Weeding Accuracy Improvement: Stereo Camera Metrology and Delta Arm Kinematic Calibration by Machine Learning

Autor: Rémi Porée

End-of-study internship report

20 august 2023

ENSTA Bretagne
Autonomous robotics
FISE 2023

Student:	Rémi Porée remi.poree@ensta-bretagne.org
Supervisor:	Ewoud Pool ewoud@odd.bot
	Luc Jaulin luc.jaulin@ensta-bretagne.fr

Acknowledgments

I want to express my sincere gratitude to everyone who made it possible for me to complete my internship. I especially thank my supervisor, Dr. Ewoud Pool, for his valuable guidance and insightful advice. He granted me autonomy while also being readily available to help. I also want to acknowledge my colleague and friend, Mr. Nivard Koenis, for meticulously proofreading my report and improving its clarity and precision.

Abstract

Odd.Bot weeding solution is made of visual detection and delta arm picking. In this paper, the system is studied to evaluate the errors' causes, and a method of accuracy improvement is proposed and experimentally demonstrated. A precise visual measurement is developed both for weeding and delta arm kinematic calibration, the latter being original by the uses of visual measurement and machine learning. Finally, results show that software can compensate for manufacturing inaccuracy and so improve weeding performances.

Keywords

kinematic calibration, vision-based metrology, Fiducial markers, machine learning, parallel mechanisms, computer vision, agricultural robotics, kinematic models

Contents

1 Introduction 4
- 1.1 Personal motivation 5
- 1.2 Context 5
- 1.3 Problem statement 5

2 State of the art 7
- 2.1 The weeding system 7
- 2.2 Overview of arm calibration 8
- 2.3 Measurement methods 9
- 2.4 Machine learning for identification 10
- 2.5 Chosen method 10

3 System modeling 11
- 3.1 Error identification 11
- 3.2 Parameters impact 11

4 Measurement 13
- 4.1 Coordinate measurement 13
- 4.2 Method of measurement 16
- 4.3 Camera position and results 16

5 Joint level calibration: Visual homing 18
- 5.1 Homing algorithm 18
- 5.2 Results 21

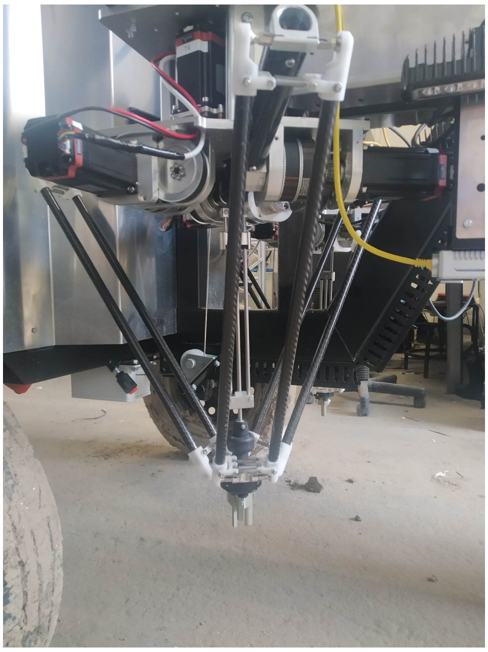
6 Kinematic level calibration: Machine learning 22
- 6.1 Model comparison 22
- 6.2 Data sets comparison 22
- 6.3 Results 23

7 Conclusion 25

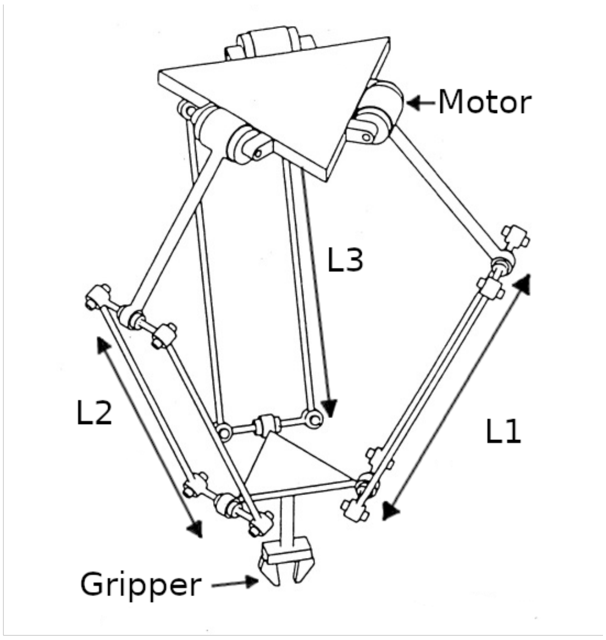
8 Appendix 26
- 8.1 Equations 26
  - 8.1.1 Inverse kinematic equation 26
  - 8.1.2 Forward kinematic equation 27
  - 8.1.3 Triangulation equation 28
- 8.2 Gantt diagram 28

1 Introduction

Odd.Bot is a company that needs to do mechanical weeding with high accuracy to compete and replace the current use of herbicides. For removing the weeds they chose to get inspiration from hand weeding by using visual detection and a mechanical arm. To localize the weeds, the camera needs to see in three dimensions, which is why Odd.Bot has chosen a stereo camera. For high productivity, this arm needs to be quick, which is why they chose a delta arm (figure 1a), which is mainly used in the industry for fast pick and place operations. In our case, the delta arm is either used to pick and throw weeds or push weeds down into the ground. Besides speed, the arm also needs to be accurate for perfect weed removal, accuracy is also needed for weed localization using a stereo camera. Unfortunately, the current accuracy of the system can cause it to damage crops or miss weeds, the system's precision needs then to be improved. That is my mission here: improve the weeding accuracy.



(a) Delta arm installed on the robot.



(b) Delta arm scheme.

However, the delta arm can only be moved accurately if the dimensions of each arm are precisely known, but each arm is unique due to manufacturing tolerances and assembly errors. For example, in figure 1b the length of L1 is not perfectly known, and can be slightly different from L2 and L3. Then, higher accuracy would require the accurate determination of these dimensions.

Due to the same issues, neither the position of the camera is perfectly known. These manufacturing tolerances add to measurement inaccuracies leading to weeding errors, even with precise AI weed detection. The accuracy improvement should then focus on more than the delta arm, by integrating camera position determination and camera calibration evaluation.

Additionally, for moving with high accuracy, the starting angles of the motors have to be precisely known. For now, it is measured with a dedicated position sensor on each motor. But the placement of these sensors can't be done with accuracy due to manufacturing inaccuracies, which causes errors in the measurements. So for avoiding these errors, improving accuracy, and reducing the complexity and the cost of the system, it can be beneficial to find another way to measure the starting angles of the motors.

This report will describe how to calibrate the system by finding accurately the dimensions of the arm and the position of the camera, evaluating the measurement, and determining the initial angles of the motors. However, first, this section will introduce personal motivations, the context of this work, and the goals of this study. It ends by giving an overview of the entire paper.

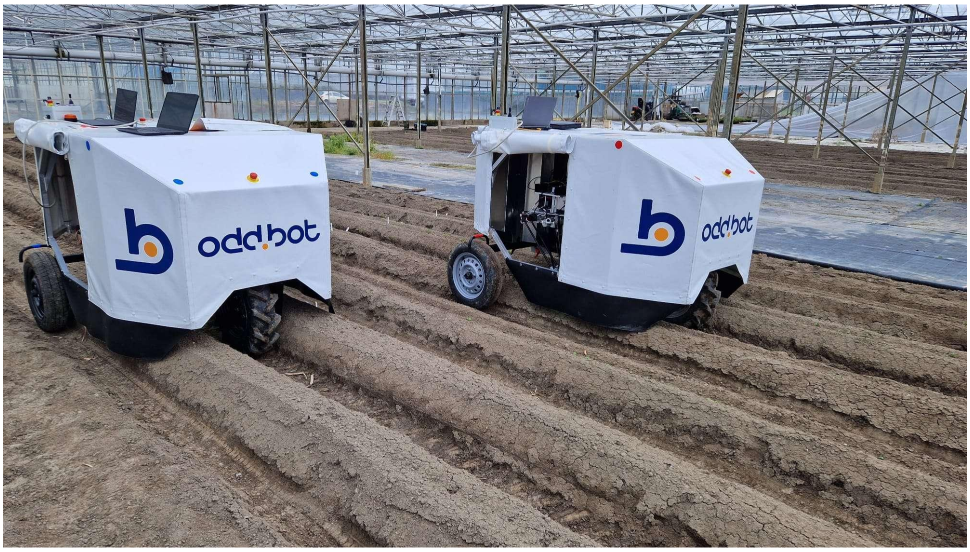


Figure 2: Two Weed Whackos in a greenhouse.

1.1 Personal motivation

My learning at the ENSTA Bretagne finished with a 6-month internship. I found it in a Netherlands start-up based in Rotterdam: Odd.Bot. The internship advantage is double: first in Odd.Bot improves my technical skills, more specifically image processing and deep learning. Also learning about a big in-development project and about start-up issues. And secondly, in a foreign country to improve my English and discover another culture with its own vision of work life.

1.2 Context

With the awareness of the climatic crisis, new challenges are emerging. One among them is finding a way to reduce the use of herbicides without producing less. For now, the only way to do weeding is to either use a lot of herbicides, which kill biodiversity and damage the soil, or do manual weeding, which is long and painful. To avoid this practice many companies started to work on robotized solutions, developing autonomous robots which go weeding in the fields. Odd.Bot is one of them: it is a 5 years-old start-up that is doing its first real partnership this year by selling its product as a service. Even if their robot is capable of weeding, there are still plenty of things to work on for improving and robustifying it.

The solution chosen by Odd.Bot to weeding is a delta arm that will take out or push down the weeds. The robot, called Weed Whacko (figure 2), is composed of a moving platform (the Odd.Carrier) and, fixed to the Odd.Carrier, the Odd.Core: the camera for weed detection and the delta arm for weed removal. The Odd.Core, as the name suggests, is where is the core technology of the company, and the part I will work on.

1.3 Problem statement

Now that Odd.Bot starts working with farmers, they have to do a proper job to retain customers. But some farmers require high precision and performance, which is why the robot capacities need to be improved. My mission here is related to the full system, which consists of three parts: a delta arm (which can be separated into the dimensions of all components and the starting angles), a camera, and their relative position to one another. The main research question is then: is it possible to improve the accuracy of the entire setup, such that an object (in our case, a weed) seen by the camera can be accurately removed by the delta arm? To answer the question, this paper will dive into the following subquestions:

1. Given the way the full system is modeled at Odd.Bot, how would a mismatch in dimensions affect the accuracy?

2. How accurate is the camera as a measurement device, using existing methods and libraries? This includes estimating the relative position of the camera from the delta arm as well.
3. How accurately can we get the starting angle of the motors using the camera?
4. How accurately can we get the dimensions of the delta arm using the camera and machine learning?

This report starts with a preliminary study of the state of the art, then dive into the modeling of the delta arm and influences of dimensions errors. It afterward investigates the measurement accuracy by camera calibration and camera position estimation, and sets up a measurement method for delta arm calibration purposes. The latter is presented in two parts: first, this report proposed a calibration for the motor's starting angles, then it developed the kinetic calibration and the final results that come with it. Finally, the conclusion is a doorway to further improvement proposals.

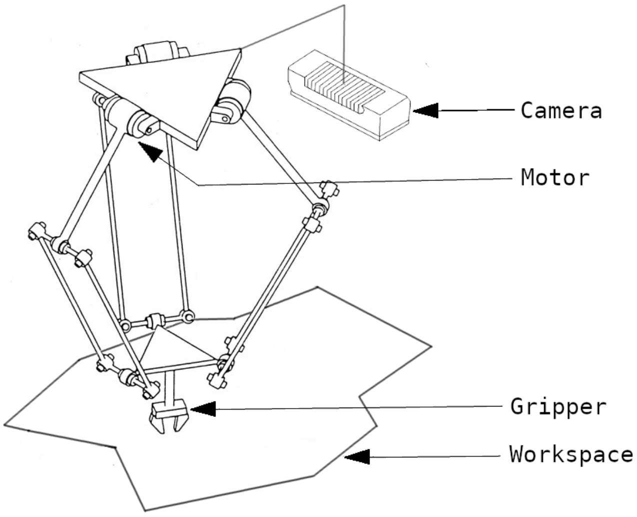


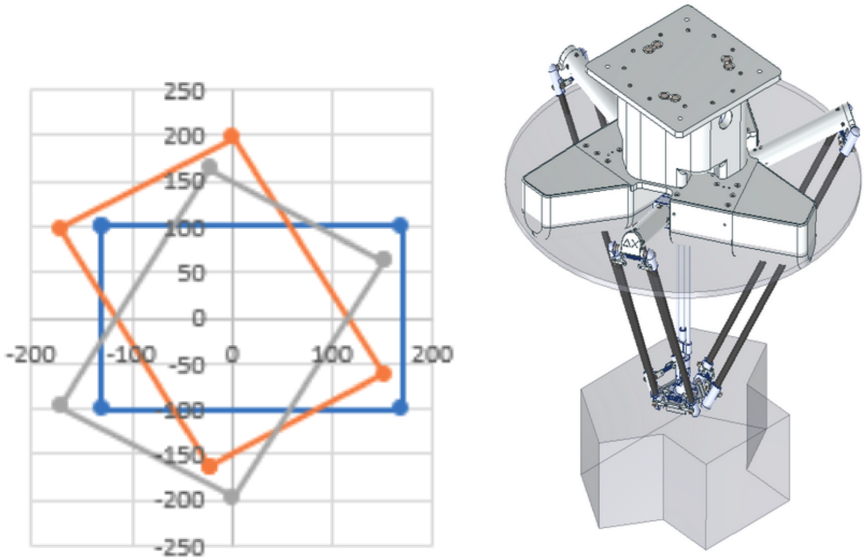
Figure 3: Scheme of the Odd.Core.

2 State of the art

State of the art describes the preliminary research of this project, it introduces the tools and concepts used throughout this report. First, this section introduces the system that this report is about. Then it explains what calibration is and how it works, then describes the measurements: essential for weeding accuracy improvement and arm calibration. The second to last subsection shows how machine learning could be used to improve the command of the delta arm. Finally, this section ends with the chosen method of weeding accuracy improvement.

2.1 The weeding system

Detection of the weeds is handled by a camera linked to the delta arm by a fixed connection, with the embedded computer this formed the Odd.Code (figure 3). This delta arm consists of three connected arms, each moved by one stepper motor. This parallel configuration allows a maintained orientation of the gripper placed at the end of the whole arm. For moving the gripper to the wanted position, the command of the motors is calculated by the inverse kinematic equation. Those commands assume that the motors are initialized perfectly, this initialization happens at each startup with the homing process, using inductive sensors [1](figure 5). Due to his configuration and the length of his limbs, the delta arm has a limited action workspace, which is where it operates optimally (figure 4), causing naturally bigger positional errors at the boundaries than in the middle of the workspace. This Odd.Core is transported by the autonomous carrier which travels in the field following the lines. In the carrots fields where the robot is working, the crops are planted on ridges, so the robot's wheels are guided by the empty space between the rows (figure 2). When the robot moves, the weeds are detected at the front of the delta arm, and then removed when they arrive in the workspace. Based on the number of weeds and the speed of the robot, the removal can happen close to the camera view or further in the workspace. The weeds are only removed around the crops, so it's where the accuracy needs to be high.



(a) Range of each limb (mm). (b) 3D view of the workspace.

Figure 4: Workspace of the delta arm.

2.2 Overview of arm calibration

Calibration is a process for improving robot accuracy by modifying the software rather than the design of the robot [2]. It involves identifying a more accurate relationship between the arm's joints and the gripper's position and using it to permanently change the robot positioning software. Compared to an adaptive control that continuously adjusts the parameters, the calibration happens one time and improves definitely the accuracy of the arm. There are three levels of arm calibration. The joint calibration is the first level, the goal is to establish the relationship between the joint displacement and a joint sensor (like the current homing that calibrates the motor using an induction sensor), and it is typically run at each power-up of the robot. The second level aims to calibrate the kinematic model of the robot. To improve the arm accuracy, the relationship between the kinematic model and the real joint displacement is improved. For this level, the limbs are assumed rigid and the joints perfect, so we assume that there are no non-kinematic errors such as joint compliance, friction, or clearance. These types of errors are handled by the level 3 calibration, which aims to improve the dynamic model.

All of those calibrations have the same four steps [3]: modeling, measurement, identification, and compensation.

- Modeling is choosing a relationship that suits the behavior of the robot, a good calibration leans on a relevant model.
- The second step is here for getting real data by measuring input and output, the quality of this measurement is critical for a good calibration. The measurement error and the real robot error are not easy to separate, to compensate for this it is possible to integrate the error of position and rotation of the measurement device in the model [4].
- The third step consists of the identification of the parameters based on the collected data, it also involves the identification of the error in those parameters due to the noise or the measurement error.
- The final step is compensation, which is the implementation of the new model in position control. Calibration can lead to a significant accuracy improvement [5], but it could not be perfect due to the measurement accuracy and the robot repeatability.

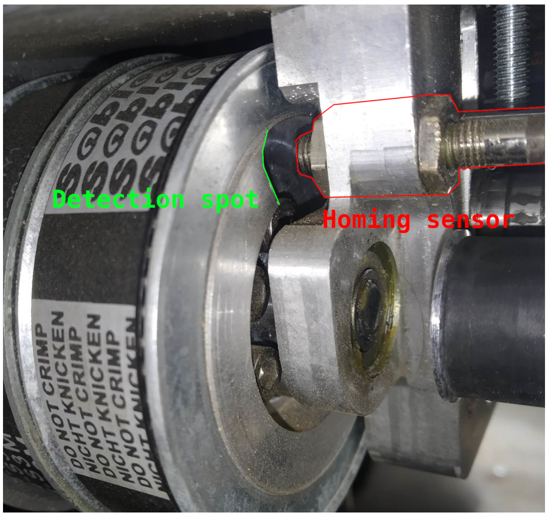


Figure 5: One of the motor's homing inductive sensor.

Sensors	Reachable accuracy (mm)	Commentary
Lidar [14] [15]	~ 10	Not accurate enough
Sonar [16] [17]	0.02 - 10	Underwater
Gnss [18] [19]	~ 10	Not accurate enough
Optoelectronic sensor [20] [21]	< 3	Don't measure in 3D space
Laser tracker [22]	< 0.1	Target have to be slow
Stereo camera [23] [24] [25] [26]	< 1	Have to be calibrate accurately
Depth camera [27] [28] [8]	~ 5	Not accurate enough

Table 1: Sensors comparison table

2.3 Measurement methods

For the arm calibration, the measurement of the gripper position is essential: whatever the wanted precision, the measurement has to be more accurate [6]. For choosing the measurement method, first, let's look at the existing sensors that can be used in our case (Table 1). The sensors able to measure with sub-millimeter accuracy in this use case, for precise arm calibration, are the laser tracker and the stereo camera [7]. Even if the laser tracker is easier to use and provides better accuracy, it is expensive. Especially for a start-up. Add to the fact that Odd.Bot already has stereo cameras makes me choose this second option. The camera used by the company is an Oak D POE camera[8], which has the advantage to be already calibrated. However, calibration is a big part of camera measurement accuracy. Then it could be interesting to evaluate it. Now, if the gripper is clearly detected in each stereo image view, it is possible to calculate the distance to the camera using triangulation equation: 8.1.3[9]. Accurate detection of markers is a deeply explored search field, especially for robots [10][11]. Those specific markers come with robust detection algorithms that assure a good pose estimation [12][13], however for our requirements, cause of the known environment, a newly original detection could be more accurate. To compare this measurement to the command, the position of the camera to the delta arm has to be accurately known[7]. Cause of this, the camera already in the Odd.Core will be used. But even in this configuration, cause of the manufacturing tolerances, the knowledge of the camera position could be improved.

The research on accurate measurement for arm calibration has another goal: the precise camera position will benefit weed detection and removal, improving the whole system's accuracy.

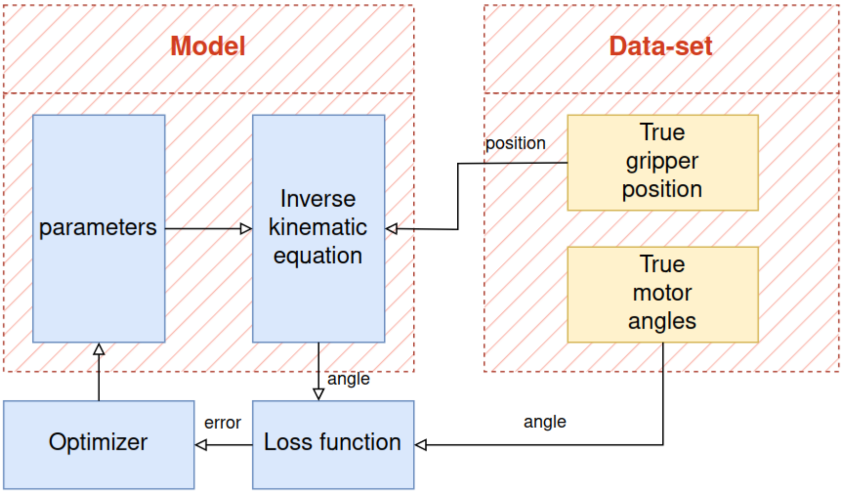


Figure 6: Inverse kinematic model training by machine learning

2.4 Machine learning for identification

The compensation is the final part of the calibration, and there are a lot of different methods for solving parallel kinematic mechanisms[6]. One of those is using computational intelligence.

The concept of machine learning dates back to the 1950s when researchers began exploring the idea of creating computer programs that could learn from data. Progress was relatively slow until the 1990s, when advancements in computing power and the availability of large data sets led to significant breakthroughs. Today, machine learning is able to make complex decisions. it can now recognize patterns, understand language, and perform tasks with remarkable accuracy.

A neural network is one of many models that can learn from data. It modifies its inner parameters to match inputs with outputs. In our case, the model can be the forward or inverse kinetic equation, and our data is the motor angles and the gripper positions (figure 6). Therefore the model parameters will get modified for the equation to match the real arm behavior.

The constraints inflicted by the inverse kinematic equation can improve the convergence of the model [29], so this model seems more numerically efficient [30]. Therefore those constraints are already there in our data set, cause this equation is already used for the control, and the errors, bigger at the gripper than in the joints, could improve the training results. As well, some experimentation proves that the forward kinematic model can lead to better results [7]. All of that shows that the choice of the model depends on the use case, so we will use the simulation to make our choice between the inverse and forward kinematic models. For implementing those models we will use Pytorch, which is a library that allows the creation of personalized models like our arm equations, and personalized loss function for the training.

2.5 Chosen method

As identified before, weeding performances are based on measurement and delta arm accuracy. A method of improvement has been chosen ground on this preliminary work and the available resources.

For this project, the weed measurement will be studied by evaluating the camera calibration and verifying the relative position from the camera to the arm. The delta arm will be calibrated using the method chosen based on this preliminary work and the available resources. The choice of the forward or inverse kinematic model will be made based on the simulation results. Regardless of the chosen model, the parameters will be selected based on the dimensions of our delta arm, and their impact will be identified. For the arm calibration, the gripper measurement will be done using a stereo camera, and the detection will be evaluated to verify accuracy. Finally, the calibration will be done in two phases: a joint calibration with the replacement of the current homing using the camera, and a kinematic calibration using machine learning to identify the model's parameters.

3 System modeling

Before every improvement, it is essential to identify inaccuracy causes and evaluate their impact on the weeding performances. This section introduces and uses the equation of the delta arm as a model to represent and modify the behavior of the system. First, the delta arm is modeled to include parameters that expressed inaccuracy causes. Then, using this model, the impact of those parameters is evaluated in the simulation.

3.1 Error identification

To move the gripper where the camera detects the weed, the distance and orientation between the arm and the camera are needed. We assume to know it with an accuracy of 1mm and 0.01 rad. For moving the gripper to the wanted position, the command of the motors is calculated by the inverse kinematic equation. This equation uses the delta arm dimensions which are known with a precision of 1mm due to manufacturing tolerances. Those commands assume that the motors are initialized perfectly, but in reality, this initialization is handled by inductive sensors with an accuracy of 0.006 rad. There are other geometric error sources [22][30], but this study is confined to those previous ones.

The delta arm is made of three arms. an arm is composed of two limbs joined together, the upper one is linked to the chassis plate and the lower one to the gripper. An arm i is therefore modeled by four parameters:

$$\{R, r, L, l\}_i \tag{1}$$

Where L and l are the lengths of the upper and lower limb, R is the distance from the motor to the chassis plate middle and r is the distance from the gripper center to the lower joint. The delta arm is modeled by three arms which aims at a model of 12 parameters (equations 8.1.1 and 8.1.2).

The error during the homing process aims at a homing angle error, it is modeled by an offset on the angle command in each motor:

$$\{e_h\}_i \tag{2}$$

Since this offset is different at each initialization, this parameter is dissociated from the previous ones as it will be calibrated another way.

Another factor that influences the gripper position is the relative position from the camera to the arm. Indeed if this position is incorrectly known, even if the weed detection is perfect, the removal accuracy will be relative to this position. A translation and a rotation are introduced to depict the real camera position:

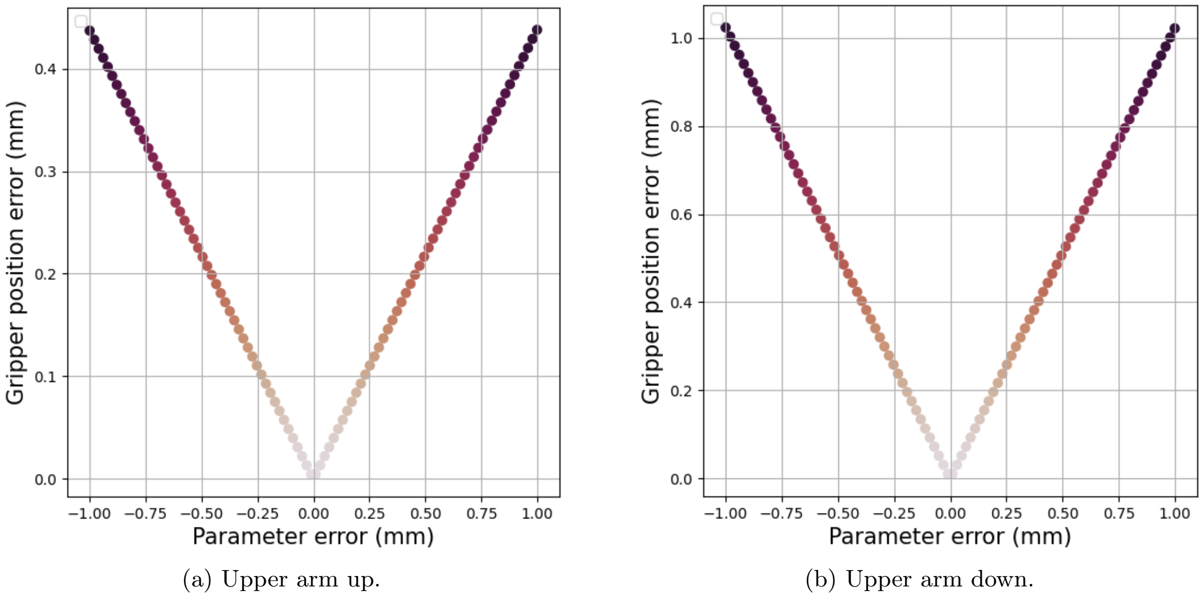
$$\{\beta_x, \beta_y, \beta_z, \theta_x, \theta_y, \theta_z\} \tag{3}$$

Those 6 parameters will be determined first, during the measurement improvement phase, cause a good measurement is essential to the calibration of the delta arm model.

The modelization of the arms, the homing error, and the camera position lead to 21 tunable parameters. Those, fitted well, will lead to improvement of the measurement and of the arm behavior, and this way improve the weeding accuracy.

3.2 Parameters impact

Once those parameters are identified, it is essential to evaluate their impact on the gripper position. For this, the delta arm is simulated using the inverse kinematic equation. Then, by modifying a parameter on the simulation, it is possible to see its impact by comparing the gripper position to the command. Each parameter has a different impact on every three dimensions X, Y, and Z the depth. For this study, the accuracy is expressed by the distance of the gripper from a referential point: a value that includes the error in every three dimensions.



(a) Upper arm up.

(b) Upper arm down.

Figure 7: Simulated impact of parameter $R1$ on the gripper position accuracy, function of command position (figure 4a).

First, this subsection identified the impacts of the delta arm dimensions which is known with a precision of 0.1mm. Due to the parallelism of the arm, the error of the parameters can compensate or accentuate themselves, and their impacts are different depending on the position of the gripper. Each R_i has a more significant influence when the arm i moves to the front (upper arm down). As shown by the error in different positions in figure 7, an error in this parameter impacts the accuracy from 40% to 100% of the parameter error. By the same process, it is identified that r_i has the same impact as R_i, L_i impacts the accuracy the most when arm i moves to the back: by 40% to 60% of the parameter error. Finally, l_i has a constant impact of 40% of the parameter error. Each of those parameters hasn't a big impact individually, but all together they can lead to a substantial non-linear error.

Afterward, the impact of the homing offset is discussed. This parameter is known with an accuracy of 0.006 rad and each $\{e_h\}_i$ leads to a constant linear error of 1.5mm maximum. As for the dimensions of the arms, the parallelism of the delta arm conducts at each $\{e_h\}_i$ inflict on each other impacts. Those parameters have a big impact on the system's accuracy, but compared to the arms dimensions there are easy to dissociate cause are just an offset on the command, and so can be calibrated independently.

Finally, the camera position error is the last parameter to evaluate. It is composed of a translation and a rotation known with a precision of 1mm and 0.01 rad. The translation simply induces an inaccuracy equal to itself cause it shifts the measurement related to the arm. The rotation is the one that creates the bigger error since its impact is based on the depth from the camera to the gripper. This error is linear and it caused a maximum inaccuracy of 4mm to 5mm depending on the depth in the workspace. This is the biggest identified error by far, showing that the dimensions of the support of the camera are really relevant parameters.

Now that the system is widely known and the impacts of mismatch in dimensions have been identified, it is time to dive into the system accuracy improvement, starting with the visual measurement.

4 Measurement

Good delta arm calibration required accurate measurements, indeed the more the data set is close to reality the more the calibration will rectify the behavior of the arm. For this, this section will start by diving into the measurement of the 3D coordinate of the gripper, with particular attention to marker detection. Then, the method of measurement is explained, and finally, the camera position is determined and the measurement accuracy is discussed.

4.1 Coordinate measurement

For a good measurement, a good camera calibration is vital. Fortunately, the used camera is calibrated in the factory, assuring good distortion correction and rectification. However, to estimate the measurement accuracy it is useful to evaluate camera calibration. For this, a simple test is set up: a marker is fixed to a flat straight sliding device (figure 8), and measurements are made horizontally and vertically by sliding the marker on the device. Then each bunch of measurements is fitted by a straight line using least squares fitting: figure 9. The calculation of the sum of squared residuals informs on the curvature of the camera image: table 2. The biggest error is, as expected, on the borders of the image, with a maximum of 0.47mm localized at the bottom of the picture. The errors induced by the camera calibration are through less than 0.5mm, which is sufficient to calibrate some of the identified errors and promise good weed detection.

To get the coordinates of the gripper we need to detect its position in both stereo images. For sub-pixel detection, a clear marker and a good detection algorithm are needed. The chosen marker has to be small to allow its detection in the biggest part of the camera view, indeed in the setup the camera is close to the gripper, so the viewing angle is constrained. Here three markers with their detection algorithms are compared. One is the ArUco marker, well-known in robotics for its robustness, the two others are original markers created for accurate detection.

- The chosen ArUco marker is 4x4 bits to get the best results with the size restriction, and it is detected using openCV tools with standard detection parameters. For the following original markers, the detection parameters are tuned based on experimentation.
- Ellipse is the easiest shape to detect accurately [31], so the first original marker will be a white dot on a black background. For the detection, the following algorithm is used: figure 11.a. The grayscale image of the stereo camera passes through a gamma correction filter to raise the contrast, then thresholding creates a binary image after which the contours of the shapes are extracted [32]. Those are filtered based on their size and circularity, and finally, the shape is fitted by an ellipse [33] for a sub-pixel detection of the marker's center.
- The second original detection is based on the first one and designed to improve accuracy and repeatability. The idea is to multiply the object detection number by creating a pattern to lower measurement error, as it could be used for camera calibration [34][35] or vision-based metrology [7]. There is no consensus on the optimal number of detection points other than the more the better, but some studies with good results use around fifteen points. Cause of the camera view detection issue, the pattern will then be made up of thirteen dots (figure 10). The detection algorithm is as follows: figure 11.b. As in the previous detection the image is binarized, the shape contours are extracted, filtered, and fitted by ellipses. To avoid similar objects from the environment slipping into the detection, the remaining points are filtered to keep the thirteen closest dots from the center of the detection cloud. Finally, to avoid missing pattern points, the rectangularity of the detection cloud is evaluated. The pattern middle is found by the average of the position of the points, thanks to the symmetrical shape.

The last step is the calculation of the 3D position of the marker from the pixel position in each image. The triangulation equation is used to get the marker coordinates (equation 8.1.3). Now that 3D coordinates can be measured with all the different detection, there are compared to choose the more relevant one. The marker comparison will be based on robustness and repeatability for getting an accurate distance between each measurement thereby generating a reliable and accurate data set.

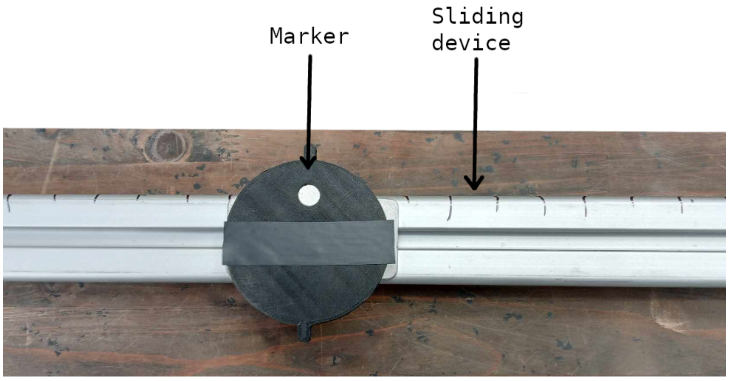


Figure 8: Setup for testing camera calibration.

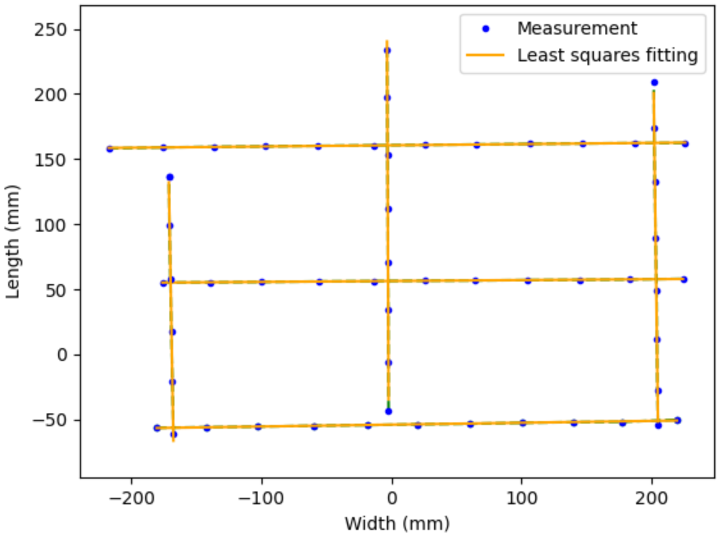


Figure 9: Measurements and least squares polynomial fitting of lines.

For evaluating repeatability, several measurements in fixed position are executed, and the standard deviation is calculated: table 3. Those values show variation between each measurement. First, it appears that all detection have a planar dispersion lower than 0.1mm, which is very good. Pattern detection has the best repeatability, followed by the dot and the ArUco detection. It is interesting to see that the depth measurement has the biggest disparity, which reaches a significant deviation of 0.5mm more than 5% of the time with the ArUco detection.

To evaluate the robustness, measurements are made in different coordinates in the workspace: table 4. Light and angle fluctuation can lead to non-detection of the marker, requiring user intervention. With big data sets to gather it is important to minimize external assistance, but usually robust detection comes with poor repeatability. As expected ArUco marker is detected in all cases, dot detection for its part needs a little assistance, and pattern detection requires light adjustment nearly 15% of the time.

The goal of this marker's detection is to measure the behavior of the arm. Since the latest already has an accuracy of nearly one millimeter, a really precise measurement is needed. For this reason, the ArUco marker is not the best choice. Indeed, even if it is more robust than the other solutions, its significant depth deviation invalidates its use. The choice goes then for dot detection: it is way more robust than pattern detection, which simplifies its use, and its repeatability is good enough compared to identified errors.

line position	horizontal	vertical
bottom/left	0.47	0.09
middle	0.06	0.01
top/right	0.22	0.07

Table 2: Sum of squared residuals of the least squares fit (mm).

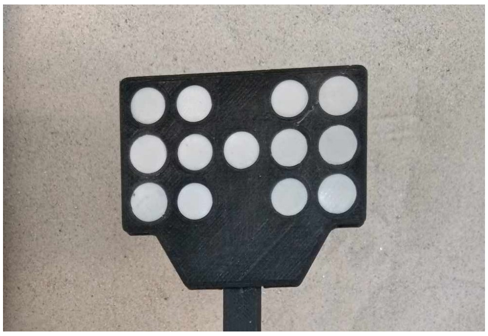


Figure 10: Picture of the original pattern.

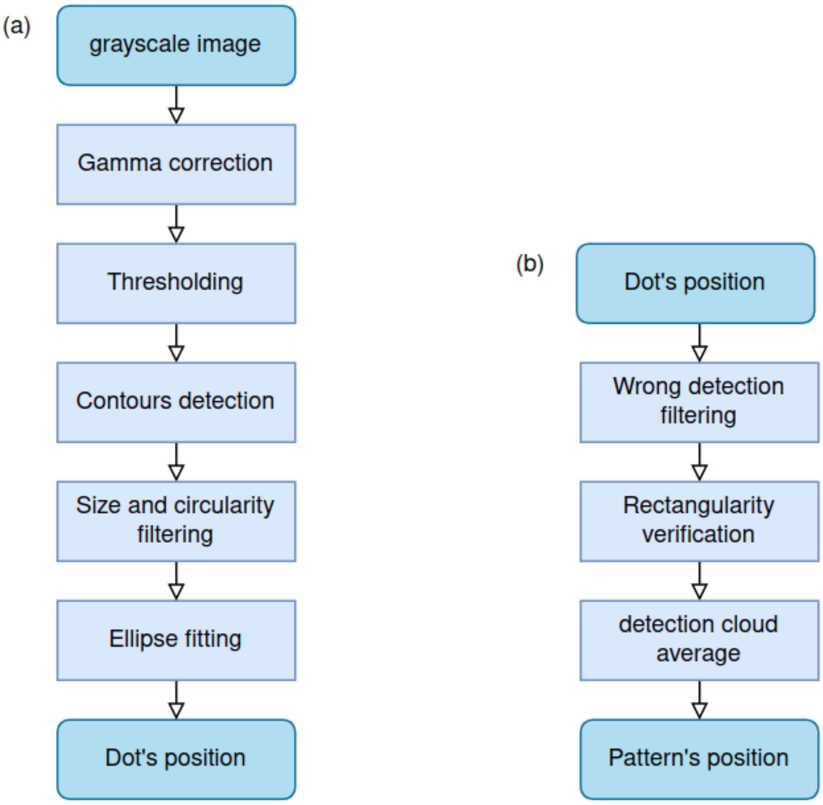


Figure 11: Diagram of the dot detection (a) and pattern detection (b) algorithm.

	Dot	Pattern	ArUco
X	0.013	0.003	0.025
Y	0.012	0.009	0.037
Depth	0.058	0.017	0.275

Table 3: Standard deviation for 100 measurements using triangulation (mm).

	Dot	Pattern	ArUco
user intervention	5	18	0
recurrences (%)	4	14.4	0

Table 4: Number of user intervention needed for 125 measurements.

4.2 Method of measurement

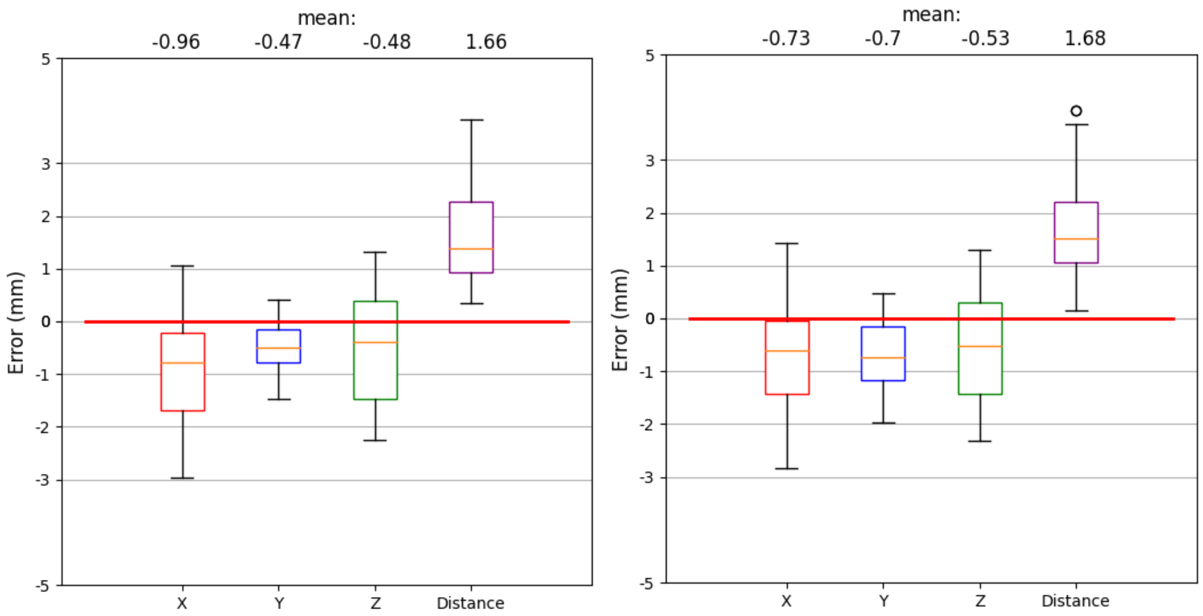
In addition to accurate detection, good acquisition is essential for an accurate measurement. Due to the setup (figure 3) the camera can't see the gripper in the whole workspace. To resolve that an extender is fixed to the gripper: figure 13. Through this, by shifting the measurement, it is possible to get the gripper's position in a bigger zone. To measure data sets of the gripper's position, the arm is moved to different locations, and the program saves the mean value of several measurements of this location. To be sure that the measurement matches the arm behavior, the gripper should have finished moving to the wanted position, and the detection detected the marker. To confirm this, the program verifies that the measurement is closer than three centimeters from the command and that each measurement of the same position is close to one another. In addition, to avoid positional offset, the measurement is initialized in (0,0,0). So each measurement measures the position of the marker relative to the initial measurement.

4.3 Camera position and results

The measurements produced by the latter method for each detection are compiled in figure 12a to 12c. It displays the error between the measurement and the command, composed of the errors of the system identified previously in this report. As expected, each measurement is pretty close, and the significant depth deviation of the ArUco detection is visible by an extra variation of 1mm.

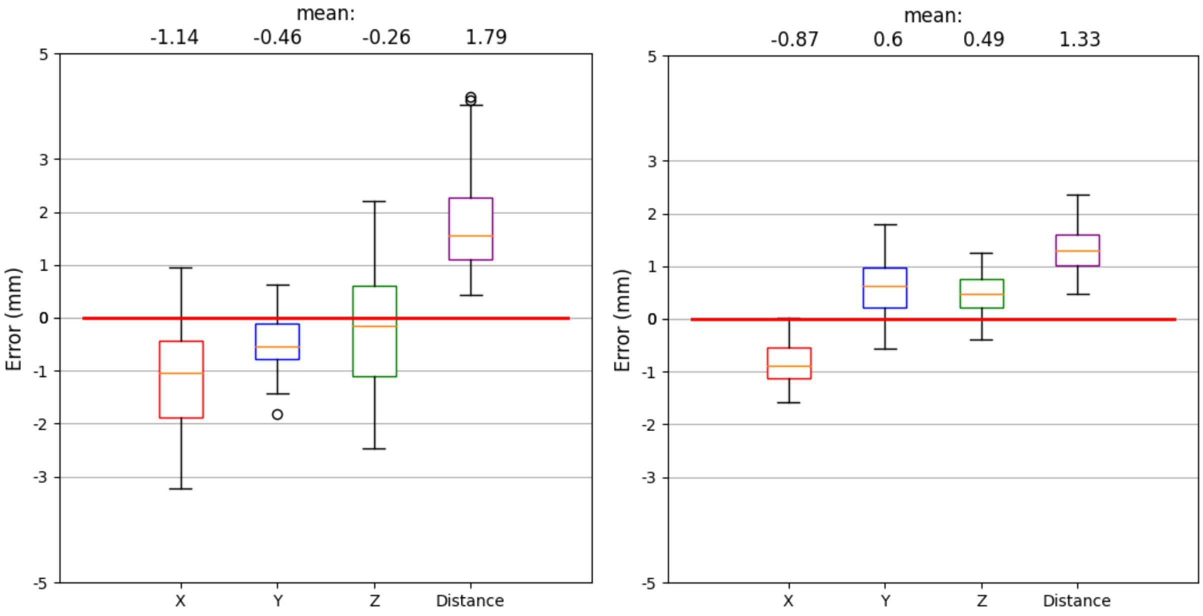
For a good calibration of the delta arm, those data sets should only include the error of the delta arm and the homing. The goal here is so to reduce errors in camera position and measurement. The measurement has already been improved in this section, then only the camera position improvement is missing. The latter is done by determining the transformation between the data set and its command (translate to the error in the camera's position) using singular value decomposition (SDV). By applying this transformation $\{\beta_x, \beta_y, \beta_z, \theta_x, \theta_y, \theta_z\}$ during the measurement a new data set is generated: figure 12d. It demonstrates the improvement induced by a better camera position. From figure 12a to figure 12d variation of planar and depth error have been decreased by 2mm, resulting in a data set closer to the behavior of the delta arm, and especially a significant accuracy improvement of the whole system. The remaining offset is due to the homing offset $\{e_h\}_i$, errors in delta arm dimension, and camera calibration.

To conclude, Three different markers for visual measurement accuracy have been used to determine visual measurement accuracy, improvement of accurate detection, and position determination. It is now a fact that the camera as a measurement device assures sub-millimeter accuracy, with the worst case being an error of 0.5mm, a consequence of image distortion and detection variation. This improvement ensures, first, better weeding, and also the gathering of an accurate data set for delta arm calibration.



(a) Dot detection measurement.

(b) Pattern detection measurement.



(c) ArUco detection measurement.

(d) Dot detection measurement with determined $\{\beta_x, \beta_y, \beta_z, \theta_x, \theta_y, \theta_z\}$.

Figure 12: Error between measurement and command for 125 measurements throughout the workspace.

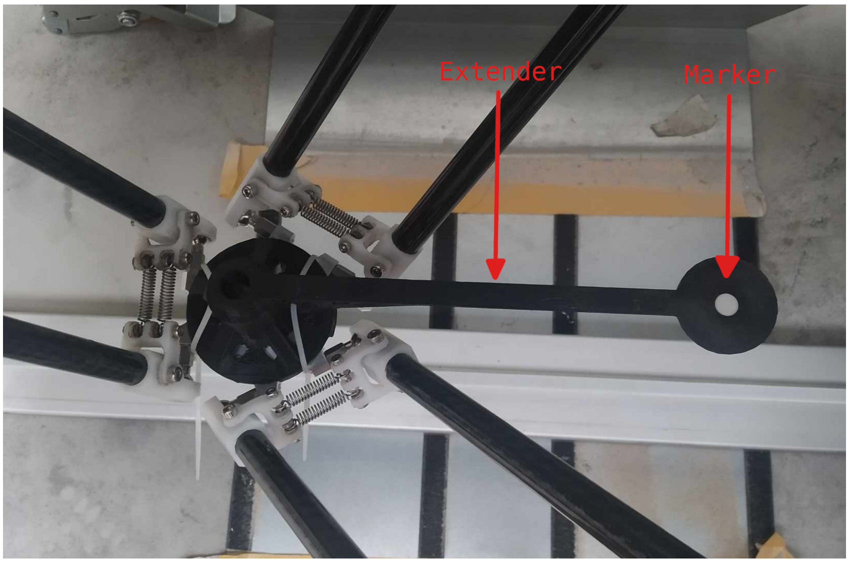


Figure 13: Extender fixed on the gripper.

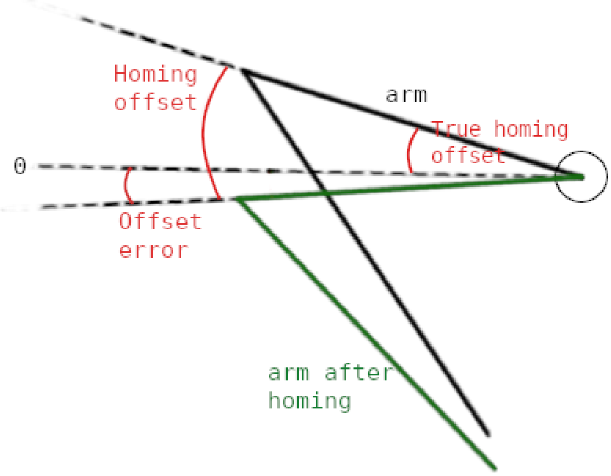


Figure 14: Diagram of an limb homing.

5 Joint level calibration: Visual homing

The first step of arm calibration is joint calibration, translated here by the homing of the motors. First, a subsection will explain the new homing that uses the camera rather than the inductive sensor, and how it takes as input the measurement error. Then, the accuracy of both the previous homing and the visual homing are measured and compared, and the results and limitations are discussed.

5.1 Homing algorithm

The goal of the homing is to find the offset α_h between the startup angle of the motor and the requested initial angle, so we can initialize the motor's position. For now, according to the references of the inductive sensor, the current homing finds α_h with an accuracy of 0.006 rad: maximal error of homing offset error e_h, displayed in figure 14 by the difference between the homing offset and the true homing offset α_h. This error aims at a maximal positional error of 1.5 mm according to the simulation (section 3.2).

This section implements a new homing using the stereo camera. The following algorithm, through three steps, using multiple measurements and measurement errors, determined the homing offset.

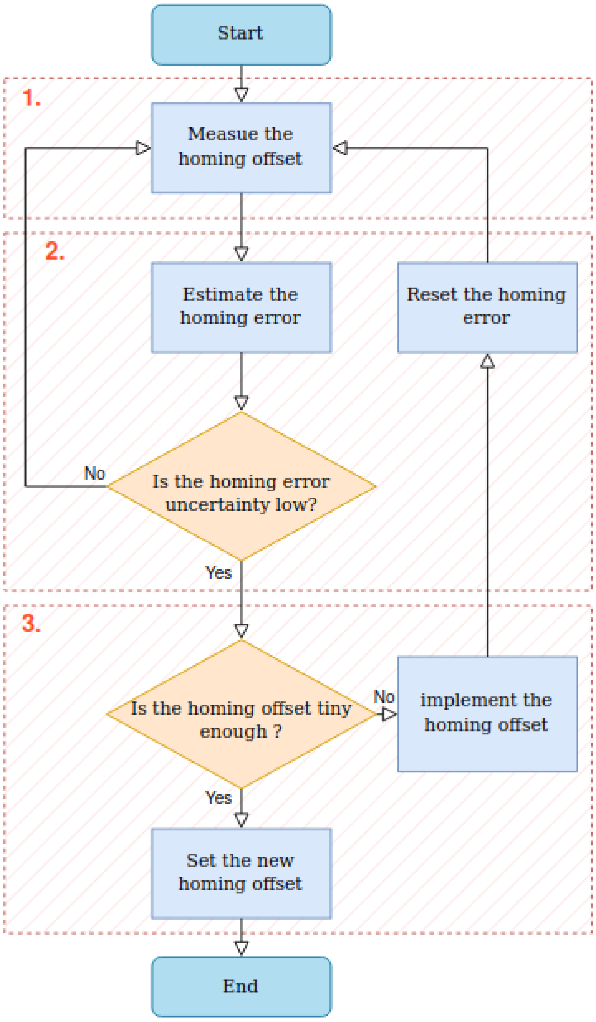

Start
1.
Measue the homing offset
2.
Estimate the homing error
Reset the homing error
No
Is the homing error uncertainty low?
Yes
3.
Is the homing offset tiny enough ?
No
implement the homing offset
Yes
Set the new homing offset
End

Figure 15: Diagram of the homing algorithm.

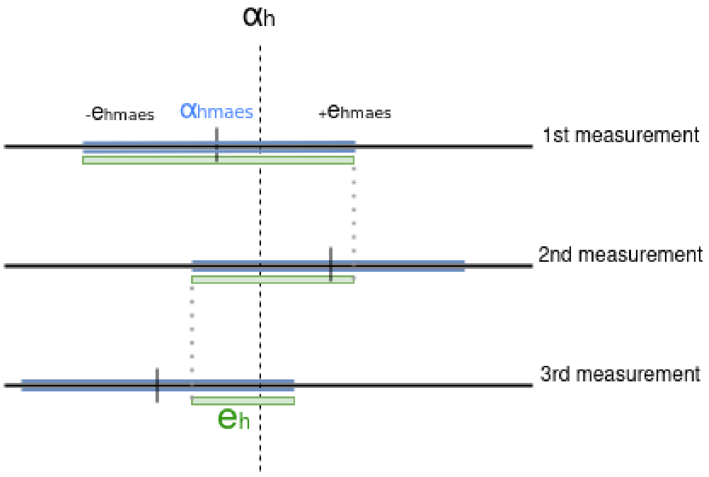


Figure 16: Estimating the homing offset by several measurements using intervals.

1. Measure the homing offset
   For measuring the angle offset $\alpha_{h_{maes}}$, the motor command α is compared to the measured motor angle α_{meas}.
   $$\alpha_{h_{maes}} = \alpha - \alpha_{meas} \tag{4}$$
   The measured motor angle is calculated from the measured gripper position using a marker fixed on it, coordinate transformation from the camera to the arm, and the inverse kinematic equation: equation 8.1.1. But due to the error of both the measurement and the model, There is a homing offset error e_h (figure 14). It is therefore necessary to improve the accuracy of this value for meticulous calibration

2. Estimate the homing error
   Now the algorithm considers the measurement error $e_{h_{maes}}$. By several measurements on different gripper positions, the algorithm will refine its knowledge of the true offset angle by estimating the homing error e_h. To achieve this, the algorithm uses the interval generated by the measurement and its error: $[\alpha_{h_{maes}} - e_{h_{maes}}, \alpha_{h_{maes}} + e_{h_{maes}}]$. By merging several measurement intervals the algorithm is sure to both include the true value α_h and get closer to it (figure 16).

   For this algorithm to estimate the homing error e_h it needs different measurements. To speed up the algorithm, rather than a random gripper position, it is better to choose a position with a big error, like near the borders of the workspace. This part of the algorithm multiplies the measurements until the uncertainty of the homing error is tiny enough.

3. Validate an accurate homing offset
   With the latter steps, by subtracting the homing error, the algorithm gets the homing offset with great accuracy. At least when this homing offset is tiny. Indeed, according to the experiments, when the homing offsets are big or far from each other, the algorithm doesn't find the perfect value (Table 5). This final step is here to verify that the found homing offset is good by rerunning the two first steps with the homing offset implemented. If the resulting homing error of this rerunning is lower than one increment, then the first homing offset was good, and the homing is finished. Otherwise, the algorithm adds this new homing error to the previous homing offset and reruns the two first steps another time.

Fixed homing error			Founded homing error			Std for seven experiments		
motor 1	motor 2	motor 3	motor 1	motor 2	motor 3	motor 1	motor 2	motor 3
5	0	0	4,7	-0,6	-0,1	0,2	0,4	0,4
5	-5	20	3,7	-6,6	18,6	0,2	0,3	0,5

Table 5: Homing error (in motor increments) found by the two first steps of the algorithm.

Standard homing			Visual homing		
motor 1	motor 2	motor 3	motor 1	motor 2	motor 3
3,68	1,86	3,56	0,37	0,49	0,47

Table 6: Standard deviation (in motor increments) of each homing method.

5.2 Results

The precision of a homing is evaluated by the standard deviation of the homing offset. Indeed, with perfect homing, the remaining homing error found by the algorithm should always be less than one increment. Table 6 compared the standard deviation of the homing offsets after standard or visual homing for several experiments. It appears that, as expected from the sensor references, the standard homing has an accuracy of 10 increments 97.7 % of the time (three times the standard deviation). The new homing, for his part, is accurate to one increment 97.7 % of the time, which is a strong improvement.

To validate this result it is possible to use an inclinometer, which is a sensor that measures the angle between two planes. After the homing, the angle between the arm's chassis and the upper arm in the initial position should be zero (offset error equal to zero in figure 14). This experiment shows that this angle is closer to zero after a visual homing than after a standard one. This experiment also validates the visual homing by finding a homing error that looks like the one estimated by the algorithm.

Despite that real improvement, it also appears that the standard homing is accurate to 3 increments 68 % of the time (one time the standard deviation), which aims at a maximal positional error of 0.4 mm according to the simulation. That means that the standard homing error is negligible most of the time compared to the whole position error. Moreover, the visual homing has a limitation: The gripper has to start under the camera, otherwise, the camera can't detect it and so can't do the homing. Therefore, for now, it needs the current homing to move the gripper in the camera view. However, a new homing is essential for sub-millimeter accuracy, and this visual solution leads to a nearly perfect homing without any new sensors than the camera already there, it therefore is an accurate and affordable solution.

6 Kinematic level calibration: Machine learning

After the modeling and the measurement, the kinematic calibration includes the fitting of the parameters and the compensation inducted by the new model. For this, first, the forward and inverse kinematic models are compared to choose the more relevant ones in this use case. Then this section evaluates which data set should be used to train the model. Finally, the results of the calibration are shown and discussed.

6.1 Model comparison

The kinematic calibration fits the model to the delta arm real behavior by modifying the value of the parameters $\{R, r, L, l\}_i$. For this, the machine learning model is fed with measurement of the arm: a data set composed of motor angles and related gripper positions. At each iteration of the training, the machine learning model evaluates the accuracy of the parameters using the loss function. The chosen loss function is the mean value of the gripper positional error.

The delta arm has been modeled with both forward and inverse kinematic equations. Both can be trained to get new dimensions, but one should get more accurate parameters, which is why comparing these models is essential for a good calibration. The comparison method is as follows: generating a data set in simulation, training both models on this data set, generating a new data set using the parameters from the training, and finely comparing the first data set with the ones from the training. A low error between the compared data sets signifies good model training, therefore the chosen model will be the one with the lowest error. To match the real system the first data set is generated with randomized parameter errors ranging from 0 to 1mm, where the models are initialized with currently used dimension values. Table 7 gathers the results of the comparison method for different data sets and training epochs. It appears clearly that the forward kinematic model doesn't manage to converge since the error gets bigger with the epochs. In contrast, the inverse kinematic model error converges to zero, no matter the data set and its size. Therefore it will be used to calibrate the delta arm, with this time a real data set generated by visual measurements.

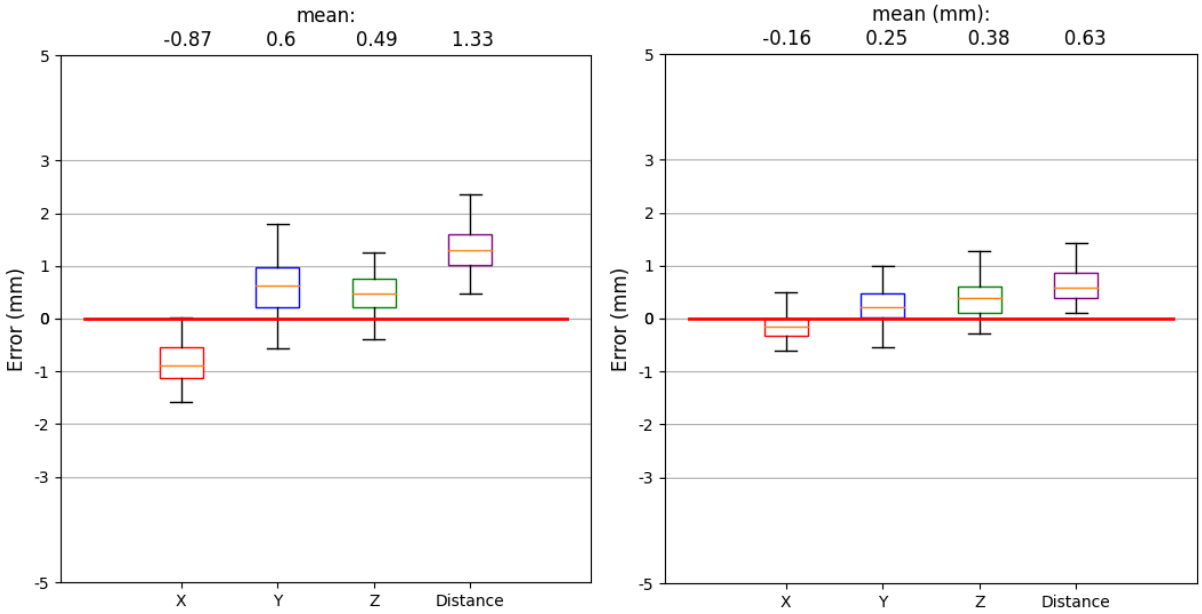
6.2 Data sets comparison

But before generating a real data set, it is essential to identify the data that will assure good training results. A light data set could cause overfitting, and a data set with too much measurement could cause underfitting. As well, the location of measurements is also important: with a uniform distribution around the workspace, the model will easily capture the behavior of the arm. location where the kinematic error is bigger, like at the boundaries of the workspace, could also improve the training [22]. But the present measurement method does not allow all locations, therefore it is important to compare the training with data sets including measurements at the workspace boundaries, all around the workspace, and just under the camera. This is to verify that the data set from the measurement method is worth training.

Table 7 gathers the results of the training with different data sets (in the camera view, uniformly distributed in the workspace, or at the boundaries) of different sizes, trained during various epochs. First, it appears that whatever the measurement location, they all reach a sub-micrometer error with 5000 epochs. As well, whatever the size of the data set they all surprisingly reach the minimal value with 5000 epochs. The latter is surely due to the simulation that doesn't model all the errors. But the real data sets include errors due to the measurement accuracy. Therefore, to attenuate the rate of inaccurate measurements it could be better to choose a big data set. Cause of this, and due to the fact that the camera view training gets as good results as the other data sets training, the chosen data set for delta arm calibration is generated by the measurement method with 125 different locations.

6.3 Results

After the visual homing, the measurement of a data set using the measurement method, and the correction of the camera position, the inverse kinematic model is trained. The delta arm is calibrated by the substitution of the dimensions with the parameters from the model training. For evaluating this calibration a new data set is gathered: figure 17b. The comparison between the measurement for the training (figure 17a) and the results of the calibration (figure 17b) is unequivocal: the mean error has decreased by 0.70mm and the variation by 0.50mm, reaching a system accuracy of 1.63mm, lower than 1mm 75% of the time. It proves that the dimensions of the delta arm can be determined quite accurately using the camera and machine learning. But those new parameters also compensate for measurement errors since the model training aims to minimize errors in the given data set, so the new dimensions are not exactly the ones of the delta arm. However, the kinematic calibration assures a better weeding by lowering the system errors and completes the process of accuracy improvement.



(a) Measurement for the training.

(b) Results of the kinematic calibration.

Figure 17: Error between measurement and command for 125 measurements before and after kinematic calibration.

trained data set	bash size	epochs	error inv (mm)		error fwd (mm)	
			mean	max	mean	max
camera view	24	50	0.68	0.99	0.66	0.98
workspace			0.68	1	0.66	0.98
boundaries			0.68	1	0.67	0.99
camera view	24	500	0.63	0.83	0.9	1.16
workspace			0.57	0.78	0.96	1.22
boundaries			0.72	0.94	0.85	1.25
camera view	24	5000	20×10^{-5}	27×10^{-5}	0.95	1.11
workspace			5×10^{-5}	7×10^{-5}	0.93	1.04
data set boundaries			19×10^{-5}	25×10^{-5}	0.9	1.13
camera view	64	50	0.68	0.99	0.66	0.98
workspace			0.67	0.99	0.66	0.98
boundaries			0.68	0.99	0.66	0.98
camera view	64	500	0.62	0.83	0.89	1.15
workspace			0.62	0.85	0.93	1.21
boundaries			0.6	0.83	0.94	1.24
camera view	64	5000	48×10^{-5}	59×10^{-4}	0.95	1.12
workspace			53×10^{-5}	74×10^{-5}	0.91	1.11
boundaries			22×10^{-5}	30×10^{-5}	0.92	1.2
camera view	125	50	0.68	0.99	0.66	0.98
workspace			0.68	0.99	0.66	0.98
boundaries			0.67	0.99	0.66	0.98
camera view	125	500	0.63	0.84	0.9	1.16
workspace			0.62	0.84	0.91	1.19
boundaries			0.62	0.84	0.92	1.24
camera view	125	5000	19×10^{-5}	24×10^{-5}	0.94	1.11
workspace			33×10^{-5}	46×10^{-5}	0.91	1.15
boundaries			35×10^{-5}	48×10^{-5}	0.92	1.19

Table 7: Simulated experimentation of models training with different data sets.

7 Conclusion

Odd.Bot does robotic weeding. With the rising number of autonomous alternatives to herbicides and hand weeders, the company has to stand out for retaining customers. Consequently, the weeding should be beyond reproach and the system's accuracy should avoid damage to crops or miss weeds. With this in mind, this paper has searched to improve the accuracy of the entire setup, such that an object (in our case, a weed) seen by the camera can be accurately removed by the delta arm.

During the first section, the impact of mismatch in dimensions has been identified. Using simulation, it has been proved that every determined mismatch has its own relevant impact on the system's accuracy, leading to a whole error of several millimeters. The latter preliminary identification achieved, the visual measurement has been the first to be improved since it causes the biggest error and it is used to calibrate the other dimensions. During this part, the camera calibration has been evaluated to estimate weed detection accuracy: results show that this can lead to errors of 0.5mm. For the following arm calibration, the detection accuracy of the gripper has been estimated by the comparison of several visual markers, assuring accurate measurement of the delta arm position. In closing of this section, the system's precision is improved by 2mm with the determining of the camera position relative to the delta arm. After that measurement accuracy has been identified and improved, the remaining dimensions are calibrated, starting with the motor homing. The latter, by an original algorithm using the camera, is accurate to 0.0006 rad despite some limitations of use. This study ended with the kinematic calibration of the delta arm dimensions using machine learning. After identification of the model to train, and by using the improvement of the previous sections, it results in a mean system accuracy of 0.6mm, and a variation 3mm lower than initially. Assuring the system to remove the weeds with higher accuracy.

Finally, with the studies and experimentations developed in this paper, it is possible to conclude: yes, it is possible to improve the accuracy of the entire setup, such that an object seen by the camera can be accurately removed by the delta arm.

However, the remaining accuracy is not perfect. For lack of time, the camera calibration hasn't been improved, and some leads, like adding joint dimension in the model, have not been successful. The identification of parameters' impact has also been a big part of the work since every error overlaps others. To go further the model could include gripper orientation, which has been assumed horizontal. Dynamic calibration of the delta arm could be realized, camera calibration improved, and a laser tracker could be used to overstep camera limitations.

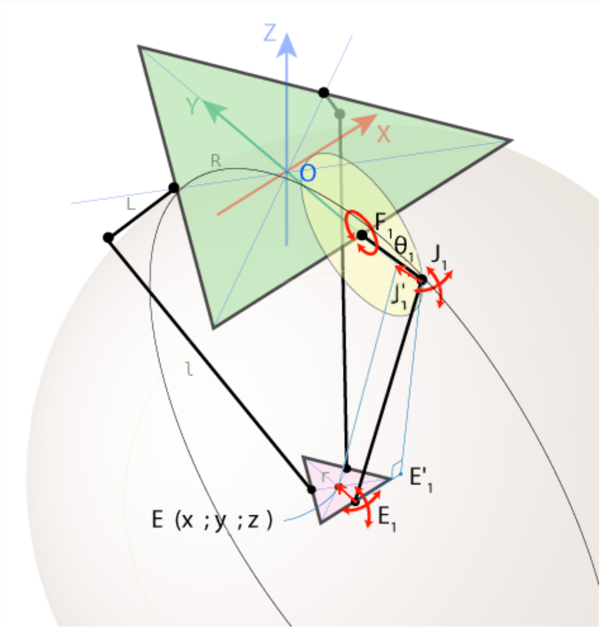


Figure 18: Diagram of the arm.

8 Appendix

8.1 Equations

8.1.1 Inverse kinematic equation

The inverse kinematic equation relates the position of a robotic arm's gripper in three-dimensional space (X, Y, Z) to the angles of each motor θ_i, $i = 1, 2, 3$. This equation allows us to control and move the robotic arm by specifying the desired gripper position.

The equation of a limb is expressed in the YZ plane (figure 18). For the first limb θ_i is equal to:

$$\theta_1 = actan\left(\frac{z_{J1}}{y_{F1} - y_{J1}}\right)$$

Let's find the coordinate of $J_1(0, y_{J1}, z_{J1})$ using Pythagore theorem on F_1J_1, $E_1'J_1$. Let L_i and l_i define the lengths of the upper and lower arm segments, and r_i and R_i the gripper and upper arm radius.

$$E_1(X, Y - r_1, Z) \Rightarrow E_1'(0, Y - r_1, Z)$$

$$E_1'J1 = \sqrt{{E_1J_1}^2 - {E_1E_1'}^2} = \sqrt{l_1^2 - X^2}$$

$$\begin{cases} (y_{J_1} - y_{F_1})^2 + (z_{J_1} - z_{F_1})^2 = L^2 \\ (y_{J_1} - y_{E_1'})^2 + (z_{J_1} - z_{E_1'})^2 = l_1^2 - X^2 \end{cases} \Rightarrow \begin{cases} (y_{J_1} + R)^2 + z_{J_1}^2 = L^2 \\ (y_{J_1} - Y + r_1)^2 + (z_{J_1} - Z)^2 = l_1^2 - X^2 \end{cases} \Rightarrow J_1(0, y_{J1}, z_{J1})$$

Due to the design of the robot, the coordinate system can be rotated to get each motor angle with the same equation. So θ_2 and θ_3 are found using the equation in the new reference frame $X'Y'Z'$.

$$\begin{cases} X' = Xcos(\alpha) + Ysin(\alpha) \\ Y' = -Xsin(\alpha) + Ycos(\alpha) \\ Z' = Z \end{cases} , \alpha = \frac{2\pi}{3}, \frac{4\pi}{3}$$

8.1.2 Forward kinematic equation

The kinematic equation relates the angles of the motors (θ_1, θ_2, θ_3) to the position of the robotic arm's gripper in three-dimensional space (X, Y, Z). The strategy here is to calculate the coordinate of $E(X, Y, Z)$, the intersection of three spheres representing the rotation of each limb. Those spheres have l_1 as radius and J'_i as center, the translation of J_i from the vector E_iE (figure 18).

Let L_i and l_i define the lengths of the upper and lower arm segments, and r_i and R_i the gripper and upper arm radius. Let's find the coordinates of J'_i:

$$\begin{aligned}
&OF_1(0, -R_1, 0)\\
&F_1J_1(0, -L_1\cos\theta_1, -L_1\sin\theta_1)\\
&J_1J'_1(0, r_1, 0)
\end{aligned}$$

$$\begin{aligned}
OJ'_1 &= OF_1 + F_1J_1 + J_1J'_1\\
\Rightarrow OJ'_1&(0, r_1 - R_1 - L_1\cos\theta_1, -L_1\sin\theta_1)
\end{aligned}$$

J'_2 and J'_3 are get by coordinate rotation:

$$OJ'_2([r_2 - R_2 - L_2\cos\theta_2]\cos\frac{2\pi}{3}, [r_2 - R_2 - L_2\cos\theta_2]\sin\frac{2\pi}{3}, -L_2\sin\theta_2)$$

$$OJ'_3([r_3 - R_3 - L_3\cos\theta_3]\cos\frac{4\pi}{3}, [r_3 - R_3 - L_3\cos\theta_3]\sin\frac{4\pi}{3}, -L_3\sin\theta_3)$$

$E(X, Y, Z)$ is calculate by resolving the sphere interactions equation:

$$\begin{cases}
(X - x_{J'_1})^2 + (Y - y_{J'_1})^2 + (Z - z_{J'_1})^2 = l_1^2\\
(X - x_{J'_2})^2 + (Y - y_{J'_2})^2 + (Z - z_{J'_2})^2 = l_2^2\\
(X - x_{J'_3})^2 + (Y - y_{J'_3})^2 + (Z - z_{J'_3})^2 = l_3^2
\end{cases}$$

$$\Rightarrow Z = -\frac{b + \sqrt{d}}{2a}; \quad X = \frac{a_1Z + b_1}{dnm}; \quad Y = \frac{a_2Z + b_2}{dnm}$$

with

$$\begin{aligned}
w1 &= y_{J'_1}^2 + z_{J'_1}^2\\
w2 &= x_{J'_2}^2 + y_{J'_2}^2 + z_{J'_2}^2\\
w3 &= x_{J'_3}^2 + y_{J'_3}^2 + z_{J'_3}^2\\
dnm &= (y_{J'_2} - y_{J'_1})x_{J'_3} - (y_{J'_3} - y_{J'_1})x_{J'_2}\\
a1 &= (z_{J'_2} - z_{J'_1})(y_{J'_3} - y_{J'_1}) - (z_{J'_3} - z_{J'_1})(y_{J'_2} - y_{J'_1})\\
b1 &= -\frac{(w2 - w1)(y_{J'_3} - y_{J'_1}) - (w3 - w1)(y_{J'_2} - y_{J'_1})}{2}\\
a2 &= -(z_{J'_2} - z_{J'_1})x_{J'_3} + (z_{J'_3} - z_{J'_1})x_{J'_2}\\
b2 &= \frac{(w2 - w1)x_{J'_3} - (w3 - w1)x_{J'_2}}{2}\\
a &= a1^2 + a2^2 + dnm^2\\
b &= 2(a1b1 + a2(b2 - y_{J'_1}dnm) - z_{J'_1}dnm^2)\\
c &= (b2 - y_{J'_1}dnm)^2 + b1^2 + dnm^2(z_{J'_1}^2 - \left(\frac{l_1 + l_2 + l_3}{3}\right)^2)\\
d &= b^2 - 4ac
\end{aligned}$$

8.1.3 Triangulation equation

The triangulation equation determines the coordinates of an object in a 3D space from its position in several images. For this, first, the distortion has been corrected using the intrinsic parameters, and the images rectified with the extrinsic parameters.

Let represent the focal length f, the distance between the cameras' optical center b, and the width and the height w and h. Consider u_1 and u_2 as the 2D homogeneous vectors, capturing the object's coordinates within images left and right, respectively.

$$Z = \frac{b \cdot f}{X_{u1} - X_{u2}}$$

$$X = \frac{Z \cdot (X_{u1} - w/2)}{f}$$

$$Y = \frac{Z \cdot (X_{u1} - h/2)}{f}$$

Here, X, Y, and Z are the spatial coordinates of the object.

8.2 Gantt diagram

	Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	Monday	13/03	20/03	27/03	03/04	10/04	17/04	24/04	01/05	08/05	15/05	22/05	29/05	05/06	12/06	19/06	26/06	03/07	10/07	17/07	24/07	31/07	07/08	14/08
	Friday	17/03	24/03	31/03	07/04	14/04	21/04	28/04	05/05	12/05	19/05	26/05	02/06	09/06	16/06	23/06	30/06	07/07	14/07	21/07	28/07	04/08	11/08	18/08
Delta arm calibration	check literature (method and parameters)	🟩	🟩																					
	identify our own parameters	🟩																						
	Create a simulation of the arm	🟩																						
	Create a model to improve the parameters	🟩	🟩																					
	How accurate do we need the arm to be in his workspace		🟩																					
	Try in simulation several data set		🟩																					
	Find the one that gets the good parameters fast		🟩	🟩																				
	Measure the current accuracy in the workspace						🟩	🟩	🟩															
	Get a real data set								🟩															
	Get parameters with the real dataset								🟩	🟩														
	How accurate is the arm in his workspace after calibration (without noise)									🟩	🟩													
	How accurate is the arm in his workspace after calibration (with noise)										🟩	🟩												
Measurement method	Read out arm angles		🟩	🟩	🟩	🟩																		
	Check literature (measurement method)		🟩	🟩																				
	Measure arm position			🟩	🟩	🟩																		
	Find the accuracy of this measurement (have to be =< than the arm needed accuracy)				🟩	🟩																		
	Defind a method for getting the arm accuracy					🟩	🟩																	
	Make this process easy to use					🟩	🟩	🟩																
Delta arm homing	Check literature (Visual detection)												🟩											
	How accurate does the homing need to be												🟩											
	Get the arm into camera view safely (for user and machine)												🟩	🟩										
	How accurate the position measurement need to be												🟩	🟩										
	Measure arm position with camera													🟩	🟩									
	How accurate is this measurement														🟩	🟩								
	Simulate the homing															🟩	🟩	🟩						
	Create a model to improve the parameters																	🟩						
	Try in simulation several data set																	🟩	🟩					
	Find the one that gets the good parameters fast																		🟩	🟩				
	Get a real data set																			🟩				
	Get parameters with the real dataset																			🟩	🟩			
	How accurate is the homing after calibration (without noise)																				🟩	🟩		
	How accurate is the homing after calibration (with noise)																				🟩	🟩		
Planning and report	Discover the project	🟩																						
	Identify goals/ make the planning	🟩	🟩																					
	Write intro	🟩	🟩	🟩	🟩																			
	Write report																🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

References

[1] *GTRIC proximity switch M12 series Sensing distance 2mm Or 4mm DC 10-36V PNP NPN Smart Metal Inductive proximity sensor*. URL: https://www.alibaba.com/product-detail/GTRIC-proximity-switch-M12-series-Sensing_1600515487637.html.

[2] Z. Roth, B. Mooring, and B. Ravani. "An overview of robot calibration". In: *IEEE Journal on Robotics and Automation* 3.5 (1987), pp. 377–385. DOI: 10.1109/JRA.1987.1087124.

[3] Fan Liang Zhi A.Y. Elatta Li Pei Gen, Yu Daoyuan, and Luo Fei. "An Overview of Robot Calibration". In: *Information Technology Journal* 3 (2004), pp. 74–78.

[4] M.Becquet J.M Renders E.Rossignol and R.Hanus. "Kinetic Calibration and Geometrical Parameter Identification for Robots". In: *IEEE Journal on Robotics and Automation* 7.6 (1991), pp. 721–731.

[5] L. Notash and R.P. Podhorodeski. "Kinematic calibration of parallel manipulators". In: *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century.* Vol. 4. 1995, 3310–3315 vol.4. DOI: 10.1109/ICSMC.1995.538296.

[6] Ana C. Majarena et al. "An Overview of Kinematic and Calibration Models Using Internal/External Sensors or Constraints to Improve the Behavior of Spatial Parallel Mechanisms". In: *Sensors* 10.11 (2010), pp. 10256–10297. ISSN: 1424-8220. DOI: 10.3390/s101110256. URL: https://www.mdpi.com/1424-8220/10/11/10256.

[7] P. Renaud et al. "Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology". In: *IEEE Transactions on Robotics* 22.1 (2006), pp. 12–22. DOI: 10.1109/TRO.2005.861482.

[8] *OAK-FFC-ToF-VGA camera module specifications*. 2022. URL: https://docs.luxonis.com/projects/hardware/en/latest/pages/DM0256.html.

[9] Nirav R. Shah Dhaval K. Patel Pankaj A. Bachani. "Distance Measurement System Using Binocular Stereo Vision Approach". In: *International Journal of Engineering Research and Technology (IJERT)* (2013).

[10] K.Michail C.Brennan C.Sabrina A.Anand W.Camden and V.Nikolaos. "Fiducial Markers for Pose Estimation". In: *Journal of Intelligent & Robotic Systems* (2021). ISSN: 1573-0409. DOI: https://doi.org/10.1007/s10846-020-01307-9.

[11] A.Haggenmiller E.Olson M.Krogius. "Flexible Layouts for Fiducial Tags". In: (2019).

[12] Hong Liu Yang Liu Zongwu Xie. "Fast and robust ellipse detector based on edge following method". In: (2019). DOI: https://doi.org/10.1049/iet-ipr.2018.5687.

[13] E.Olson. "AprilTag: A robust and flexible visual fiducial system". In: (2011).

[14] Carter Kelly Benjamin Wilkinson Amr Abd-Elrahman Orlando Cordero H. Andrew Lassiter. In: *Accuracy Assessment of Low-Cost Lidar Scanners: An Analysis of the Velodyne HDL–32E and Livox Mid–40's Temporal Stability* (2022).

[15] *HDL-32E velodyne lidar definition*. URL: https://velodynelidar.com/wp-content/uploads/2019/12/97-0038-Rev-N-97-0038-DATASHEETWEBHDL32E_Web.pdf.

[16] A. Bellettini and M.A. Pinto. "Theoretical accuracy of synthetic aperture sonar micronavigation using a displaced phase-center antenna". In: *IEEE Journal of Oceanic Engineering* 27.4 (2002), pp. 780–789. DOI: 10.1109/JOE.2002.805096.

[17] *Ping Sonar Altimeter and Echosounder*. URL: https://bluerobotics.com/store/sensors-sonars-cameras/sonar/ping-sonar-r2-rp/.

[18] Sergei Ivanov Vladimir Kuptsov Vladimir Badenko and Alexander Fedotov. "Method for Remote Determination of Object Coordinates in Space Based on Exact Analytical Solution of Hyperbolic Equations". In: (2020).

[19] Jinling Wa n Yanming Fen. "GPS RTK Performance Characteristics". In: (2008).

[20] Christian Thamm Herbert Daxauer Thomas Mayer. In: *Method for the determination of the 3D coordinates of an object* (2010).

[21] *Dual Ultrasonic Sensor Module features.* URL: https://asset.conrad.com/media10/add/160267/c1/-/en/001616245ML01/gebruiksaanwijzing-1616245-iduino-st1099-ultrasoonsensor-1-stuks.pdf.

[22] Tian Huang Pujun Bai Jiangping Mei. "Kinematic Calibration of Delta Robot using Distance Measurements". In: ().

[23] Pascal Brand and Roger Mohr. "Accuracy in image measure". In: *Videometrics III*. Ed. by Sabry F. El-Hakim. Vol. 2350. International Society for Optics and Photonics. SPIE, 1994, pp. 218–228. DOI: 10.1117/12.189134. URL: https://doi.org/10.1117/12.189134.

[24] Lu Yang et al. "Analysis on Location Accuracy for the Binocular Stereo Vision System". In: *IEEE Photonics Journal* 10.1 (2018), pp. 1–16. DOI: 10.1109/JPHOT.2017.2784958.

[25] J.J. Aguilar, F. Torres, and M.A. Lope. "Stereo vision for 3D measurement: accuracy analysis, calibration and industrial applications". In: *Measurement* 18.4 (1996), pp. 193–200. ISSN: 0263-2241. DOI: https://doi.org/10.1016/S0263-2241(96)00065-6. URL: https://www.sciencedirect.com/science/article/pii/S0263224196000656.

[26] David Daney et al. "Interval method for calibration of parallel robots: Vision-based experiments". In: *Mechanism and Machine Theory* 41.8 (2006). Special issue on CK 2005, International Workshop on Computational Kinematics, pp. 929–944. ISSN: 0094-114X. DOI: https://doi.org/10.1016/j.mechmachtheory.2006.03.014. URL: https://www.sciencedirect.com/science/article/pii/S0094114X0600067X.

[27] H Ingensand T Kahlmann Fabio Remondino. In: *Calibration for increased accuracy of the range imaging camera SwissRanger* (2005).

[28] Cheng ML Kurillo G Hemingway E and Cheng L. In: *Evaluating the Accuracy of the Azure Kinect and Kinect v2. Sensors* (2022).

[29] Jamshed Iqbal Toufik Bentaleb. "On the improvement of calibration accuracy of parallel robots - Modeling and optimization". In: *Journal of theoretical and applied mechanics* (2019).

[30] J. Wang and O. Masory. "On the accuracy of a Stewart platform. I. The effect of manufacturing tolerances". In: *[1993] Proceedings IEEE International Conference on Robotics and Automation*. 1993, 114–120 vol.1. DOI: 10.1109/ROBOT.1993.291970.

[31] Jean -Marc Lavest, Marc Viala, and M. Dhome. "Do we really need an accurate calibration pattern to achieve a reliable camera calibration?" In: *Computer Vision — ECCV'98*. Ed. by Hans Burkhardt and Bernd Neumann. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 158–174. ISBN: 978-3-540-69354-3.

[32] Satoshi Suzuki and KeiichiA be. "Topological structural analysis of digitized binary images by border following". In: *Computer Vision, Graphics, and Image Processing* 30.1 (1985), pp. 32–46. ISSN: 0734-189X. DOI: https://doi.org/10.1016/0734-189X(85)90016-7. URL: https://www.sciencedirect.com/science/article/pii/0734189X85900167.

[33] Roboert B. Fisher Andrew W. Fitzgibbon. "A Buyer's Guide to Conic Fitting". In: (1970). URL: https://www.researchgate.net/publication/2237785_A_Buyer%27s_Guide_to_Conic_Fitting.

[34] R. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses". In: *IEEE Journal on Robotics and Automation* 3.4 (1987), pp. 323–344. DOI: 10.1109/JRA.1987.1087109.

[35] Zhenzhong Xiao et al. "An accurate stereo vision system using cross-shaped target self-calibration method based on photogrammetry". In: *Optics and Lasers in Engineering* 48.12 (2010), pp. 1252–1261. ISSN: 0143-8166. DOI: https://doi.org/10.1016/j.optlaseng.2010.06.006. URL: https://www.sciencedirect.com/science/article/pii/S0143816610001247.

ENSTA Bretagne

ASSESSMENT REPORT

This report should be signed, endorsed with company stamp by the internship supervisor and returned into aurion “Mise à jour de PFE”

Host organisation	Odd.Bot
Dates of the internship	13/03/2023 - 18/08/2023
Name of the intern	Rémi Porée

	Please tick the corresponding boxes					
	F	E	D	C	B	A
Personal and interpersonal skills						
Adaptability						X
Availability						X
Corporate culture					X	
Work commitment						X
Language skills (oral/written)					X	
Project management						
Task identification					X	
Organization / distribution of tasks over internship period						X
Respect of deadlines (deliverables, reports, etc.)					X	
Proactivity						X
Potentially: team work						X
Internship report						
Format, style, etc.						X
Content						X
Usability for company/laboratory						X
Appreciation of ENSTA Bretagne Training						
Scientific and technical skills meet my expectations						X
The methodological skills meet my expectations					X	
On what subject did the student have to be trained before he became autonomous ?	None					
Which skills or training content should be reinforced ?	Prioritization of tasks, based on importance, time and chance of success (rather than what is interesting)					

Overall appreciation

Rémi is very hard working and very motivated. He was able to take our goal for the internship, use that to set up his own planning, and follow this. He required little supervision (1hr/week), and in this time he could present the current status and propose an action plan for the next period.
Overall, a great intern who did very useful work.

In the event of a position becoming available, would you consider employing this student?
☒ YES
☐ NO

Name of the supervisor: Ewoud Pool **Date: 13/08/2023**

Title/Position: CTO **Signature:** Eai