
- y_std_meas: standard deviation of the measurement in y-direction

In our case the definition that improved the results the most is the following

dt	ux	uy	std_acc	std_meas_x	std_meas_y
0,8	0	0	4	0,5	1

Table 2: Kalman filter parameters used

9. Experiments

Once we finish the first draft of the program in Python, it is ready to do experiments to see what the operation is and finish adjusting some parameters.

The experiments are performed by placing the structure at a marked point with markings on the ground in which the camera has a correct view of a part of the ground free of imperfections that could interfere with the experiment.

As an initial protocol after placing the structure, a series of steps are performed to verify that each time the experiment is carried out, it is done in the most similar conditions possible:

- The lights are turned on and it is checked that the luminosity of the area is consistent, experiments are also tried at the same times of the day (usually at nightfall to ensure that the only source of light is the fluorescents in the room).
- The floor is swept away.
- The circuitry and cooling (a small fan) of Arduino are checked and that the wiring is fixed. The screws of the structure are also checked.
- Versions of the software are checked, as it is a shared computer.
- It is checked that the camera focuses correctly, getting the central point of the image to fit with the central point of the light cone emitted by the halogen focus, since it will be the point of greatest energy transmission.

9.1. Objectives

To perform the experiment, we must know what we are looking for. Our main goal is accuracy in the tracking system, as it is directly proportional to the transfer of energy to the receiver.

We are currently using a halogen bulb that easily covers the entire surface of the receiver, but the central area has a higher energy density so that is what we are looking for. In addition, if in the future the energy source were changed to a laser, we would no longer have the advantage of a large area of light, and the precision of the system would become even more necessary for the proper functioning of the system.

Initially, the use of capacitors in the receiver circuit was also considered, since, if for a short period of time the system did not reach to point correctly, the receiver would continue to work. This was ruled out as our current goal is to refine the tracking system, and without capacitors, we would be able to see more clearly when the system does not track properly.

The question here would be: What is proper follow-up? So, we decided, that, since the area of the solar panel in the receiver at the average distance of the experiment is approximately 100x100 pixels, to use as validation an area of 80x80 pixels.

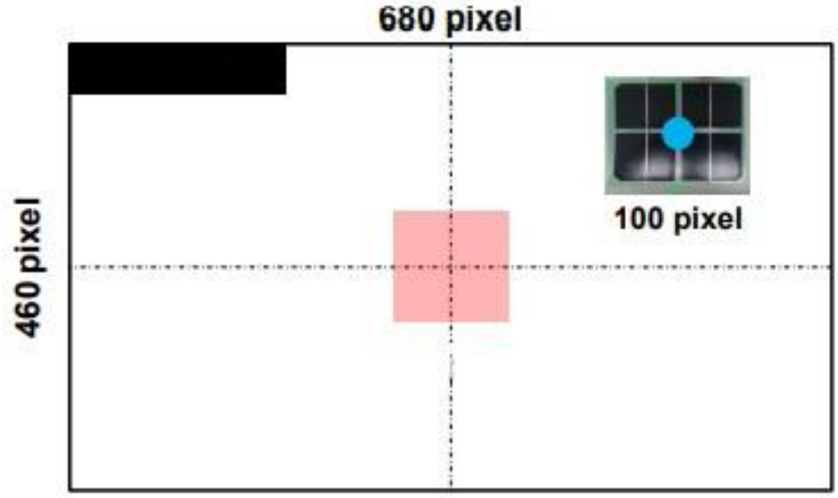


Image 21: Camera area, 80x80 zone in pink

9.2. Car

The first receiver to be tested is the toy car inside its tracks.

The tracks are placed centered on the structure and at a distance of about 30cm between the outside of the loop and the foot of the structure. The experiments are carried out with the car turning in either direction, and preferably by the external road, since it presents fewer problems when making the curves.

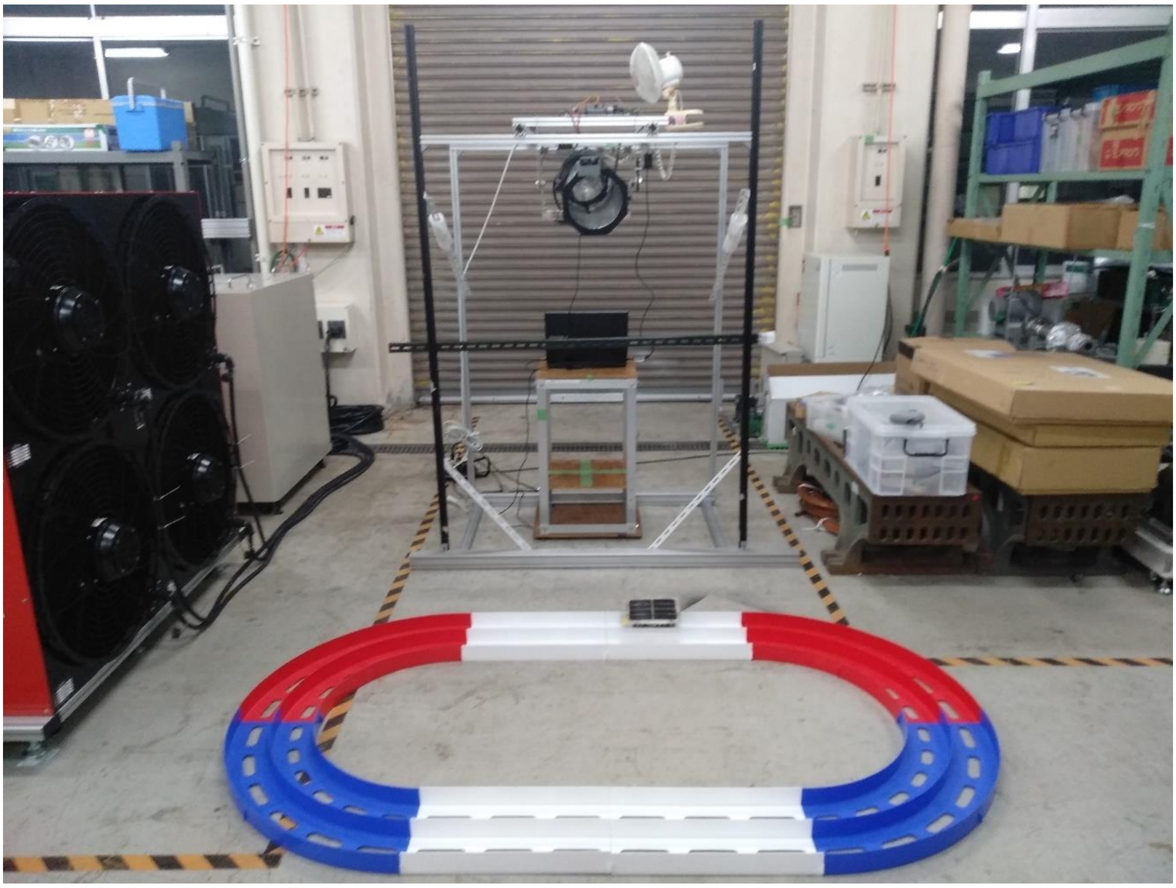


Image22: Car circuit

9.3. RC Car

The motivation to switch from one type of receiver to another is due to several factors:

- Having a car with a free trajectory would help to create a more complete database that would greatly improve the recognition system by Yolov5.
- It can be determined in which ranges of distance the system transfers enough energy for its operation.
- It is closer to the implementation of a real system, since the changes of direction in the trajectory are not guided by rails as in the previous assembly.
- The speed of the receiver would be higher, which allows us to be more demanding with the tracking system.

We encountered the problem that the energy transferred by the system, even over short distances, was not enough to move the steering motor and the translation motor simultaneously. So, the steering was blocked, causing the receiver to spin.

The car that will make circles is placed in the same position in each experiment, although tests have been carried out at different points of the surface within the operating margin of the system, to make a correct interpretation of the data collected, it is placed on a mark on the ground from which the car always starts.

10. Results

The first test that is carried out is with the toy car system in its guides.

The recognition system has been trained with a video of this same car in the same guides, so it is expected that the recognition will be good.

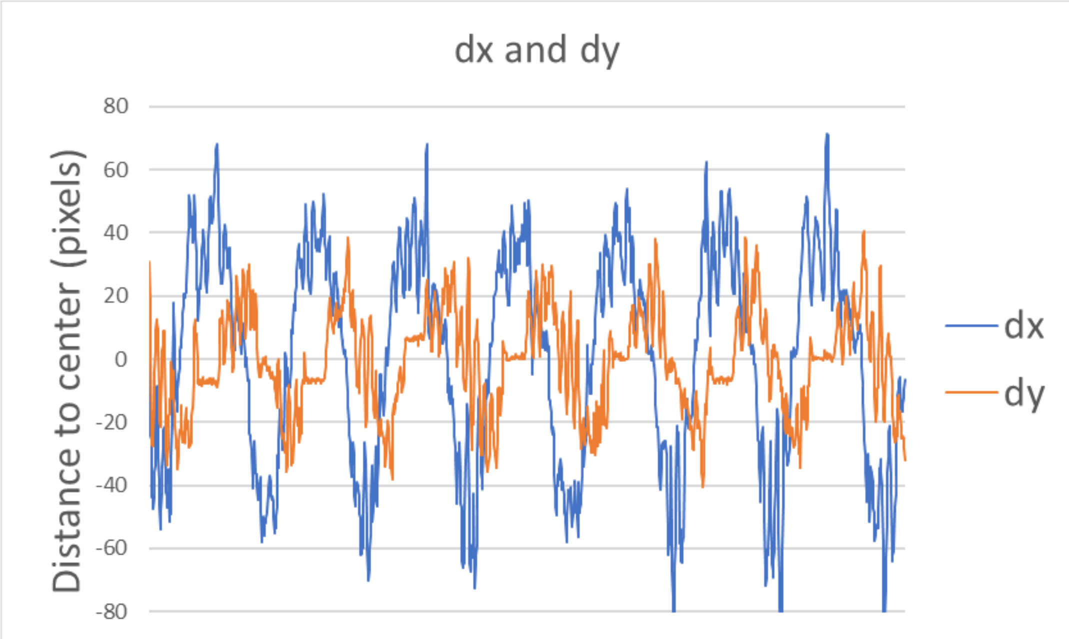
Once the experiments with the guided toy car are done, it is changed to radio control car and tested with the original AI training, but later it is perfected. So, it could be said that we have changes in several aspects during the experiment:

1. Toy car
    a. First AI training model
2. Car radio control
    a. First AI training model
    b. Second AI training model (new video)
    c. Third model of AI training (magnification by Roboflow) and engine shutdown system when the center is in a square of 5x5 pixels in the center of the receiver
        i. No filter
        ii. Lagrange interpolation
        iii. Kalman Filter

As mentioned in previous sections, the objective is to ensure that the tracking system remains within a square of 80x80 pixels, with the center in the center of the receiver and have a percentage of recognition as high as possible.

10.1. Experiment 1-a

In the first experiment there is a recognition failure in **3%** of the frames, it is a low value that we can consider correct, but in the future, we will try to lower it more.



Graphic6: Experiment 1-a

The graph shows the distance from the central point of the receiver to the central point of the recognition chamber. Ideally it should be as close to 0 as possible.

10.1.1. X-axis

The X axis has maximums close to 80 pixels away. We double the value considered acceptable.

There is also a lot of rattling, ups and downs in areas where the receiver does not make those kinds of movements.

10.1.2. Y-axis

In this case, flat areas are observed on the Y axis since the loop has an elongated shape as has been observed in the photographs, and during the straights, the Y axis does not make any movement. Here it can be seen that the Y axis is kept about 10 pixels away from the center in the most unfavorable case.

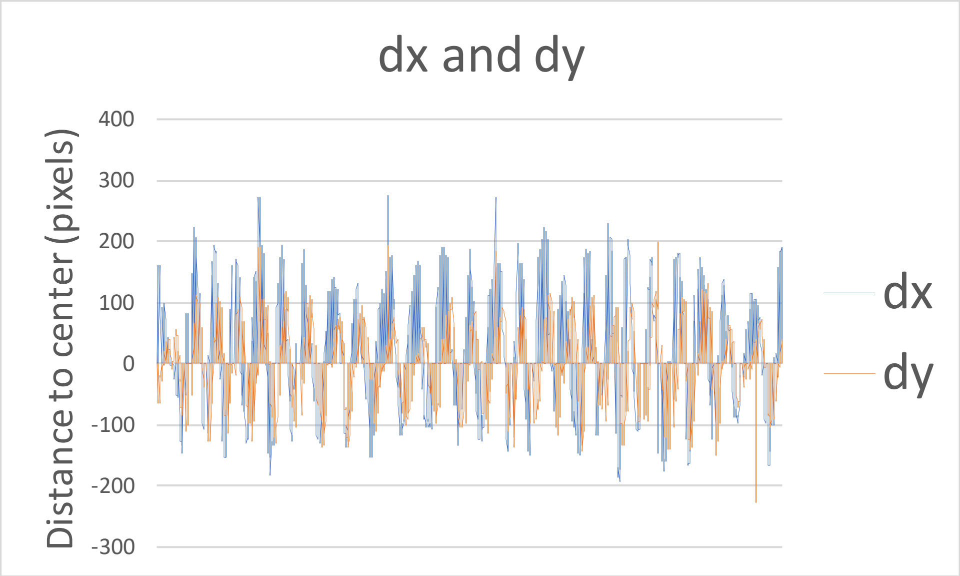
The peaks of the graph are located within the established safety zone of ±40.

10.2. Experiment 2-a

In this experiment there is a recognition failure in **42%** of the frames. This value is unacceptable, so in the next experiment we will change the training method and the files referring to the recognition of the receiver.

In each frame in which the receiver has not been recognized, the value of the distance to the center has been replaced by a "0", so in the following graph a practically constant line is observed in that value.

This may be due both to the recognition program and to the fact that the speed of this receiver is higher.



Graphic7: Experiment 2-a

10.2.1. X-axis

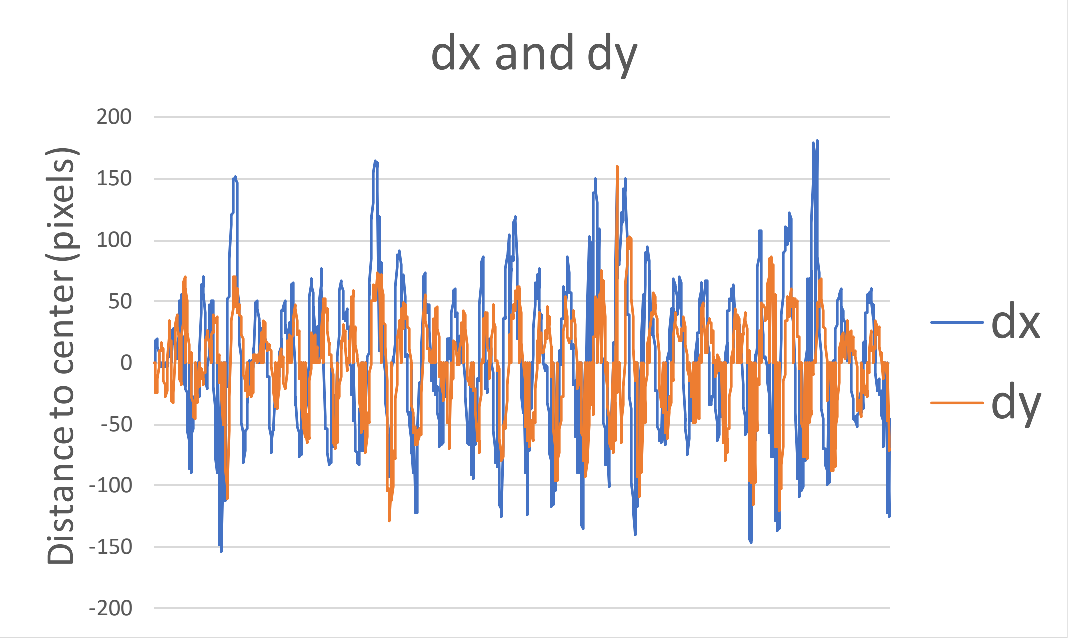
The X axis has peaks of almost 300 pixels away from the center with an average close to 200. This may be because the speed of this receiver is approximately 130% of the speed of the previous one and added to the recognition errors makes it almost impossible to track correctly.

10.2.2. Y-axis

The Y axis has peaks of more than 200 pixels away from the center with an average close to 100. This may be because the speed of this receiver is approximately 130% of the speed of the previous one and added to the recognition errors makes it almost impossible to track correctly.

10.3.Experiment 2-b

There is a recognition failure in **12%** of the frames. While it is true that it has decreased drastically compared to the previous one, it is still a very high value. As in the previous experiment, frames that failed in recognition are presented with a value of "0" in the graph.



Graphic8: Experiment 2-b

This improvement in the recognition system has decreased the overall maximum distance peaks and it can be observed that the higher peaks are preceded by time intervals in which the recognition system mainly fails.

This confirms that the improvement of the recognition system is key to the correct monitoring of the receiver.

10.3.1. X-axis

Although the improvement in the recognition system is not enough, changes can be observed for the better in the peak distances. While in the previous one the peaks practically touched 300 pixels away, in this one they have decreased, giving peaks that do not reach 200 pixels.

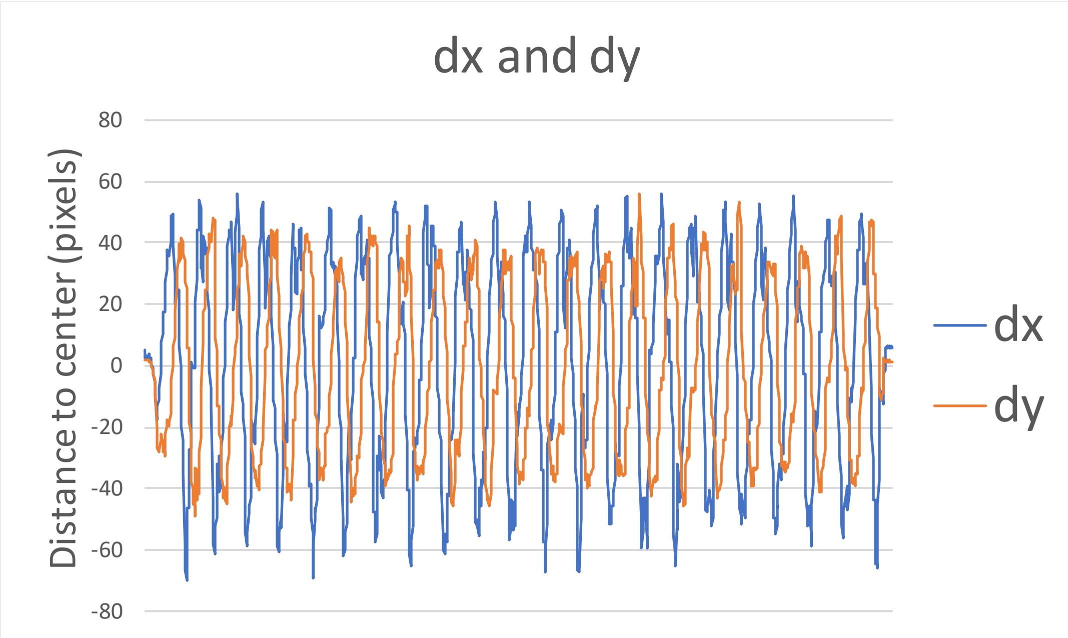
10.3.2. Y-axis

Now that the recognition system is a little more acceptable, it can be seen that although the car is making circles, the Y axis has a more accurate behavior than the X axis.

This may be because the movement in Y supposes a distance and approach of the camera, that is, that its movement supposes fewer pixels on average than the movement in X that does not suppose that change of distance.

10.4. Experiment 2-c-i

Recognition failure occurs in less than **1%** of frames. This value could be considered as fully valid.



Graphic9: Experiment 2-c-i

A much more stable graph is observed that recalls the values of the first experiment despite the fact that the speed of the receiver is higher.

By adding the engine shutdown system when it is in a central area, it means that when it is closer to the center, there is less vibration. This produces a substantial improvement in aiming when the receiver is still or moving at slow speeds.

10.4.1. X-axis

It is observed that the values of the peaks have been considerably reduced to values much closer to the objectives.

It is also appreciated that there is a tendency to have higher peaks on the negative side of the graph. This could mean that motor tracking when the receiver moves to the left of the screen is slower than when heading to the right.

Mechanical system and software checks performed have not yielded conclusive results as to why this may occur.

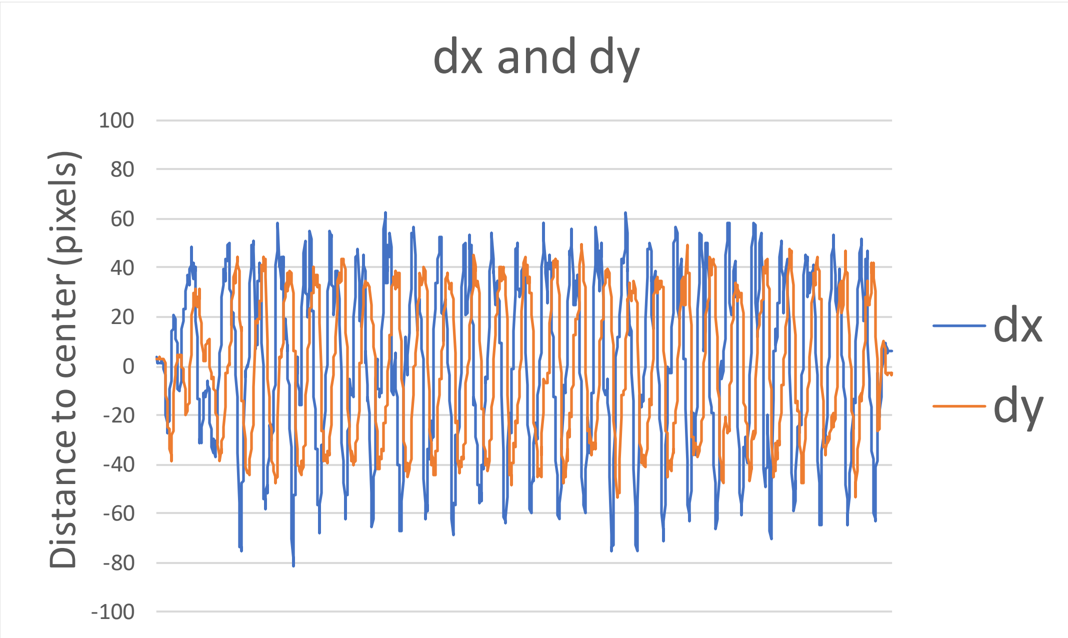
10.4.2. Y-axis

The peaks of the Y axis are much smaller overall than those of the Y axis possibly for the reason explained in the 2-b experiment.

The values of this axis are practically within the square of 80x80 pixels considered as a target.

10.5. Experiment 2-c-ii (Lagrange interpolation)

Recognition failure occurs in less than **1%** of frames.



Graphic 10: Experiment 2-c-ii

By implementing The Lagrange interpolation trying to predict the position of the receiver in the next frame, we get the system to vibrate more in the peaks and in the decision of the movement.

10.5.1. X-axis

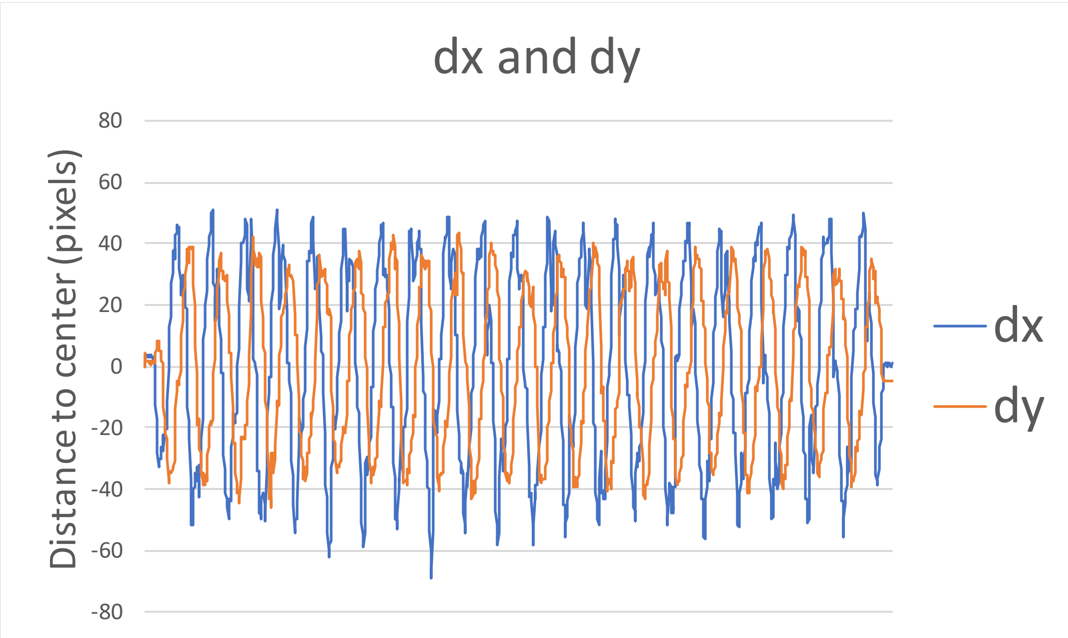
It continues to present the misalignment of the center, and the maximum values, although they remain at an average of 60 pixels away, there are peaks that approach 80 pixels.

10.5.2. Y-axis

This axis does not seem to have been greatly affected by the prediction system. The only observation that can be made is that the average of the most distant peaks is slightly higher.

10.6. Experiment 2-c-iii (Kalman filter)

Recognition failure occurs in less than **1%** of frames. This value could be considered as fully valid.



Graphic 11: Experiment 2-c-iii

When implementing the Kalman filter, it is observed that the graph is reduced compared to the Lagrange interpolation, creating lower peaks and vibrations in the less pronounced peaks. There is also a slight improvement compared to the experiment without any added filter.

10.6.1. X-axis

The peaks presented do not suppose a maximum of more than 70 pixels of distance to the center.

Although the misalignment of the center remains, it can be seen that it has decreased considerably compared to the previous experiments, implying that it is a problem solvable by software, at least to some extent, and not entirely mechanical as had been budgeted in the previous experiments.

10.6.2. Y-axis

Again, this axis is not greatly influenced by filters. Despite that, the small improvement means that the vast majority of peaks are within the safe zone of 80x80 pixels away.

10.7.Problems

As the experiments have been carried out, it has been possible to locate a series of problems to consider if they are recreated:

- The radio-controlled car with the direction fixed moves slowly as it circles. It must be taken into account and repositioned every time the experiment is repeated.
- The structure that fixes the halogen focus (along with the rest of the structure) is susceptible to loosening the screws, both when placing the entire structure and throughout the experiments. The correct tightening of these screws directly influences the ability of the system to follow the receiver, so you have to take it into account and pay attention.
- The fixing structure of the solar panel to the radio control car is misadjusted as it is used. It must be relocated to each experiment parallel to the ground. In the future, the structure should be changed to achieve a more optimal fit.

Conclusions

Throughout the project it has been observed that the prediction systems are necessary for a correct functioning of the set as it is currently proposed, since the response speed of the computer, programming and engines is not fast enough to carry out a hurried follow-up of the receivers.

The filter that has acted best has been that of Kalman, which together with the modification of stopping the system when it is within a square of 4x4 pixels makes it work correctly for both mobile and static receivers. Although there is still room for improvement.

A modification that could improve the accuracy of the system would be to switch to a fixed camera, since the image would not be distorted or there would be relative movement between camera center and receiver center.

Switching from halogen focus to infrared light could improve the amount of energy transferred. In case of changing the source to laser, it would also be necessary to change the solar panel of the receiver to one that responded well to the frequency of the laser.

Acknowledgements

I would like to thank Professor Noboru Yamada for his patience and trust throughout the project and for his kind words and guidance in the presentations. I also want to thank the whole laboratory for always being there and their instructions and aids for the use of machinery and facilities. Especially to Watanabe for his mentoring and guidance both for the project and on a personal level and to teach me the good gastronomy of Japan.

Bibliography

[1] YOLOv5 [https://github.com/ultralytics/yolov5, 22th of september, 2022]

[2] GETTING STARTED WITH ROBOFLOW [https://blog.roboflow.com/getting-started-with-roboflow/, 23th of september, 2022]

[3] THE PYTHON TUTORIAL [https://docs.python.org/3/tutorial/, 22th september 2022].

[4] MANAGING ENVIRONMENTS, USER GUIDE [https://conda.io/projects/conda/en/latest/user-guide/index.html, 22th of september, 2022]

[5] OBJECT TRACKING: 2-D OBJECT TRACKING USING KALMAN FILTER IN PYTHON [https://machinelearningspace.com/2d-object-tracking-using-kalman-filter/, 21th of september 2022].

[6] Object Detection using YOLOv5 and OpenCV DNN in C++ and Python [https://learnopencv.com/object-detection-using-yolov5-and-opencv-dnn-in-c-and-python/, 22 of september, 2022]

[7] KLAMAN FILTER [https://www.kalmanfilter.net/, 22th of september 20