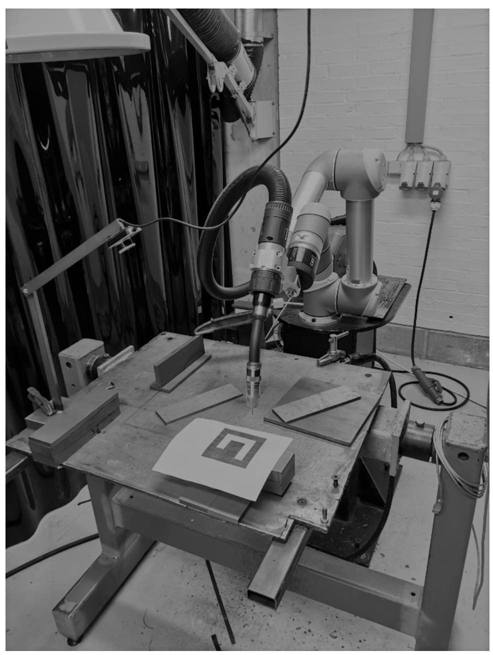
Automatic Pre-Production for Robotic Welding

– And Dynamic Calibration for it –

Project Report



Aalborg University
Department of Materials and Production
Fibigerstræde 16
DK-9220 Aalborg

AALBORG UNIVERSITY
STUDENT REPORT

Manufacturing Technology
Department of Materials and Production
Fibigerstræde 16
DK-9220 Aalborg
www.mp.aau.dk/

Title:
VT4-Project

Theme:
Robotic Welding

Project Period:
From February the 1^{st} 2022
To June the 2^{nd} 2022

Project Group:

Participant(s):
Peter Tybjerg Sørensen

Supervisor(s):
Poul Kyvsgaard Hansen

Copies: 1

Page Numbers: 65

Date of Completion:
June 2, 2022

Abstract:

This report has been made in collaboration with Sjørring Maskinfabrik and PDM Technology. It explores the issues concerning robotic welding and how automatic pre-production of it can facilitate more robotic welding in low volume, high variance productions. An automated offline programming model is proposed and examined with the specific type of production companies in mind for how to automatically go from a CAD model of a product to having a robot welding on a workpiece. The model utilises the concept of elementary operations which holds information about how to weld a small part of the product. Further work is conducted into how this company type will cope with introducing more robotic welding without using fixtures. A sensor system is proposed consisting of a camera and ArUco markers to dynamically and automatically calibrate the robot to a workpiece. Calibration of the camera is performed to acquire specific camera parameters later used for pose estimation of an ArUco marker. The dynamic calibration using a marker works, but is, as of the results in this report, not precise enough to be used for robotic welding yet.

Contents

Preface

This project was made by 4th semester Master student Peter Tybjerg Sørensen, from Manufacturing Technology (VT4) at Aalborg University from the 1st of February to the 2nd of June 2022. This project was supervised by Poul Kyvsgaard Hansen (Department of Materials and Production), and made in collaboration with PDM Technology and Sjørring Maskinfabrik.

Signatures

Peter Tybjerg Sørensen

Peter Tybjerg Sørensen

Signature

Part I

Introduction

Chapter 1

Introduction

This project has been made in collaboration with "Bluestar PLM by PDM Technology" and "Sjørring Maskinfabrik" as a small part of a MADE project (Manufacturing Academy of Denmark) in collaboration with a PhD project, Sarivan et al.[1]

1.1 PDM Technology

PDM Technology is a company with its main location in Aalborg in NOVI Science Park. It is a company with around 50 people located in Denmark, Czech Republic, USA and Germany. It was founded in 2001 by Jørgen Schiønning Larsen. It is a tech company, who offers their own unique software solution, "Bluestar PLM", a product lifecycle management (PLM) tool, and the implementation of it. The customers of this product are manufacturing companies, who look to move their production towards a more digitised direction. PLM software helps with standardising procedures and tracking product data, such as CAD files, Bill of Materials (BOMs), etc. Bluestar PLM differentiates itself by being a Microsoft Partner and has embedded their solution within Microsoft Dynamics 365 and Dynamics AX enterprise resource planning (ERP) systems, as an additional module to the system. This offers seamless integration of operations, finance, engineering and manufacturing and simplifies the digital communication between the departments. Bluestar PLM is compatible with more than eight CAD software system, making it possible for most companies to seamlessly continue working with their preferred system when integrating with Bluestar PLM[2][3][4][5][6].

1.2 Sjørring Maskinfabrik

"Sjørring Maskinfabrik" (Sjørring Machine Manufacturer) is a company located in Sjørring, in the north western part of Denmark. It was founded in 1946 and their CEO is Klaus Kalstrup. They specialise in manufacturing of buckets/attachments/shovels for heavy machinery including wheel loaders, excavators, and buckets for the mining industry. It is an engineering to order (ETO) business, meaning the customer request a customised order, before they at Sjørring Maskinfabrik start to produce it. A large quantity of their orders come from "Volvo Entreprenørmaskiner A/S" (Volvo Construction Equipment). Their production consists of multiple processes for the manufacturing of these buckets including welding. The products ordered are variations of the three bucket types, meaning a lot of them are brand new buckets, that they have to create. If the order volume of a shovel is less than five, they typically produce them through manual welding. Only if the volume is larger, they start to consider robotic welding. This is due to the time it takes to create a robot program from scratch[7][8].

The information used in this report regarding Sjørring Maskinfabrik's production have been gathered through multiple online meetings, written communication, and a visits to their factory. In this report Sjørring Maskinfabrik will sometimes simply be referred to as "Sjørring".

1.3 Digital Manufacturing

Manufacturing is the conversion of raw materials into finished goods and products. Manufacturing companies' purpose is to create products through one or a series of processes. Products can be very complex like cars, or simple, like a plastic spoon[9]. Companies can operate in different ways depending on the type of product they create. Mass production, where a large volume of the same product is produced with a specialised and integrated series of processes. Or job shops, where a low volume of many product variations are produced in a flexible production that can handle many different cases with low change over time. Examples of processes could be assembly, drilling or welding.

Digital manufacturing revolves around applying computer systems and digital technologies in the production, supply chain, product and processes. Using for instance simulation and modelling, the output and quality can be increased, and cost and time to market for new products can be decreased. A production can be simulated to find and better cycle times or flow of it. Simulations can also be used for robotics, eliminating the need to halt production when programming it. Sensors have become better and cheaper and can now be used all over a factory, both big and small, to create a network of communication between the machines in the factory floor and can feed real-time data to computers that utilises the digitisation and aids the automation of production[10][11]. Another example is Computer-Aided Design (CAD) models, which is a geometric representation of a product or workpiece in 3D with millimetre precision. CAD software are used in manufacturing by engineers designing new products or altering already existing ones. However, a significant area where the digital development has been a bit slower, is streamlining the processes used in production. Currently CAD models only encompass product geometry, but nothing in terms of which processes are applied to create the product, if and how drilling operations will handle a certain hole or how a certain edge shall be cut. And when it comes to a very complex process, such as welding, it can be difficult to represent not only the explicit knowledge, but also all the tacit knowledge hidden in the experience of professional welders[6]. This is an area which could greatly benefit industry if explored further.

1.4 Outline

This project will investigate how robotic welding is currently conducted and what kind of challenges companies face when implementing it. It will focus both on what steps to take in order to automate the preparation of robotic welding and how it will then be executed on the robot. In Part II, first welding and robotic welding will be explored, as well as the pre-production of the latter. Following this a model is proposed and presented with steps to undergo in order for automation of this process can take place. This part will end with a description of the hardware and software used for this project and a problem formulation. Part III will dive into the model from the perspective of a low volume ETO production company and find out where the limitations are regarding automation of robotic welding pre-production currently. This will also serve as a jumping-off point for diving into concretely dealing with one of the limitations on the execution side of automatic production preparation, namely dynamic calibration without a fixture for robotic welding. A sensor system is put forth with a camera, camera stand and a fiducial marker. In Part IV calibration of first the camera and then the robot with the marker is undergone. This is concluded in Part V.

Part II

Problem Analysis

Chapter 2

Context Analysis

2.1 Welding

The welding process refers to the fusing of two parts or workpieces, usually metals. This is done by applying heat or pressure or both. When cooled the two parts have become one, connected with a weld bead or joint[12]. This was already a phenomenon during the Iron and Bronze Age, which consisted of hammering the two parts until enough heat was present that the welding would occur. The beginning of traditional welding methods started in early 19th century. It began with the use of a battery and two carbon electrode creating the first electrical arc[13][14]. Today, welding encompasses a variety of methods that can all be put into five broad categories. Arc welding, oxy-fuel welding, resistance welding, solid state welding, and other types of welding[15]. The last one features all other types not encompassed by the previous four, for instance laser beam welding. This report will mainly deal with a specific type of arc welding, gas metal arc welding (GMAW), also often referred to as metal inert gas (MIG) welding.

2.2 Robotic Welding

Automated welding started to emerge in the 1920s and has since become a huge focus for industry[14]. In 2017, 82,000 of 381,000 industrial robots shipped were meant for welding and soldering[16]. Automatic welding is implemented in the form of either robotic industrial manipulators or automatic welding machines, where the latter can be the preference for a large quantity of typical small workpieces, whereas robots offer more flexibility with their welding capabilities. A further distinction for an automated solution can be made with either using semiautomatic welding and fully automatic welding. Whereas semiautomatic welding incorporates a worker to load and unload workpieces as well as run the welding program, fully automatic welding can run without any interference from humans for an extended period of time. Depending on the use case one or the other can be preferable. Fully automatic welding is typically a too large step to make when converting from manual welding and semi-automatic can be used as a stepping stone. Or it might simply not make sense fiscally for the company to implement a fully autonomous welding setup [17][18].

Robotic welding can take the form of a variety of welding method. The most common practice is resistance spot welding, since it is used car manufacturing. Laser welding is becoming increasingly popular for robotics, due to it being a non-contact process. Besides this, arc welding is also widely used for robotic welding, especially GMAW (MIG), because of its speed, when working with thick metal workpieces, and is great for robotics instead of humans, due to the gas and vapors, that occur during the process[19][20][21].

There are a few different ways of programming the robots. With online programming, a worker operates the robot using the appertaining teach pendant. Here the worker can move the robot to certain positions and record the program by declaring what type of movement the robot should make to move between the points, as well as, what times the welding torch are on and off. Teach-in programming utilises the welders experience in hands-on welding by letting them move the robot itself and register the different points to go through and when to weld using either a controller tool attached with the end-effector or the teach pendant. Offline programming is achieved using software tools to create a program in a virtual environment. This can be done on a separate computer and uploaded to the robot, which have the benefit of providing zero operating down time when creating and testing the program. One of the great benefits using this method is that it can be done concurrently with the ongoing production [22][23].

Advantages

The reasons for implementing automated welding, and specifically robotic welding are many. The American Welding Society predicts there will be a shortage of professional welders by upwards of 400,000 by 2024 in the United States alone [24]. And the problem has existed for while, with 40% of manufacturing companies having to decline new contracts, due to the deficit in skilled employers in 2014 [22].

A potential way of meeting the increasing welding demand for the industry is by utilising robotic welding. The benefit of using robots can be found in their precision and repeatability. Most industrial manipulators have a repeatability below 0.1 mm [25], which makes them more precise than humans. This leads to improved quality, due to the weldment being consistently applied correctly on the workpiece, while also creating less scrap due to the higher accuracy and precision. This also creates a financial incentive, since less products and resources will be destroyed in the welding process, and therefore also less time wasted spent on welding something that later will be scrapped. The repeatability also creates a more uniform production, where the same weld will be executed in the exact same time, using the exact same resources, which makes it easier to plan around throughout the entire production pipeline.

For the production itself, robots are able to decrease the process time, due to them operating faster than humans and therefore facilitates a higher greater output of products, if the welding process is the bottleneck. Distractions and human fatigue can also be disregarded with the use of robotics, which can affect both the processing time and the quality of the weld. On top of this, robots can, in theory, run 24 hours a day. So in the case where the system does not require human supervision, it has the potential for significantly more operating time per day, than a human welder.

In terms of safety, some welding processes creates toxic gasses, which human welders are exposed to. Moreover, some of the weld work to be executed can require workers to place themselves and weld in awkward and difficult positions, which potentially can be damaging over time, without a proper ergonomic setup. With robotic welding the need for workers to operate in these environments are at least reduced and potentially eliminated. Additionally, welding related injuries and hazards will also decrease, and machines can always be repaired or replace [18][26].

When using an automated welding system the actual welding can run continuously, which not only conserves time but also energy for the same amount of welding due to cutting energy expensive start-ups of a welding system [19]. And since the precision is higher and less error prone than manual welding, energy will be saved due to a reduction in the need for corrective welds. On top of this, using robotic welding decrease the labor cost, since the only expense of having a robot weld, compared to manual welding, is the energy it consumes.

Disadvantages

Now, that there are so many benefits when incorporating welding robots into the production line, why are they not in every production with welding processes? There are a few reasons as to why that is the case. Currently, robotic welding is not a fully mature technology yet. The features that are possible to weld are limited, with robots mainly being used for welding straight lines and soft curves. When the geometry of products are complex and the weldments complicated, manual welding is selected to perform the task. In some cases with large and complex products, in order to utilise robotic welding both pre and post manual weldments are applied in order to produce the fully finished product. These issues make it difficult to fully automate welding for a product as soon as the complexity of its geometry starts to increase.

Setting up a robotic welding system will take a while initially. Not only to have it sat up physically, but also the time needed to program the robot to execute the welding process. Therefore it is best suited for a large volume of the same product, since the program creation is a one time action and can be used again and again without extra time cost. This is also the reason why robotic welding is rarely well suited for one-of-a-kind or batch production. The time spent on programming the robot and then execute the welding can almost never be justified compared to the time a professional welder can complete the task, when the product volume is low [16]. Additionally, flexibility is sacrificed for higher precision, repeatability and faster welding pace, which is one of the main problems with robotic welding systems. According to the National Institute of Standards and Technology, the challenge of flexibility is the major reason, that 90% of manufacturing company have not acquired robots for their welding needs [22].

To utilise the high precision and accuracy of robots, the physical workspace and environment around them needs to correlate exactly to how it is programmed. Any changes to this in the real world will be neglected by the robot unless the programming is updated. To avoid this or any mistakes in the process fixtures are typically used. The fixtures will hold the workpiece in place at the precise spot the robot was calibrated to. They also serve the other purpose of eliminating the expansion of the workpiece, that occurs when heat is applied via the welding torch. Without the fixtures to keep the workpiece in place, a mismatch between the real world and the virtual world will occur and the weld quality will suffer, if not be completely useless. This mismatch will naturally be countered by a professional worker with the use of experience and senses. The issue with these fixtures are, that creating them can be complicated and expensive, and for one-of-a-kind or batch production tend to not be a feasible solution compared to manual welding, and a fully mature real-time validation system for thermal expansion is yet to be created as an alternative to fixtures.

The flexible solutions that collaborative robots can provide can not be used with welding, due to safety reasons. Having a cobot with a welding torch as an end-effector removes the ability to work in collaboration with it. When implementing an industrial manipulator, a safety cage has to be installed around it, which not only complicates the interaction with it, but also takes up a large amount of space in the factory. On top of this, it is also expensive and adds to the already high initial investment of acquiring a robotic welding system, in comparison to just hiring a professional welder [18][26].

Advantages	Disadvantages

• There is a shortage of professional welders
• Higher quality –> Better:
  – Repeatability
  – Precision
  – Accuracy
• Faster operating speed
• Less scrap
• Can work 24/h/day in theory
• Safety –> Humans less exposed to:
  – Bad ergonomics
  – Toxic gasses
  – Weld injuries
• No salary cost
	• Product geometry limited to:
	  – Straight lines
	  – Curves
	• Difficult for complex welds
	• Requires custom fixtures
	  – Many types of welds are currently impossible
	• High setup time
	  – For physically setting the system up
	  – For programming the robot
	• Less flexible
	• Safety cage required
	• High initial cost

Table 2.1: Summery table of robotic welding, pros and cons

2.3 Automated Pre-Production

Most of the focus for implementing automation has been on production. How to implement conveyor belts, CNC machines, robots, etc. This is typically very beneficial for scenarios, where a large volume of a product is desired. A great amount of time and energy is used on setting the system(s) up, put up equipment, planning, and programming the setup, the pre-production aspects. However, this investment becomes more and more favourable, the more products the system will be able to output over time. This incentive to fill the factory line with automation usually vanished for companies producing many different products at low volume, such as batch productions and one-of-a-kind production. Here the much more flexible workers are much better at adapting to a variety of product within their process specifications, and thereby minimising the time spent on production preparation. However, if there was a way of automating this preparation, automation in production could become viable for low volume productions.

Currently, automatic pre-production can mostly be found in product configuration, where a specific product's CAD model can be changed in an instant using a designed product configurator. The model is parameterised and the user can set the values they want for their product for the specified parameters, for instance, if a longer version of the product is desired. This allows for customisation of the product, so as to satisfy the customers' needs better.

Automated Pre-Production for Robotic Welding

Automatic pre-production for robotic welding is an area with huge potential, which have not been in earnest tapped into yet. There are different ways of arriving to the point of the actual welding process execution based on the method used (online programming, offline programming, teach-in programming). Automating the online and teach-in methods are difficult to automate due to their nature of involving a person to manually move the robot with the teach pendant or move the robot physically to record the movements for the weld. However, offline programming has the potential.

One of the main time consuming tasks in offline programming is creating the program for the welding robot when a 3D model of the product arrives. Creating both the many points the robot has to move through during the welding phase but also the approach and removal points for each weld. The larger and more complex of a product the more welds, and therefore more time spent on creating the program. And this is not even including all the calibration points, that needs to be included, to make sure the robot operates in the correct and exact places. Following this, the robot programmer will have to determine the order in which the different weld segment needs to be carried out. When implemented in the robot welding software, the kinematics and welding process for the robotic system is generated. Before executing the program some small adjusting can be made to calibrate correctly and correct some trajectories to avoid unwanted movement or collisions. Then the welding process is ready to be executed. In the meantime the workpiece has been made ready by being placed in its fixtures perhaps with some pre-welding done manually for certain spots the robot will not be able to weld. At Sjørring Maskinfabrik, they can spend up to a week on creating a robot program for welding their buckets. But the robot welding process itself only requires around 6-7 hours. And since a lot of their product orders are less than five of the same bucket variation, manual welding are often preferred[8]. Therefore, automating even some of these production preparation procedures described will both save time and free up workers.

2.4 Automated Offline Programming System (AOLP)

Creating a generalised method for automating the production preparation of robotic welding described in section 2.3 can be troublesome due to the complexity of the welding process and the many varieties of executions and scenarios it should be implemented in. Figure 2.1 presents five key steps to undergo for generally automating this pre-production, which this report proposes. The model assumes a CAD model of the product is given/created.

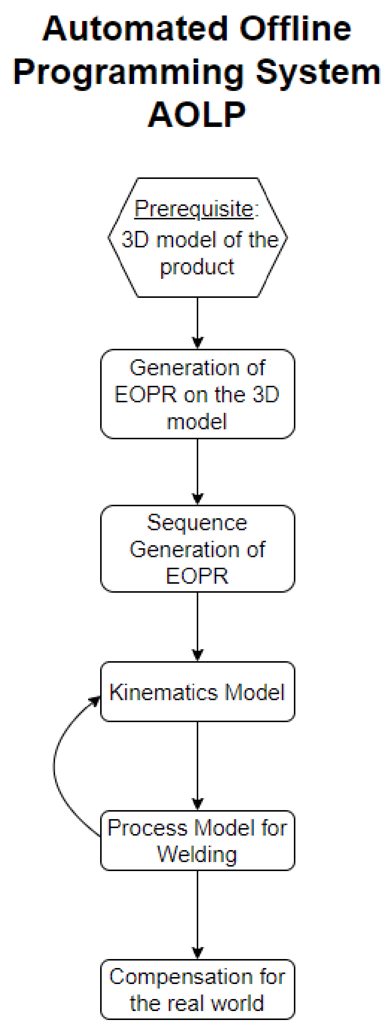


Figure 2.1: Automated Offline Programming System pipeline

As stated by Nathan Larkin et al.: *"AOLP (Automated Offline Programming) technology is critical into reducing the cost of joining materials with arc welding processes where batch volumes are low and allow SMEs with high labour cost economies compete in the global marketplace."*[27]
The model presented in Figure 2.1 is based on the ideas presented in the scientific papers "Automatic program generation for welding robots from CAD" published in 2016 by Larkin et al.[27] and "Towards Automatic Welding-Robot Programming Based on Product Model" published in 2021 by Sarivan et al.[1], combined with with research conducted in collaboration with PDM Technology[6].

The practical idea with the model is to present a way of automatically going from a CAD model of a product to a complete robotic welding program ready to be executed on a robot system capable of performing the intended task(s). The AOLP model utilises the concept of Elementary Operations (EOPR), which serves to divide up the welding task into sub-problems. Each elementary operation describes how an isolated feature of the product should be welded, for instance a straight line, a specific corner, a certain curve, etc.

Generation of Elementary Operations on a 3D Model

The first step to be able to automatically generate a robot program based on a 3D model is to have the CAD model divided up into elementary operations. This could appear as an overlay of weld segments being placed on the CAD model in the software. For this function to be as general as possible, machine learning will be required to be fused into this step in order to be able to recognise all types and complexities of geometries presented in product models. This, therefore, requires a large data bank of all types of weldments and knowledge on how a robot will manage to weld them. When the entire workpiece has its welds divided up into elementary operations the next step can take place.

Generation of an Execution Plan/Sequence of the Elementary Operations

With the CAD model divided into welding sub-problems a sequence for executing the elementary operations needs to be generated. To have this sequence be made automatically with the focus of weld quality in mind, it will most likely require a lot of tacit knowledge concerning the specific product, and how best to approach the welding task to, for instance, minimise unwanted thermal expansion. This will, as a result, also require incorporation of machine learning for the decision making. Before this step is fully mature, a robot programmer could be an alternative to handle the creation of the sequence, based on their previous experience on how they have handle the execution plan before the introduction of an automatic pre-production system.

Kinematics Model/Motion Planning

When the sequence is determined, the robot's kinematics can used to generate the motion planning automatically. Out of the five steps, this is currently the most technologically mature, since it is already being used not only for welding, but for every offline programming tasks for robots. A planning algorithm, or "planner", handles the trajectory generation based on the forward and inverse kinematics of the robot and the information from the execution sequence of the elementary operations and ensures that the robot pose is feasible at all times through the welding. This will be an iterative procedure in combination with the process model (step four), since some information and parameters are handled in this step which will influence the kinematics of the robot.

Process Model for Welding

The process model will encompass how each weld should be managed. This includes, torch intensity, the angle of the torch in relation to the workpiece, the distance to workpiece, the specific trajectory required from weld start to end, etc. It is therefore necessary to feed all this information back to the kinematics model to iteratively update the motion planning for the entire welding process. This will consequently require robotic welding to be able to handle all types of geometries and weldments. Otherwise, the types of welds will be limited to straight lines and simple curves, thereby decreasing the generalisable part of the model.

Compensation for the Real World

To harness the potential of the automatically generated program, a system to compensate for the mismatch between the virtual world and real world will have to be incorporated, especially in low volume productions, where specially customised fixtures are typically not profitable. The mismatch can for instance be a small deviation in workpiece placement, where precise and accurate calibration can compensate. Or it can be the case of thermal expansion of the workpiece during the weld, where real-time compensation is required to apply the weld correctly. Many types of sensor systems have the potential to aid in this issue and such a system will be a necessity for completing the whole flow, which the model presents.

The reason for looking into the concept of elementary operations stems from multiple sources. In Larkin et al.[27] "tag creation" is used and refers to placing in small segments on the CAD model which identifies how to weld that certain segment of the product. At Sjørring Maskinfabrik, their procedure for creating a robot program consists of the robot programmer placing in points for the robot which are grouped up in smaller segments around different geometries[8]. And the idea of elementary operation also facilitates parameterisation and automation of it. All of this will be elaborated further in chapter 5.

Presented above is a general model for the steps necessary to automate the pre-production of the robotic welding process, using the concept of elementary operations. This will be expanded upon in chapter 5 where the model will be narrowed down to focus on low volume, complex, parameterised products and workpieces. The following chapter chapter 3 will unveil the tools used in this project.

Chapter 3

Technical Analysis

This chapter will describe the hardware and software used in this project.

3.1 Hardware

For this project, a robot will be used. The robot is a collaborative robot (cobot) from Universal Robots (UR). The model is a UR5, signifying its payload at five kilograms. It is six-axis manipulator, where all joints are rotational. It has a repeatability of ± 0.1 millimetres and a reachable workspace with a radius of 850 millimetres[28].

Figure 3.1: Picture of the UR5 robot

Figure 3.2: Picture of the UR5 robot

Since it is a cobot it is created with the intend of working in combination with people with real-time interactions. It has built in sensors that will register if contacts has been made with either obstacles or humans and will consequently stop its motion and thus avoid damage to itself or its surroundings including people. Cobots therefore, do not require safety cages and are in general inexpensive, more flexible and simpler to program than traditional manipulators[29]. The UR5 can be programmed on-line or offline, with the use of the teach pendant or external devices respectively.

Figure 3.3: Picture of the end-effector welding torch

The end-effector of the robot is a welding torch from Abicor Binzel. The model is a W500 welding torch [30], visualised in Figure 3.3.

3.2 Software

Python

In the 1980s, Guido van Rossum started developing what is now known as Python, later realising it in 1991. It is one of the most popular programming language due to it being easy to get started with for beginners and versatile, with Python being applied for web, interface, software, and scientific purposes. It has a large quantity of third party libraries, created to be used with Python[31][32][33]. Another reason for its popularity is the integration with machine learning and artificial intelligence, which is supported throughout a number of packages/libraries including OpenCV[34].

OpenCV

Computer vision is an area of computer science regarding images and videos, how to extract and analyse them and, for instance, identify and recognise features or objects in them. As us humans see the world through our eyes, machines and robots require cameras or image sensors to do so. And computer vision serves to process the information obtained[35]. This information can then be used by decision-making algorithms to perform actions based on the image sensor input[36][31]. There are many applications for computer vision. Face recognition for social media apps, surveillance, smart cars for autonomous driving etc[37]. One of the uses tied to this project is robot vision.

The Open Source Computer Vision Library (OpenCV) is an open source library for computer vision and machine learning. It is a software tool that realises computer vision with programming. The development of OpenCV began in the 2000s with Gary Bradsky and Vadim Pisarevsky. The library has 2500+ optimised algorithms, which focus mostly on real-time vision applications, which also will be the use of it in this project. It is natively written in C++, but has interfaces with a number of other programming languages including Python[38][39].

RoboDK

In January 2015, Albert Nubiola founded RoboDK. When launched, it supported 200 robots from 20 robot manufacturers. Today, it has more than 600 robot arms from over 50 manufacturers[40]. RoboDK is a 3D simulation tool for programming robots, both for offline programming and for simulation. It can be used without any prior programming experience using its interactive interface where "Targets" can be placed and a certain movement can be selected to move to that point, collision and singularity free. With the RoboDK API it is possible to program the robots through scripts with a variety of programming languages[41][42]. The scripts created in RoboDK for this project will be using Python programming language. For this project an environment of the robot setup in RoboDK has been made available by MADE (Manufacturing Academy of Denmark). The environment is presented in Figure 3.4.

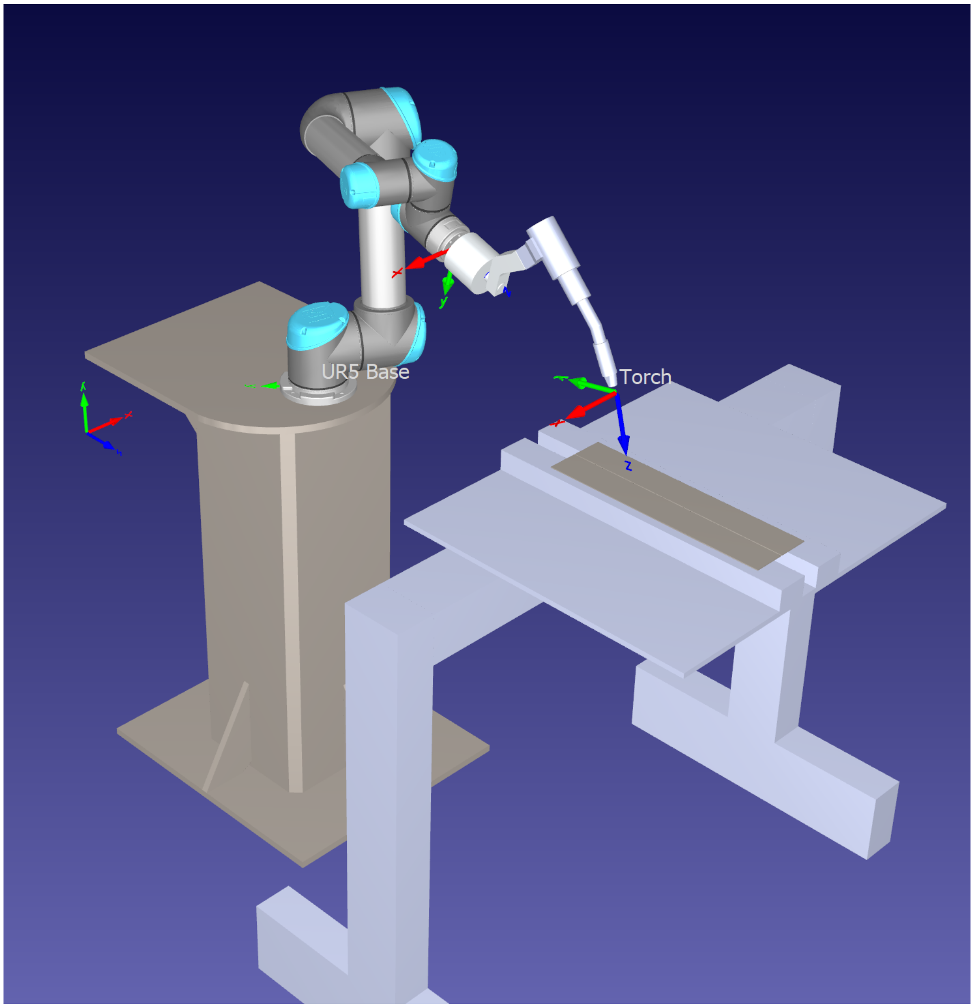


Figure 3.4: The RoboDK environment of the UR5 robot setup

Chapter 4

Problem Analysis Conclusion

This chapter will conclude on the problem analysis, as well as provide a problem formulation and the delimitations of this project.
In section 2.1, baseline of welding was established, which lead into section 2.2, where the background was introduced and the pros of cons of robotic welding put forth, which can be seen in Table 2.1. There is a shortage of professional welders for the amount of welding tasks there are. Introducing robotics could be a solution. On top of this it is more precise and faster than human labour. However, robotic welding can not as of yet handle all of the welding needs, due to it not being able to weld geometries other than straight lines and curves. Furthermore, it is a less flexible solution and requires precise and durable fixtures for a high quality weld. Uncovered in section 2.3 this decrease in flexibility is especially troublesome for ETOs and manufacturers who produce a variety of their products with low volume. The trade-off of the time and resources spent on preparing for robotic welding is never reached for the minimal time it is used in production. Therefore, it might be worth looking into how to automated the pre-production of robotic welding. In section 2.4 a model is proposed for encapsulating the steps necessary to realise that idea, both with a focus on preparation and the measures that needs to be taken in order to then implement it. How to go from a CAD model of a product to have it welded by a robot. The model utilises the concept of elementary operations, which serves to divide up the welding problem and contains all the welding process information for a small part of the product. In section 3.1 and section 3.2 the hardware and software used in this project is explained, namely a UR5 robot combined with Python scripts in RoboDK, a simulation software for programming the robot.

4.1 Final Problem Formulation

The focus of this report will be on outlining a structure for how to approach automatic pre-production of robotic welding from the perspective of low volume, high variety product manufacturers. It will also use this to explore and showcase a small step towards realising this potential on the execution side on the robot as these go hand in hand (elaborated in chapter 5). This leads to these problem statements:

How can the AOLP model proposed be used for streamlining pre-production of robotic welding and where are the current shortcomings?

How can an improvement on the execution side of robotic pre-prodcution in the form of calibration work and which benefits will it bring for automatic pre-production of robotic welding?

This will be further explored in the following part of the project Part IIIPart IVPart V.

4.2 Delimitations

This project is created by a master student at Aalborg University, with a background in robotics. Previous work has mainly been with the focus on robotics. The project is created in collaboration with Sarivan et al. [1], where the technical aspect has been focusing on adapting a CAD model to automatically generate elementary operations. But for this project the technical aspect will focus on the execution side of the robot, due to both topics require some synchronous development, and since this student for this project has no prior knowledge, experience, or competencies within the field of welding nor CAD programming and machine learning.

Part III

Design

Chapter 5

Product Scope

The robotic welding process and the preparation of it is highly complex. It is therefore also difficult to generalise a solution for automating the production preparation of robotic welding. To combat this, the scope for the model will be narrowed down within the type of product to be looked at. For the case of robotic welding, both the process and the preparation process of it is heavily influenced by what kind of products to be manufactured, which can be visualised in Figure 5.1, showing the interdependency between the supply chain, market, process and product. Limiting the product scope could thus present a more generalised method or structure for automating the pre-production of robotic welding.

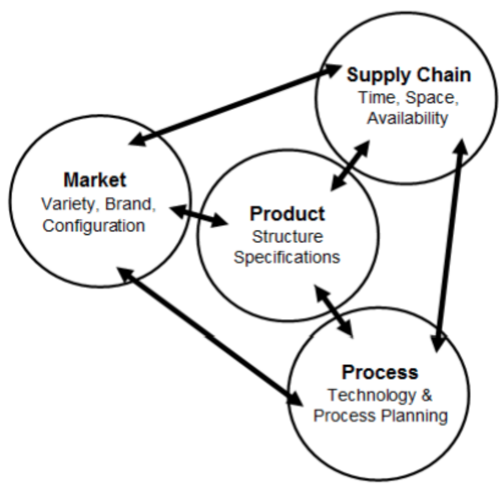


Figure 5.1: The 4D Modularisation Model [43]

This project will focus on low volume, complex products that can be parameterised. It will be productions that typically produce one or a couple of different products that can have many variations based on the same fundamental elements. For production companies fitting this case a lot of time has to be spend on production preparations, creating a robot program for when a customer wants a new product order. And for engineering to order (ETO) companies, this is a regularity, that often lead to them stick with manual welding, since the time spend preparing for the process is significantly less and much more flexible[8]. However, parameterised products, where the fundamental structure is the same, but varied for certain elements, such as width or length, might have the potential to save some preparation time by reusing parts of earlier robot welding programs and fitting it to a new products with slight variations of the previous. And elementary operations may be able to facilitate that.

Product Reference - Bucket from Sjørring Maskinfabrik

In the following sections the different steps of the AOLP model presented in section 2.4 will be evaluated with this product specification in mind. Both what the possibilities are and the limitations as of right now. An example from the industry, "Sjørring Maskinfabrik" (Sjørring Machine Manufacturer), will be used for reference. Sjørring is an engineering to order company, as mentioned in section 1.2, which produces buckets for wheel loaders and excavators that require multiple complex welds using GMAW. Since Sjørring is an ETO business, their orders are customised variations of their base products set by the customer and only after this, will they start the manufacturing process. Thus, the market/-customers play a large part in how their products will turn out, and consequently also how their processes in the production operates (see Figure 5.1). To cope with a varying demand from the market some of the elements of their products, these buckets, are parameterisable[8], where for instance the depth or width of the bucket type can be shortened or extended, based on the customer needs. These variations of their buckets, however, puts up a challenge for the robot programmer, who will have to create a new robot program if the order will undergo automatic welding, thereby having the market and product influence the preparation process to a great extent. An example of their products can be seen in Figure 5.2, Figure 5.3, and Figure 5.4, which shows a CAD model and a context drawing of one of their buckets for excavators.

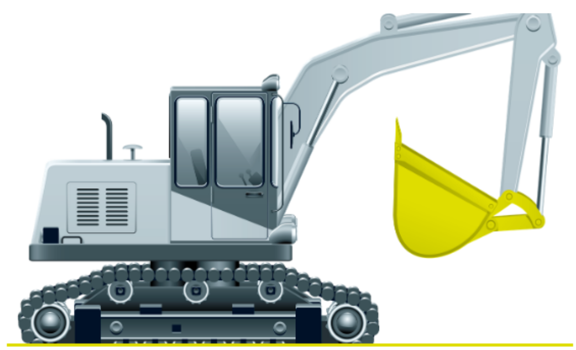


Figure 5.2: A drawing of a generic excavator highlighting in which context the buckets appear in [7]

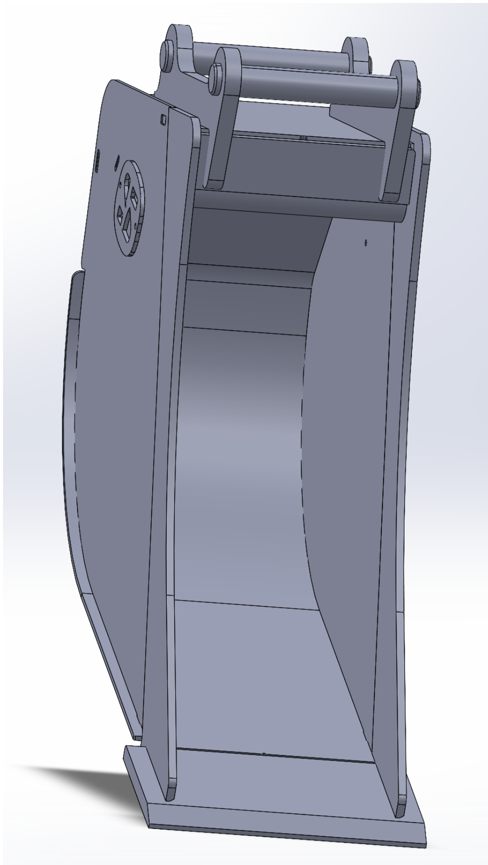


Figure 5.3: A CAD model of a bucket for an excavator produced by Sjørring Maskinfabrik[8]

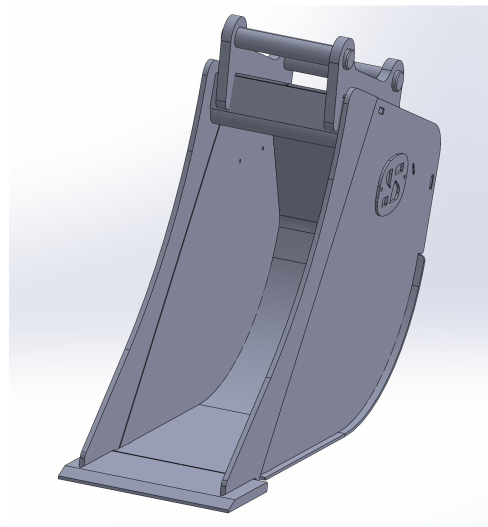


Figure 5.4: A CAD model of a bucket for an excavator produced by Sjørring Maskinfabrik[8]

5.1 Generation of Elementary Operations

When having acquired or created a 3D model of the product at hand, the idea is to have a system in place being able to identify the different geometries of the workpiece and match them with previous data for the same type of corner, curve, etc. The welding problem is thereby divided up in sub-problems, which can have some flexibility in execution order when it comes to the full welding process. To be a complete general solution, information, such as alloy type, plate thickness, weld thickness, weld offset and weaving pattern has to be incorporated[1]. However, this could be circumvented by making it possible to be manually set these parameters, either for the complete model or individual elementary operations.

Literature Informed State of the Art

As mentioned in section 2.4 an AOLP system is presented in the article "Automatic program generation for welding robots from CAD"[27] by Larkin et al. 2016. The model presented starts with acquiring a CAD model of the workpiece needing welding. The following step is "Tag Creation" where "tags" or "process point", which for this instance is weld beads/paths, from the CAD model are extracted, closely resembling the ideas of elementary operations. These tags are created by the end user, where the position, length and weld type is determined by them. This information is then used for the trajectory planning.

The article "Towards Automatic Welding-Robot Programming Based on Product Model" published in 2021 by Sarivan et al.[1] investigates how to decrease the time spend on programming a robot for welding purposes (GMAW) by automating it based on a 3D model. The focus of the article is to overlay weld beads on the inside of an open metal box on a CAD model, where all the sides have to be welded together, as well as to the bottom plate of the box. This is executed in an add-on program for SolidWorks. These virtual weld beads are parameterised so the length of them can be set by the user. These weld beads contain information that, via a python script, can be used to calculate the trajectories of the end-effector. The motion plan is generated in less than a minute and can then be sent to the robot itself.

At Sjørring, when an order comes in, they receive a CAD model of the product from the customer. It sometimes requires some altercations, due to missing some geometries that Sjørring requires to run the processes, which the workpiece require, such as holes or hooks to mount the product. If the volume of the order is less than five they do not consider robotic welding for it and turn to manual welding solely. In the cases where robotic welding is used, it is combined with both pre and post manual welding, elaborated further in section 5.5. From the CAD model a robot programmer creates a welding program for the robot using offline robotic welding software such as Delfoi Arc[44] or Panasonic DTPS[45] depending on, which of their robots it is designed for. It requires the programmer to manually place in points on the workpiece, which the robot is intended to move through, and whether it will be movement from weld to another, specifically a weld, or wire seeking calibration sequence, etc. This can take upwards of a week to produce, before it can be implemented on the robot. It is organised in segments, resembling elementary operations, just applied manually in the offline programming software. For all the weld beads, the programmer will have to put in an approach point, the start point of the weld, any intermediate points, end point of the weld, leaving point, and multiple wire seeking points in that area to be executed just before the weld, in order to precisely calibrate the robot to where the bucket is in the real world. And all of these points have to be plotted in with extreme precision as well, which end up being very time consuming[8].

Delimitations

Currently CAD models only deals with geometric data. There is not a lot of information on the processes and how they should perform in order to create the product. For robotic welding third party software is used like Delfoi Arc or DTPS for this purpose. But the integration between CAD model software and welding software does not really exist. Here elementary operations could be a "common language" connecting them and serve as the foundation for extracting data from the CAD model and use it in a welding software tool, with the possible to re-parameterise the weld to fit the CAD model.

The article by Larkin et al. showcases the idea of elementary operations in the form of tags or process points, and that this step is necessary for the automatic flow from CAD model to robot program. However, the tag creation step was conducted manually, which for complex products can be highly time consuming, which is the case for Sjørring. Since Sjørring uses upwards of a week on creating a robot program for a new product it is often not feasible for them to do so, unless the order of the products increases to more than five shovels. Otherwise manual welding is the better solution for them, both in cost but also to be able to deliver on time[8]. Any way of decreasing this time will be an immediate benefit. Since the shovels are mainly variations of one another the ability to reuse weld segments will help decrease the time to create a program significantly.

The ability to parameterise weld beads as shown in [1] could prove useful in solving this issue, since the shovels vary, for instance in length. And the ability to automatically put in a weld bead to fit along a seam, will also move that task away from the programmer. However, this feature is in its early days and have only been tested for parameterising weld beads for straight lines and specifically on the steel box presented in the article. More complex geometries therefore still need to be overcome. And to fully generalise elementary operations as a tool, further development into transferring the tacit knowledge of professional welders into digital form and apply it correctly is required, which possibly necessitate machine learning and artificial intelligence. To function seamlessly, a data bank of all the types of welds there is, has to exist in order to correctly identify the geometries of the workpiece and understand how it should be welded. This is also where parameterisation can be useful in order to, for instance extend a curve to fit the 3D model of the product, where only a smaller version exist in the data bank. But before this can be realised, more work and research has to go into robotic welding of difficult geometries, which currently is limited to straight lines and curves.

5.2 Sequence Generation

The elementary operations themselves should contain information regarding the geometry of the weld bead, the torch angle, torch intensity and weld thickness. However, ordering them correctly to generate the whole welding process is saved for the following step, where the sequencing of elementary operations is determined. Here the decision making, in regard to where the best place to begin welding, end the welding, and everything in between, takes place. Determining which weldment comes when, often stems from tacit knowledge of the workers previous experience with welding the product or variations of it. This could be understanding how the workpiece will behave depending on where the welding process is first applied, or which sequence typically will have the fastest cycle time. To automate this process, this tacit knowledge will need to be digitised and work in tandem with the data bank of robotic weldment motions to find a preferable sequence based on perhaps, cycle time, minimum non welding movement or minimum workpiece distortion.

Literature Informed State of the Art

In Larkin et al. [27] (and as mentioned in section 5.1) tags are used to digitally describe weld paths based on the CAD model of the product worked on. In this case, these tags, which contain the information regarding the weld used for, among others, generating the trajectories for the robot, also contains a name representing not only the weld type and procedure, but also the weld order. Since these tags are created by the end user, the sequencing of the tags are also manually determined.

At Sjørring the sequencing of the weldment segments placed as points in the welding software is done when all these points have been plotted in. The sequence is determined by the robot programmer and handled manually. A lot of this work is accomplished based on experience from previous robotic welding work on product variations of the buckets. This is e.g. starting in the correct spot on the bucket in order to avoid a too high distortion in the product where the bucket will fold in on itself[8].

Delimitations

The idea of generating a plan for the order in which to execute the welding is not new and quite necessary when creating the robotic programming. Automating this process is, however, and currently has not been accomplished yet. Tacit knowledge often plays a part in where to start the welding on the workpiece and where to stop. This has to be explicitly stated and translated somehow to a machine learning decision making algorithm. It will therefore be very difficult to generalise for all production companies and will even be difficult to generalise for different products within one company, unless the different products are variations of the same base product, where some information can be transferred and recycled from product to product (eg. parameterisation). With Sjørring as an example, so much local knowledge are relied on when applying robot welding, also within determining the order in which parts of the bucket are welded. So before this step can be fully automatic, all that information, which can be difficult to come by or to even express has to be made explicit. Alternatively this could be done manually by a robot programmer to utilise their tacit knowledge and experience with the products to simply build the sequence based on the elementary operation segments. This could turn out to be a solid compromise, which would give the development of automation within this area a lower priority.

5.3 Kinematics Model

The kinematics model is the step following a determined sequence of the elementary operations or weld segments. It will handle the motion planning using the kinematics of the robot and information from the CAD model's elementary operations, as well as information regarding the relation between the robot and product in the virtual world/simulation. It is an iterative process. The motion planning will be an iterative process combined with the process model to incorporate the physical constraints of the welding task into the overall motion planning in search of the most optimal path planning, using the information from the elementary operations and their execution sequence.

Literature Informed State of the Art

In Larkin et al.[27], the motion planning task is divided into configuration space (C-space) and task space (T-space). The C-space revolves around planning the motion between a weld segment end point and a weld segment start point. The T-space deals with finding and optimising the tool path while satisfying the welding constraints. The difficulty of T-space path planning for welding is because the high requirement for quality, which will be elaborated in section 5.4. An A* algorithm was initially used, but turned out too expensive computationally. The main problem was instead divided into sub-problems. When solved sequentially, the sub-problem can form a solution for the entire problem. A customised Probabilistic Road Map (PRM) planner was used for solving the planning issue in C-space. In 250 tests a path was found in under a second 83% of the time, but some scenarios with difficult geometry ran upwards of 56 seconds. The complete motion planning was tested in simulation in five welding scenarios with the average planning time per weld being 24.9 seconds. However, there was a high variation in execution time due to difficult geometries with restrictive constraints for the tool poses and motion during certain welds.

In "Benchmarking framework for robotic arc welding motion planning" by De Maeyer et al. 2020 [16], different motion planning algorithms for robotic arc welding are evaluated and compared. Siemens, Verbotics, Delfoi Arc, etc. offer automatic path planning abilities for robotic welding, however it is unclear what the success of these algorithms in finding collision free paths for complex workpieces are. Based on their research and knowledge, there is no single planner capable of solving a full welding task.*"Often, a single motion planner is not sufficient to solve all types of path constraints encountered in a robot arc welding path"* As a result their focus was on a few specific low-level planners best suited for certain tasks and combining them with a high-level planner. The high level planner's job was to determine which planner to delegate what tasks. It contains the overview of the entire welding process a divide out specific planning for each weld segment to low level planners and combine it all to create the full robot motion. The low level planner's job is then to find a series of joint positions, which will follow the weld bead. Another task could also be determining the intermediate motion from one weld to another. The planning algorithm used for in-between motion was RRTConnect and for the weld paths a custom grid search planner (from previous work by the author [46]) was used. The solution was simulated for two different robotic welding cases and showed a success rate of 100% and 80% respectively over 20 runs, using 0.21 s and 0.92 s for planning time. This will be expanded on in the future with more low level planners and high level planners to solve numerous complex welding tasks.

At Sjørring, the robotic welding software they use, whether it being Delfoi Arc or DTPS handles the initial motion planning. Then the programmer will check for collisions and correct for any infeasible movements.

Delimitations

As mentioned in section 2.4 the kinematics model step is the most mature one. Automatically generated path planning for robotics is used in all types of robotic application, where offline programming is utilised. It is also utilised for welding applications, just as described above. But it is not yet used fully to cover everything welding. As mentioned in De Maeyer et al.[16], complex welding tasks requires complex motion planning, which a single planner will have difficulties to cope with while ensuring high quality weld beads. The method of having multiple low-level planners designed for specific tasks and a high-level planner controlling them suggested by De Maeyer et al. might aid this issue. But as mentioned Larkin et al.[27], there will still be geometries where the optimal welding torch angle will not be feasible and will require more testing in the physical world to find a way both simulating these trajectories and achieving quality welds in the real world.

5.4 Process Model

The process model is the fourth step in the flow. The task of the process model is to influence the motion planning for when the welding torch is turned on. Using the information from the elementary operations it will constrain the motion planning problem for the weld segments in order to ensure high quality by having the best suited torch angle, movement speed, stabilised movement and correct torch intensities at given times during the process. As mentioned in section 5.3 the idea is to create a iterative collaboration with the kinematics model to end up with satisfying trajectories for the entire robotic welding of the given workpiece.

Literature Informed State of the Art

In Larkin et al.[27] the T-space handles the path planning for the weld segments themselves. The requirement for high quality when welding is additionally challenging to normal T-space operations. When the welding process begins, many parameters are suddenly constrained as opposed to moving from home position to an approach point. For instance, when welding in a straight line, the torch angle has to remain the same and the speed of the movement has to be consistent. as mentioned in section 5.3 an A* search algorithm is used here and the trajectory planning are divided into sub-problems to handle the welding path generations. Otherwise, the computational load would be to great.

De Maeyer et al.[16] puts the burden on motion planning for the weld segments onto their local planners, that serves to find a set of joint position sequence to be able to follow the weld bead line while ensuring a high quality weld.

Delimitations

An immediate problem for any robotic welding process model is that many geometries are currently infeasible to weld. And some geometries will stay unsolvable for motion planning when it is constrained by the objective of creating a high quality weld. As mentioned in Larkin et al.[27], for certain geometries the end-effector will not be able to reach the area for welding with the preferred torch angle, for example. This would have to be taken into account in the iterations between the kinematics model and the process model, that a compromise will have to be found as to not make the motion planning for the specific weld (or elementary operation) infeasible while still maintaining collision avoidance.

More standard geometries such as the inside of a corner, etc. which should be a possibility to weld with robotics, still require more research into how to facilitate it. In Larkin et al. they emphasise this as a problem, with them operating with the criteria that part intersections are not completely welded in their simulation, since a high quality method of welding such an area has not yet been made possible. The automation of going from specific welding parameters for certain welds in their "tag creation" phase to feed into the motion planning was partly manufactured since *"... control over the position, length and type of weld was controlled by the end user."*[27].

5.5 Real World Compensation

The last step to realise the full flow of automating the production preparation for robotic welding is to implement it in the physical world. Usually, fixtures are used to try to eliminate the errors that might occur due to thermal compensation. And when the fixture is created it eases the calibration to an extent since the fixture typically will be in the same location relative to the robot each time. However, for low volume production, the expense of creating a custom fixture is often to high to be made up for with the output of the welding process. As a result, the issues of knowing where the robot is in relation to the workspace environment, including the workpiece(s), and the modification of the workpiece during the weld caused by the heat need to be solved for scenarios without fixtures before any of the previous steps can be realised or at least simultaneously. A synchronisation in development between the two parts, automatic production preparations and the execution of it.

Literature Informed State of the Art

In the report "Zero-Point-Calibration 2.0" published in 2020 by Mikkelsen et al.[47], a sensor system with a camera and fiducial markers is proposed to calibrate a UR5 robot automatically. It utilises an ArUco marker (black and white squared paper) to estimate the pose from the camera to the marker. The camera is attached on top of the end-effector and by measuring where it is located and combine it with the pose estimation the robot will be able to know the location of the marker in relation to itself. The marker is placed in the middle of a metal plate with a metal rod pointing upwards in all four corners. The robot arm starts moving around to detect the marker. When this has happened the end-effector moves closer to the marker and the pose estimation is calculated with the support of a median filter. The precision is then tested by moving between the four metal rods and calculating the error, which achieved a precision of 0.35mm, and the calibration it self lasted less than one minute.

In the article "Precise seam tracking in robotic welding by an improved image processing approach"[48] by Banafian et al. 2021, a sensor system involving a camera is used to try to follow a weld seam in real-time. This is tested as an application for a GMAW process, where two workpieces are to be welded together along a straight line and slightly curved line (the seams) in the plane. It exploits the fact that in the area where the welding occurs, a weld pool appears. The light from this and the welding torch sends a gleam along the seam. This facilitates edge detection of the seam. The frames are separated into two windows, one for the weld pool and one for the line the torch should follow. After applying multiple filtering methods to remove noise, including Convolution filter, Median filter, and Laplacian filter, a new and improved Canny edge detection with dynamic thresholding, proposed in this article, is applied to detect the seam and then follow it. The deviation error stayed within $\pm$ 0.3mm, with the main challenges to address being heavy computational load from the image processing and the light interference for the welding torch.

At Sjørring, calibration of the robot to the workpiece is handled by the use of wire seeking, touch calibration. As mentioned in section 5.2, manually build into the robot program are points designed for applying wire seeking. When implemented on the robot, the end-effector will start a small distance away from the first contact point. The end-effector will then move slowly towards this point and keep going until contact has been made with the workpiece, even if this contact point is beyond the point placed in the offline welding programming software. This reference will then be updated to fit where the workpiece is located in the real workspace. One complete wire seeking action is typically performed around corners of the workpiece with three different touch points to calibrate that corner of the workpiece correctly. These wire seeking actions are done sporadically throughout the welding process to decrease the error and possibility of error between the location of the workpiece in the virtual world and the real world due to imprecise placement or expansion/discrepancy caused by heat during welding. The complete welding process takes around 6-7 hours where around 20 minutes of it are used for wire seeking calibration. This method is not the only one used for managing the thermal expansion of the workpiece. A supporting strut is manually welded onto the middle of the bucket to avoid it bending in over it self (visualised in Figure 5.5). Afterwards this supporting strut has to be manually removed before it is a completed product. But, even with these methods applied there can be a mismatch upwards of 50 millimeters from the final product to the simulation of it[8].

Figure 5.5: A bucket from Sjørring Maskinfabrik with supporting struts[8]

Delimitations

The way Sjørring approaches adapting the robot programming to the real world's discrepancies has some downsides. Their calibration using wire seeking uses upwards of 20 minutes throughout a 6-7 hour weld session. This is not a huge drawback since it is only a small part of the overall cycle time and it ensures high precision and therefore high quality. The main issue is the time it takes to program these calibration points, that has to be placed all over their shovels. Removing this workload from the programmer not only frees up a lot of their time, but also decrease the throughput time of the programming of the robot[8].

Having a system in place to compensate for thermal expansion automatically will also decrease the time the workers at Sjørring spend on preparing the buckets for the welding process, since the supporting strut they weld into the middle of the shovel has the potential to be disregarded[8].

The calibration system proposed in [47] appears to have potential due to the short execution time of it. However, for Sjørring's situation their wire seeking calibration also aids them with their issue of heat expansion, which means the dynamic camera calibration would have to be done multiple times with multiple markers strategically placed on the workpiece. That then creates the issue of finding a way to incorporate these markers into the workpiece, without them being pieces of papers taped onto the metal. This incorporation work will either have to be done in-house or by the suppliers of the materials. In cases with simpler/smaller products, this calibration system might be more feasible, due to less heat expansion or a less ingrained method of compensating for the discrepancy.

The seam tracking method proposed in [48], to achieve heat compensation was created to be performed on two metal pieces needing to be welded together. This method exploited that the gleam from the weld torch can be used to identify the gap between the two workpieces. The scenario also only took place in plane (2D), so some more development will have to take place for it to be a general solution for robotic welding purposes (other types of welds and welds in three dimensions). This solution also has the same delimitation as the calibration method in [47], that a sophisticated camera system has to be put in place together with the robotic system, upping the initial cost and the difficulties of getting it up and running.

It is clear that a sophisticated sensor system is necessary in the robotic welding industry if fixtures are to be omitted from the factories. But in the case that fixtures are preferred, when dealing with many parameterised variants of a product, parameterised fixtures could be a solution. At Sjørring Maskinfabrik, their solution is a combination of using sensors and a fixture in the form of a supporting strut welded onto the bucket. Albeit it being a solution and works as intended, welding a supporting strut to the workpiece is a tedious procedure where manual welding has to be used as well, before the robotic welding can be applied. And on top of this, the rod has to be removed afterwards without compromising the quality of the bucket. Having a parameterised fixture that can be fitted to all variants of a base product would decrease the cost for fixtures in the production. This idea could be an alternative to developing a sophisticated sensor system for combating thermal expansion and calibration, but is yet to be developed and implemented itself.

Robotic Welding Execution

The highly specific way of dealing with calibration and thermal expansion at Sjørring also highlights the difficulties of standardising a solution for robotic welding and the preparation of it, when the processes and methods are so very dependent on what the product needing welding is. The task of automating the pre-production of robotic welding is interdependent on both what comes before and after. The whole flow can be grouped into three categories;

1. Geometric definition
2. Process definition
3. Execution of process(es)

How this relates to the AOLP model is illustrated in Figure 5.6.

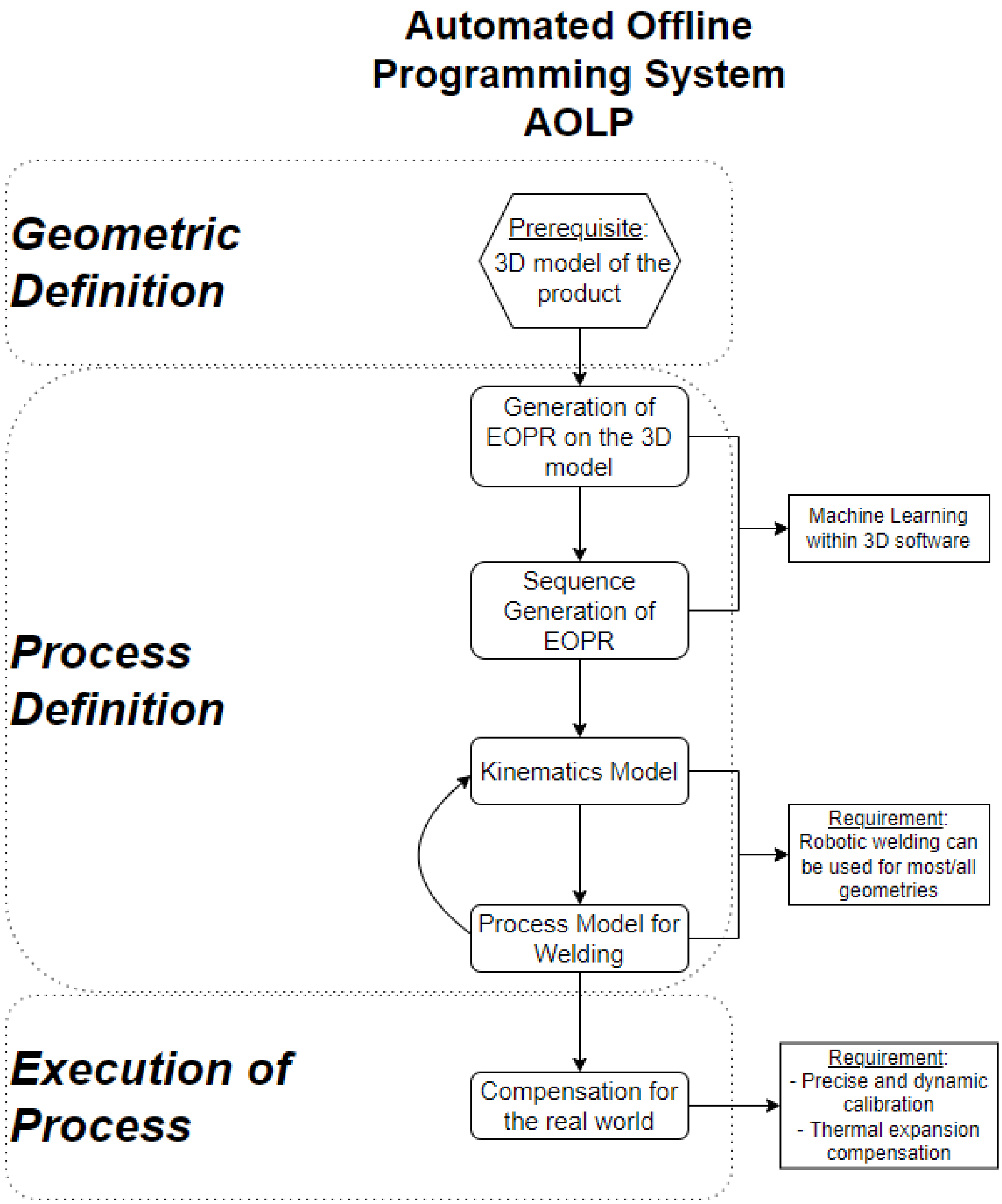


Figure 5.6: Automated Offline Programming System pipeline for complex, parameterised, low volume products.

As laid out in this chapter development along the whole chain of the model is required to see the first signs of automating the pre-production steps of robotic welding. The progress on both the process definition and execution part has to undergone synchronously, since these are interdependent for a full solution. The design of a system that facilitates the implementation of process definition structure has to be taken into account and vice versa, when developing the individual solutions. As mentioned in chapter 4 this project is conducted in collaboration with Sarivan et al.[1], which has been looking into taking the first steps of how to apply simple elementary operations on to a CAD model and parameterise them to fit the model's needs. The following of this report will look into the other part of automatic robot welding pre-production, namely how to execute the welding process with the use of these elementary operations. One of the issues on this part is the infeasibility of creating fixtures for low volume products described in section 2.2. Without fixtures in play calibration suddenly becomes larger concern. A system to take on that challenge will be investigated in the following chapters drawing inspiration from what was uncovered in the beginning of this section.

Chapter 6

Sensor System

In order for automation of the production preparations for robotic welding to be realised, it has to have the possibility of being implemented correctly. For an ETO production company with many low volume variants of a few base products, fixtures as of right now are impractical to create for every single variant, when the volume of them are so low. Therefore, this project will look into how to implement a sensor system, that serves as an alternative to the job fixtures are used for, namely information of the precise location of the product.

Described in section 5.5, at Sjørring Maskinfabrik, touch sensing is applied using the end-effector tool. This is combined with a small "fixture" taking the form of a supporting strut(s) manually welded into the middle of bucket. But this is a highly specific solution designed for their production only. Both in Mikkelsen et al.[47] and Banafian et al.[48] a sensor system with a camera is used, one for calibration of a robot to a fiducial marker and one for real-time seam tracking for arc welding. As a result, this project will be using a camera in combination with ArUco markers to dynamically calibrate a robot to a workpiece.

6.1 Concept

To dynamically calibrate the UR5 robot, described in chapter 3, a camera and markers will be incorporated in the workspace of the robot. When programming a robot offline, the environment in a virtual world is created to match the environment with and around the robot to match precisely. And to apply the robot program to the robot for, for instance robot welding, it is incredibly important for the quality of the weld, that the workpiece is located in the exact same location in relation to the robot in the real world as it is in simulation. This requires that a new workpiece is carefully placed correctly each time, which can be both time consuming and error prone. And after being placed, there is still the small risk of the workpiece being moved even by a tiny amount by unnoticed external factors, which will affect the weld bead precision and accuracy, when the welding process begins.

A recognisable feature located on the workpiece, which a camera will be able to detect, can decrease the impact of this issue. And this camera will only have to be set up once, compared to having to set up a workpiece correctly every time. The idea is to have the camera determine the location and orientation (pose) of the recognisable feature, a marker, and thereby also the workpiece, and find the relation between the camera and the robot. When combined the robot will be able to find the marker using the camera and estimate the pose of it, meaning the robot is calibrated to the workpiece.

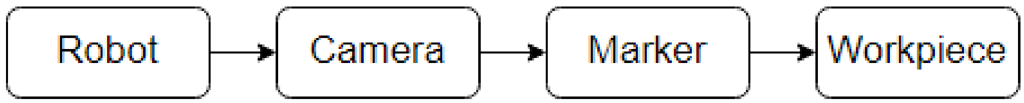


Figure 6.1: A flowchart presenting the basic concept of the relations from robot to workpiece

System Setup

The camera to be used in this project is a Phillips SPC620NC webcam from 2007. The resolution of the camera is 1.3 Megapixels, 1280x960 with a 30 frames per second. The focus ring around the lens can be adjusted and has been set to focus on the workpiece from where it is located[49][50]. A plastic support on the bottom of the camera has been removed in order to place the camera as desired.

The camera is glued to a custom created camera support/holder, which is placed in the workspace. The custom camera support has been made in order to determine exactly where the camera is located and its orientation in relation to the robot. This will be elaborated further in chapter 7. ArUco markers will be used in combination with the camera to obtain pose estimation of the workpiece in relation to the robot. ArUco markers will be elaborated in section 6.3. The workpiece where the marker will be placed upon is a metal plate, with the dimensions 200x60x10 millimetres for length, width and thickness respectively. In order to use the camera for precise pose estimation, the camera requires its own calibration to obtain its camera matrix and distortion coefficients, which the following section 6.2 will dive into.

Figure 6.2: The Phillips SPC620NC camera

Figure 6.3: The custom made camera holder

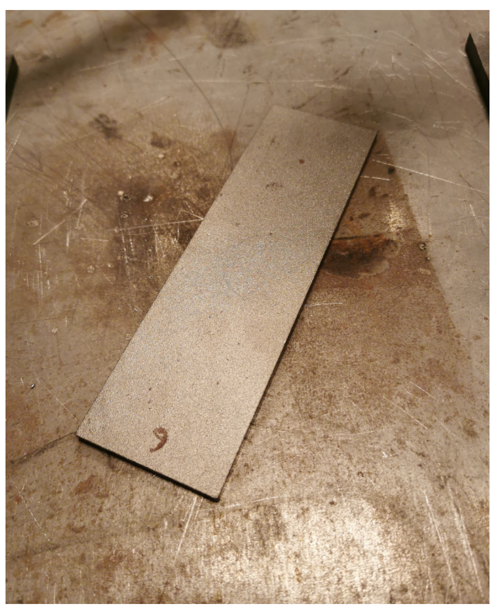


Figure 6.4: The workpiece

6.2 Camera Calibration

For gaining any useful information by pose estimation from the markers, the Phillips camera's intrinsic camera matrix and distortion coefficients need to be obtained. A camera's intrinsic matrix is a 3x3 matrix containing some relevant properties of the camera.

$$K = \begin{bmatrix} f_x & 0 & x_0 \\ a & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The focal length of the camera is described by the f_x and f_y parameters. It is the distance from camera's sensor to the lens' optical centre measured in millimetres. The optical centre of the lens is where all the light converges and creates the sharpest picture (see Figure 6.5. The focal length is what determines the angle of view. If the focal length is short, the angle of view is wider and an object in the frame will appear smaller, than for a long focal length, where the angle of view would be narrower. The parameters x_0 and y_0 contain the values for the location of the principle point offset. This is the point where there is an intersection between the optical axis and the camera sensor's plane[51][52][53][54].

To describe the distortion in the image, distortion coefficients are used. They are represented in a five dimensional vector. Distortion can be seen in a picture when the edges of the image looks curvy, when meant to be straight.

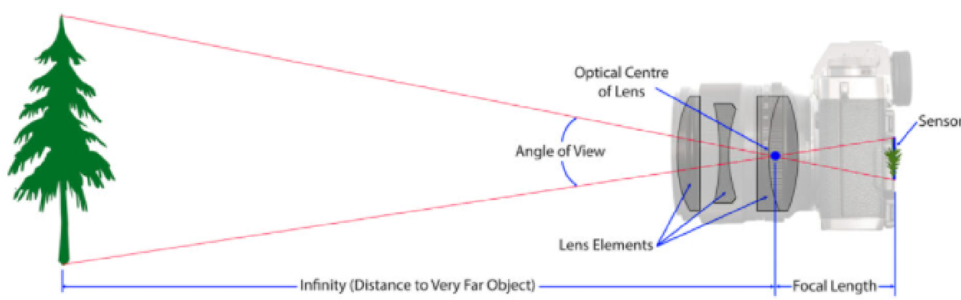


Figure 6.5: An illustration showcasing the concept of how a camera captures an image and relevant terminology[52].

Every camera has a different set of distortion coefficients and an intrinsic camera matrix. So is the case for the Phillips web cam used in this project. For finding this camera's camera matrix and distortion parameters, Zhang's method for calibration of cameras will be used [55]. A known object is used for matching what the camera sees in 2D to the reality of 3D space. In this case a chessboard is used. Acquiring at least three pictures of the chessboard in different positions and orientations, with the camera being stationary in the same place for all pictures will then provide the camera's intrinsic camera matrix and distortion coefficients, as well as a rotational vector and translational vector for each picture. The input, besides the pictures are the chessboard size in squares and side length of each square. This is implemented in chapter 8.

6.3 Fiducial Markers

With the camera calibrated and the camera matrix and distortion coefficients obtained, the next step is to focus on the kind of recognisable feature to be used for estimating the pose of a workpiece attached in 3D based on the 2D image from the camera. Fiducial markers have been chosen, specifically ArUco markers, based on previous experience and the results from Mikkelsen et al.[47] (see section 5.5).

A fiducial marker is in its core a reference point and can be used for many purposes. In the context of images and vision sensors it is used as an object in the image to be used as a reference for measurements, to put the image into context. For augmented reality in computer vision fiducial markers are used to create integration between the real world and the world seen through image sensors, for instance for correctly determining orientation or scale. One example of such a fiducial marker is an ArUco marker[56].

ArUco

ArUco markers are square markers with a set amount of square bits inside it that can either be white or black. Around this is a black border with the same width as the bits. The binary coloured inner bits creates a unique identification for each marker and stores information regarding the orders of each corner. The corners are valued from one to four going around clockwise from the top left (the first corner being the top left). The markers can be used for pose estimation. The corner information are used to determine the orientation and given the side length of the marker as an input, the translation can be determined according to the amount of pixels the marker takes up in the image. The ArUco marker was first proposed in a scientific paper from 2014[57]. Two of the authors, Muñoz and Garrido have since created a library for them to be used for programming, the aruco module, which have been incorporated into OpenCV. There are different variations of series of the marker called dictionaries, where the number of markers included varies, as well as the size for the marker with the number of bits in them[57][58][59][60]. For this project, markers with 4x4 inner bit size will be used. They have been generated using an online marker generator, which has been created by Oleg Kalachev[61].

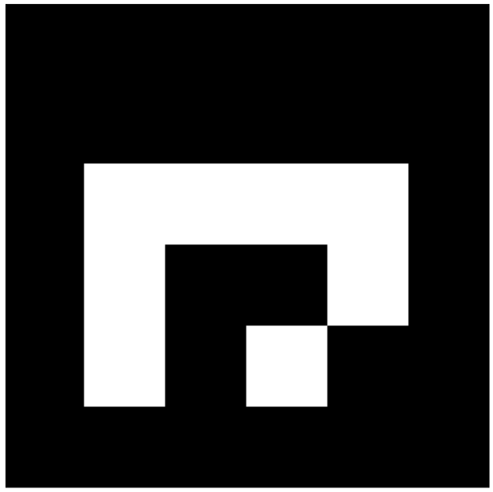


Figure 6.6: An example of an ArUco marker. Here is the 4x4, ID 1 marker[61]

Pose Estimation

Before pose estimation can be used with ArUco markers, the marker has to be detected. The detection itself is a two step process. The first step is to find candidates that could be markers. With adaptive thresholding, contours are found and non-convex objects or non-square shapes are disregarded. The second step analyses the inner bits to determine if it is an ArUco marker, and if it belongs to the correct dictionary used. If both are true the identification of the marker is determined. Figure 6.7 shows the marker detected outlined with a green border, a red dot marking the center point, and the identification number shown in the top left, which is also the first corner of the marker.

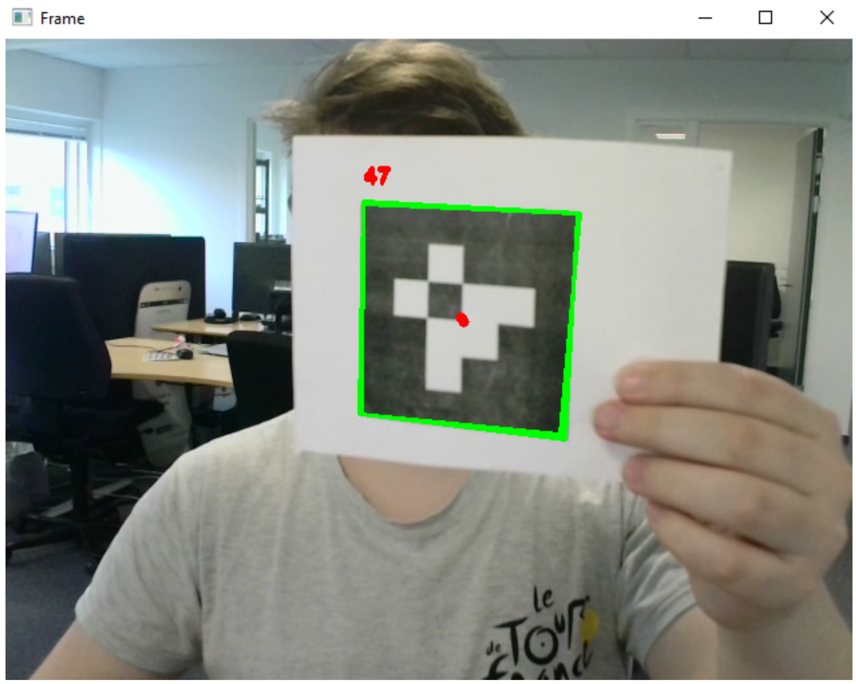


Figure 6.7: Detection of ArUco marker ID 47, 4x4

When the a marker is detected the pose estimation can take place. Described earlier in this section, the camera matrix and the distortion coefficients for the camera used are required here. These two are given as inputs along with the side length of the marker. The output from the pose estimation is a rotational vector (rvec) and a translational vector (tvec). These two vectors describe the relation between camera and marker, both the orientation and the position, based on the 2D image captured by the camera. Figure 6.8 shows an image of a marker with an green border around the marker, and a coordinate frame overlaid. The centre of the frame is located at the centre of the marker. The red axis is the x-axis, the green axis is the y-axis, and the blue axis is the z-axis pointing straight up from the marker.

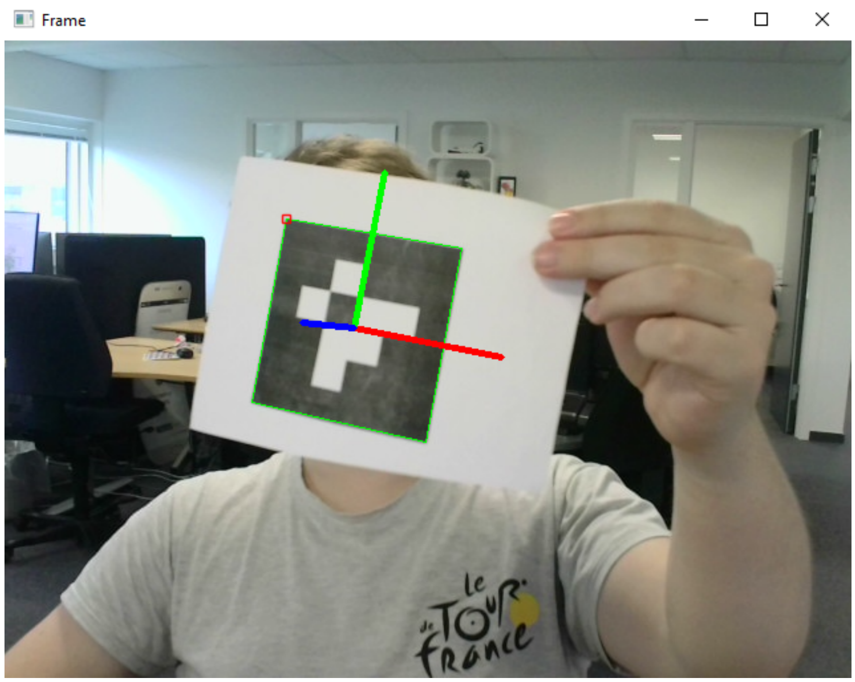


Figure 6.8: Pose estimation detection of ArUco marker ID 47, 4x4

Chapter 7

Workspace Design

In chapter 6 the steps for finding the relation between the camera and the marker was uncovered. So, in order to complete the flow from robot to marker, the relation between the robot and the camera has to be found. The camera has been attached to a custom created camera holder (describedsection 6.1), which will be stationary in the workspace. This camera holder consists of a base block and two rods connected via a rotational joint. This has been introduced to increase the flexibility and potential for the camera view. On the end of the second rod is a small plate, where the camera is attached with double adhesive tape. The dimensions of the camera holder is illustrated in Figure 7.1.

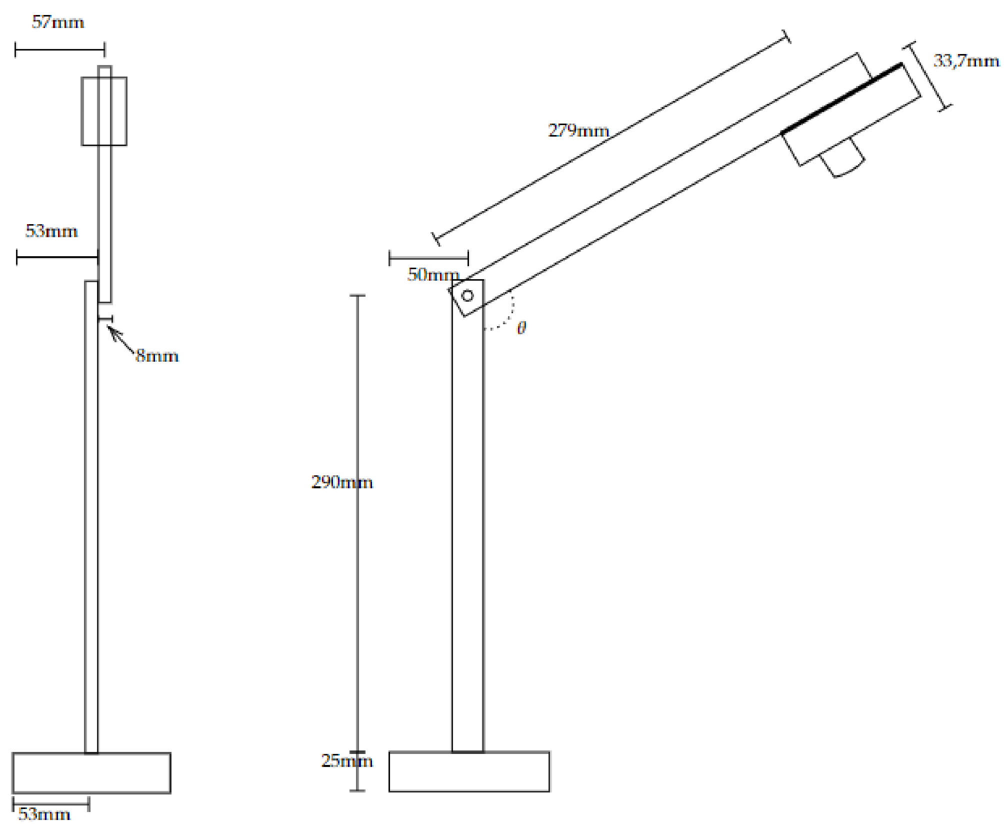


Figure 7.1

To find the pose of the camera, the relation between the camera holder and the robot has to be found first. To corners of the base block are used for determining the translation and orientation of the holder in the workspace. This is achieved by moving the robot's end-effector towards each of the two corners and noting the values for those points in 3D space. The first corner (the closest to the robot) will be used as a reference frame for the entire holder and will be denoted CHF (camera holder frame). The CHF, the side length of the base block, and the second corner (CHF2) will be used for determining the orientation offset in regard to the robot base frame's (RBF) orientation.

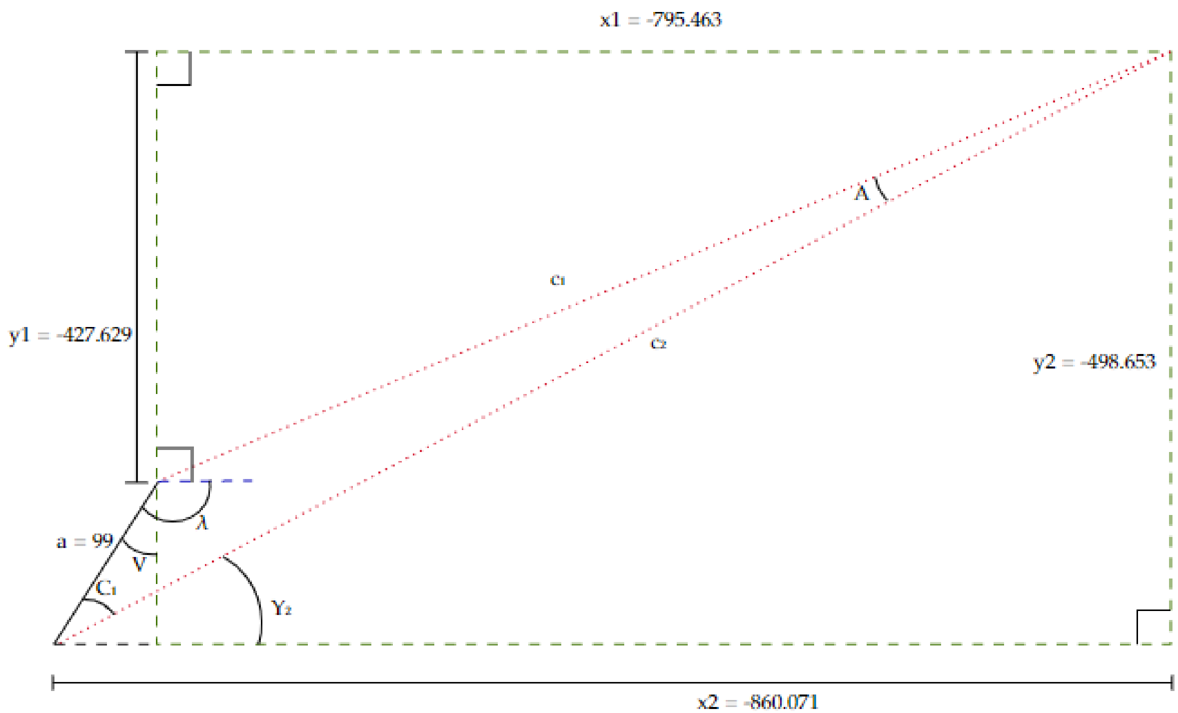


Figure 7.2: A drawing of the view of the robot and camera holder base block seen from above. The top right corner is the location of the robot base frame (RBF). The full black line 'a' is the side of the base block, with the top point of the line being CHF and the bottom being CHF2. λ is the angle representing the rotational offset between RBF and CHF (specifically how much the x-axis is rotated around the z-axis from one to the other). x1 and y1 describes the translational offset between the two in the x-y plane.

The placement of the two corners are:

$$CHF : x1 = -795.463mm, y1 = -427.629mm, z1 = 91.169mm \tag{7.1}$$

$$CHF2 : x2 = -860.071mm, y2 = -498.653mm, z2 = 92.044mm \tag{7.2}$$

The orientation offset of the holder needs to be determined for the camera holder frame. This frame (CHF) is placed on the closest corner to the robot, with its x-axis pointing along the side of the base block towards CHF2. The y-axis points in the opposite direction of the other side of the base block and the z-axis points upwards. The robot base frame has a coordinate frame, where the x and y-axis are pointing away from the table.

It is assumed that the x-y plane of the robot base coordinate frame and the table, which the camera holder is placed upon are parallel, meaning that both of the z-axis of the corner coordinate frames and robot base frame are parallel. This was also verified with an electronic spirit level.

With the use of cosine relations and trigonometry the offset in orientation around the z-axis (noted hereafter as λ) can be found (see Figure 7.2 for reference).

$$c1 = \sqrt{x1^2 + y1^2} = 903.1mm \tag{7.3}$$

$$c2 = \sqrt{x2^2 + y2^2} = 994.2mm \tag{7.4}$$

$$C1 = arccos(\frac{a^2 + c2^2 - c1^2}{2 \cdot a \cdot c2}) = 22.0° \tag{7.5}$$

$$Y2 = arccos(\frac{x2^2 + c2^2 - y2^2}{2 \cdot |x2| \cdot c2}) = 30.1° \tag{7.6}$$

$$V = 90 - (C1 + Y2) = 37.9° \tag{7.7}$$

$$\lambda = 90 + V = 127.9° \tag{7.8}$$

The offset in orientation around the z-axis λ has been found to be 127.9°. With this value and the CHF placed, the parameters can be determined for describing the relation between RBF and CHF. This transformation is also visualised in Figure 7.3 on the left, showing the placement of the frames seen from above.

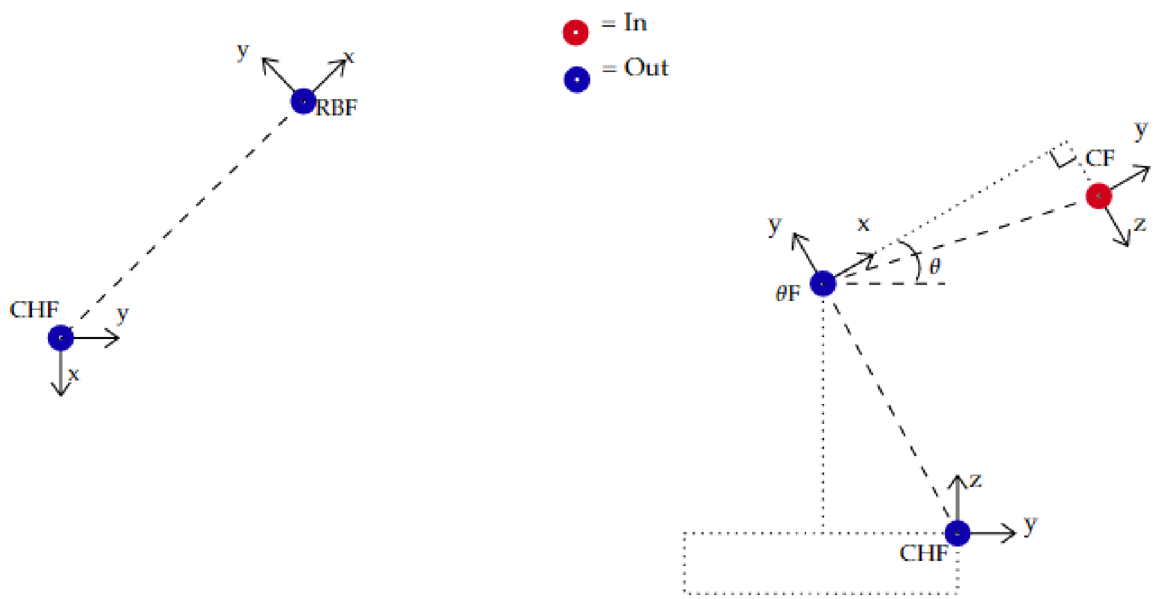


Figure 7.3: The placement of the custom placed coordinate frames in relation to each other. To the left is the robot base frame and camera holder frame seen from above. To the right is the camera holder frame, theta frame and camera frame on the camera holder seen from the side.

Two additional frames besides the CHF has been placed on the camera holder. The theta frame (θF) and the camera frame (CF). The goal is to find the relation between the robot (RBF) and the camera (CF). The two intermediate frames, CHF and θF have been placed to aid this process. The CHF for convenience, due to a parallel x-y plane with RBF and θF, since the camera holder at this point can make a rotational movement, creating a variable (θ), which changes the relation between RBF and CF. The CF has been placed due to how the camera perceives the world. The positive x-direction is to the right of the camera, seen from the cameras perspective, the positive y-direction downwards, and positive z-direction outwards from the lens. And since the camera is placed upside down on the end plate of the camera holder, the frame is placed with respect to this. The frames can be visualised placed in RoboDK in Figure 7.4 and Figure 7.5.

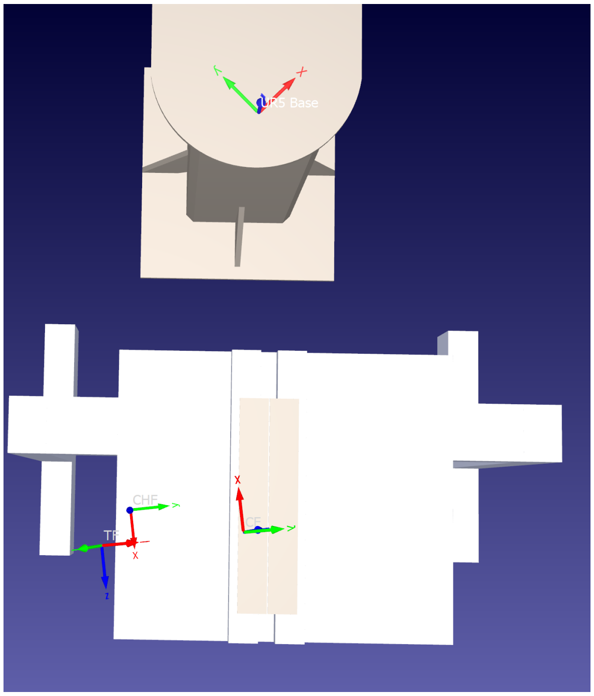


Figure 7.4: The frames of the camera holder and robot base seen from above in RoboDK

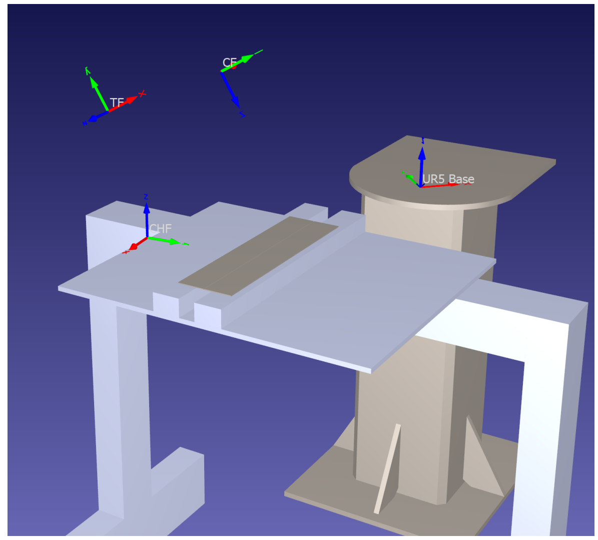


Figure 7.5: The frames of the camera holder and robot base seen from the front in RoboDK

With the frames being placed and the relations between them known as of Figure 7.1 (describing the dimensions of the camera holder), Figure 7.2 and λ the transformation matrices can be set up.

The 4x4 homogeneous transformation matrix from the robot base frame to the camera holder frame:

$$T_{CHF}^{RBF} = \begin{bmatrix} cos(\lambda) & sin(\lambda) & 0 & -795.463 \\ -sin(\lambda) & cos(\lambda) & 0 & -427.629 \\ 0 & 0 & 1 & 91.169 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.9}$$

The 4x4 homogeneous transformation matrix from the camera holder frame to the theta frame:

$$T_{\theta F}^{CHF} = \begin{bmatrix} 0 & 0 & 1 & 57 \\ cos(\theta) & -sin(\theta) & 0 & -52 \\ sin(\theta) & cos(\theta) & 0 & 290 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.10}$$

The 4x4 homogeneous transformation matrix from the theta frame to the camera frame:

$$T_{CF}^{\theta F} = \begin{bmatrix} 0 & 1 & 0 & 279 \\ 0 & 0 & -1 & -33.7 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.11}$$

θ has been set to be 30° for a preferable overview of the workspace. And with λ at 127.9°, the full 4x4 homogeneous transformation matrix from robot (RBF) to camera (CF) can be determined:

$$T_{CF}^{RBF} = T_{CHF}^{RBF} \cdot T_{\theta F}^{CHF} \cdot T_{CF}^{\theta F} = \begin{bmatrix} 0.615 & 0.683 & 0.394 & -667.629 \\ 0.789 & -0.532 & -0.307 & -599.495 \\ 0 & 0.500 & -0.866 & 491.484 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.12}$$

The transformation matrix presented above describes the mathematical relation going from the robot to camera. Combining this information with the output from the pose estimation, which describe the relation between the camera and the marker, the entire connection from the robot to the marker can be determined. This will be explored in following part, Part IV.

Part IV

Implementation

Chapter 8

Calibration of Camera

The calibration of the Phillips SPC620NC webcam is conducted using the chessboard illustrated in Figure 8.1. For the calibration, the size of the chessboard in squares has to be declared along with the side length of one of the squares. The chessboard used is 7x9 in squares and the side length of the squares is 20 millimetres. Five images has been chosen for the method, all taken from the exact same pose of the camera. In the five images the chessboard has been moved around, rotated and tilted to provide as much information as possible.

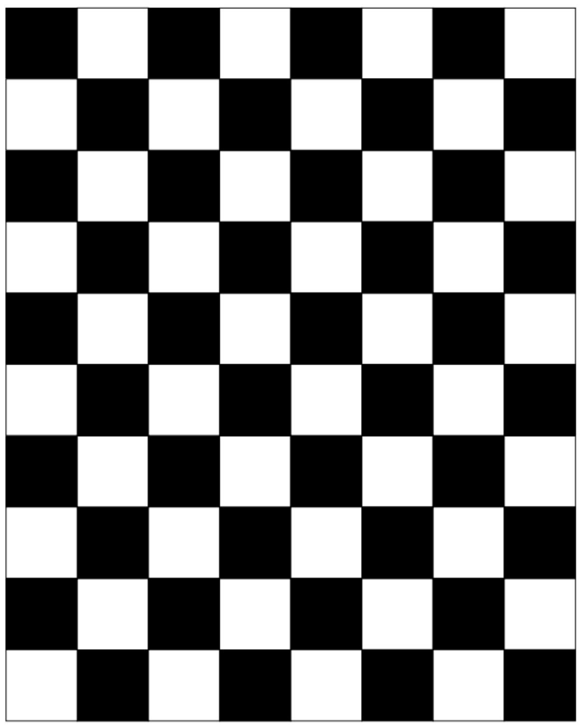


Figure 8.1: Picture of the 7x9 chessboard used for calibration of the camera

Figure 8.2: Example of one of the images used for calibration of the camera

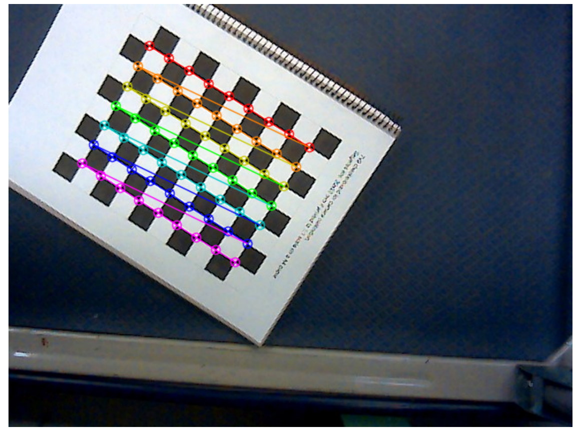


Figure 8.3: ... And here the same image with the detected points in the chessboard illustrated.

The images are deployed in a Python script created for calibrating the camera using OpenCV. The output of the calibration is the intrinsic camera matrix k and distortion coefficients d of the camera. The camera matrix has been determined to be:

$$K = \begin{bmatrix} 906.84883687 & 0 & 282.93565827 \\ 0 & 903.99325314 & 204.01842142 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.1}$$

And the distortion coefficients has been determined to be:

$$d = \begin{bmatrix} -0.000301167215 & -2.1266099 & 0.00146323008 & -0.00737973372 & 12.1961459 \end{bmatrix} \tag{8.2}$$

These parameters will be used later on to estimate the pose of an ArUco marker given an image of the marker.

Chapter 9

Pose Estimation

With the completed camera calibration of the Phillips SPC620NC webcam, the camera matrix and distortion coefficients have been found (in chapter 8). These parameters will play a big part in correctly estimating the pose of the ArUco marker. An additional parameter is necessary as well to be given as input for the pose estimation. The side length of the ArUco marker used here is 91.5 millimetres. The setup simply consist of having the marker placed in the robot's workspace and within view of the camera. This can be seen in Figure 9.1.

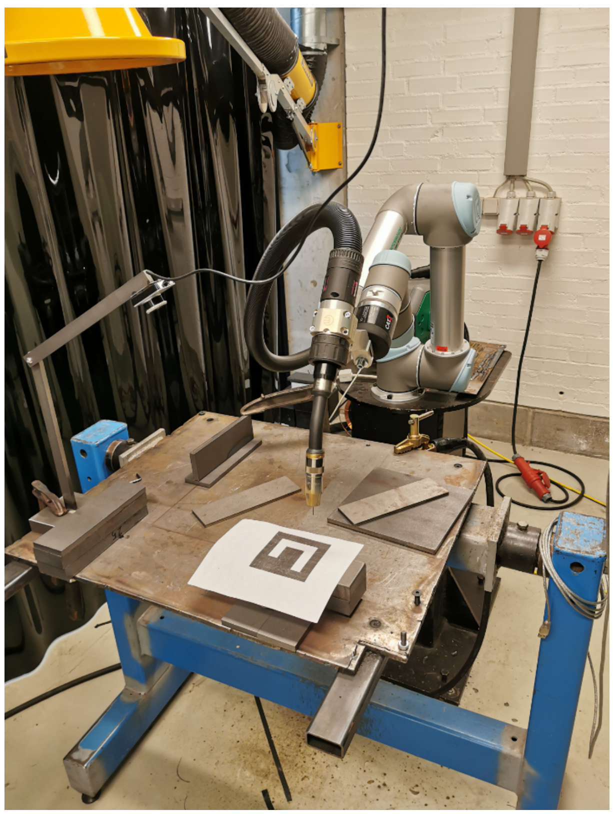


Figure 9.1: The testing setup in the laboratory

The first step in the execution process is to find the marker. This is done using the "aruco.detectMarkers" function. The function detects any marker in the image, finds the ID and the corners of it/them and stores information about where in the image they are.

```
# Detects aruco markers and obtains the ID of the marker and the pixels of the corners
corners, ids, rejectedImgPoints = aruco.detectMarkers(frame, arucoDict, parameters=arucoParams)
```

Figure 9.2: A small part of the script responsible for detecting ArUco markers

When the marker is detected, the next step can take place, the pose estimation. Using the information obtained in the detection of the marker, as well as its side length, the intrinsic camera matrix, and distortion coefficients, the pose of the marker can be estimated. This is achieved using the function "aruco.estimatePoseSingleMarkers". The other two functions in Figure 9.3 provides a visualisation overlay for the purpose of debugging.

```
for i in range(0, len(ids)):
    cv2.aruco.drawDetectedMarkers(frame, corners)  # Draw A square around the markers
    print("\nMarker IDs:\n", ids)  # Prints the marker ID(s) found

    # Calculates the pose estimation for the marker(s) (rotation and location --> rvec, tvec)
    rvecs, tvecs, markerPoints = cv2.aruco.estimatePoseSingleMarkers(corners[i], markerWidthSmall, cameraMatrix,
                                                                     distCoeffs)
    # Draws the coordinate frame of the pose of the ArUco marker(s) relative to the camera on the marker(s)
    cv2.aruco.drawAxis(frame, cameraMatrix, distCoeffs, rvecs, tvecs, 0.075)
```

Figure 9.3: A small part of the script responsible for estimation of the ArUco marker's pose.

The output when using this pose estimation function is a rotational vector (rvec) and a translational vector (tvec) to describe the relation between camera and marker. From the example showed in Figure 9.1, the rvec and tvec gathered are:

$$rvec = [-1.882, -1.991, -0.492] \tag{9.1}$$

$$tvec = [-33.858, 26.237, 467.014] \tag{9.2}$$

These values needs to be imported into RoboDK to be used for moving the robot to wherever it should go to based on the marker. For this case the chosen pose for the end-effector is 10 centimetres directly above the marker pointing downwards to the marker. The issue is the format of the rotation values presented by the pose estimation algorithm (the rvec) and the rotational input the script in RoboDK requires. The format given by the pose estimation is a rotational vector. The format RoboDK operates in is Euler Angles 'XYZ'. And before this information is useful for the robot the 4x4 transformation matrix from the robot to the camera (found in the end of chapter 8 needs to be added on chain. In order to combine the RBF to CF and the CF to the marker (MF, marker frame) the information from both requires to be on the format of a 4x4 transformation matrix. The conversion from the start format to end format can be seen in Figure 9.4

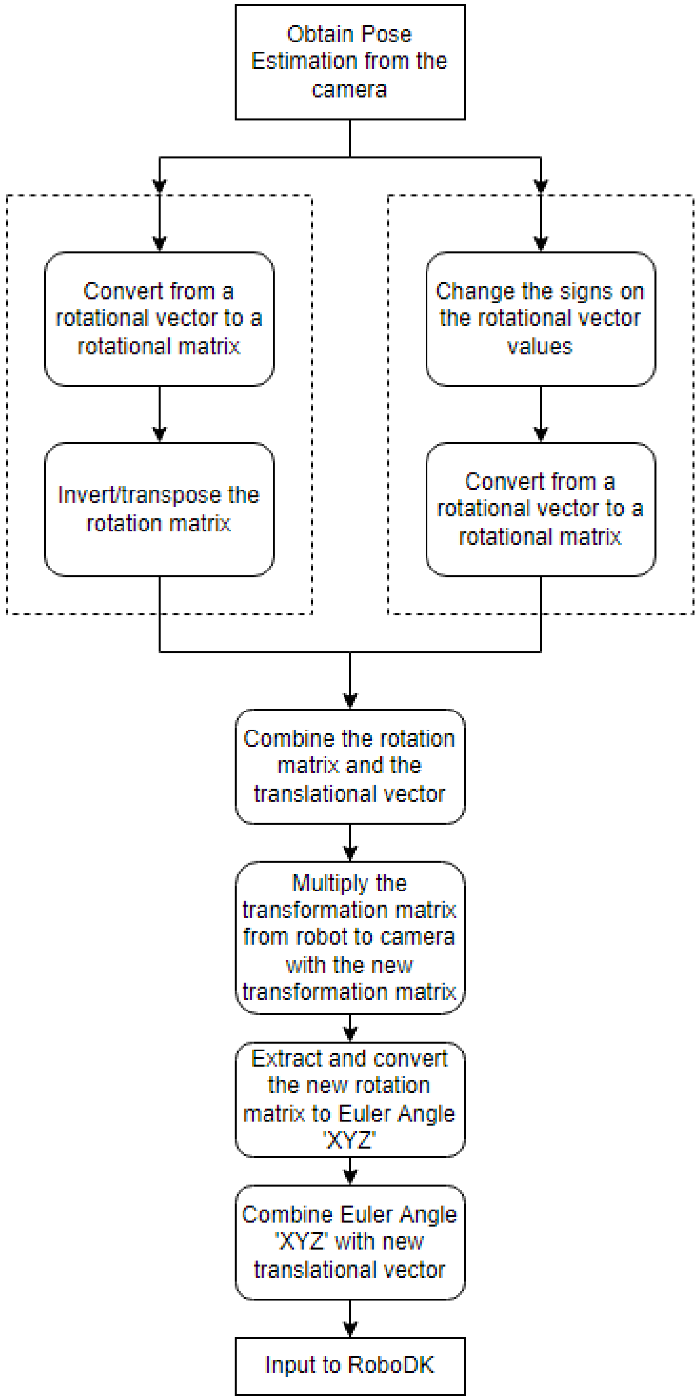


Figure 9.4: A flowchart presenting the steps to go from the output from the pose estimation to the input for RoboDK

Conversion from a rotational vector to a rotational matrix utilises the Rodrigues' rotation formula[62][63]. The translational vector is the same it just requires a change in sign for the three values, since the information desired is the relation from camera to marker not marker to camera. Using the information provided by the rvec and tvec the input for RoboDK can be found by following the steps in the flowchart. The estimated pose of the marker here is:

$$Euler'XYZ'rotation = [0.137°, -1.618°, -40.853°] \tag{9.3}$$

$$tvec = [-486.321mm, -783.688mm, 100.157mm] \tag{9.4}$$

To add the approach point placed 100 millimetres above the center of the marker requires two things. Since the z-direction of the end-effector needs to point directly down into the marker and the z-direction of the marker frame points directly upwards, this orientation needs to be flipped 180°. And 100 millimetres needs to be added to the z-value for the translational vector. The approach point is thus:

$$Euler'XYZ'rotation = [-179.862°, -1.618°, -129.146°] \tag{9.5}$$

$$tvec = [-486.321mm, -783.688mm, 200.157mm] \tag{9.6}$$

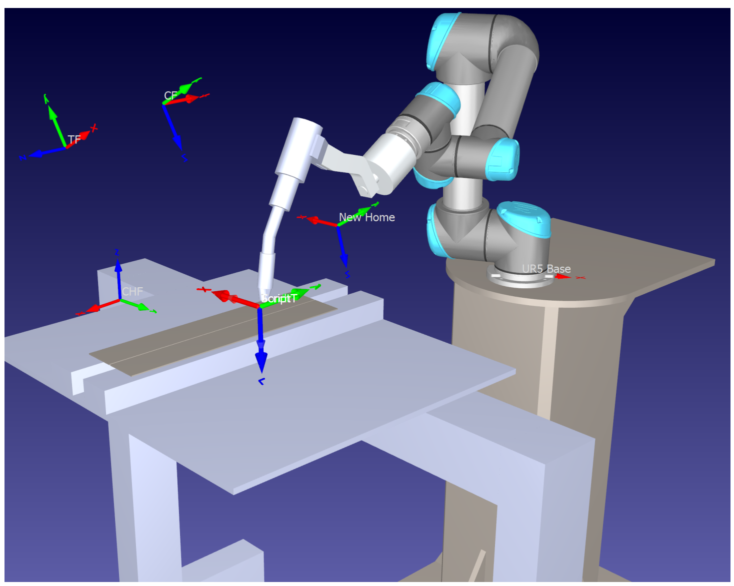


Figure 9.5: Result of the pose estimation test visualised in RoboDK

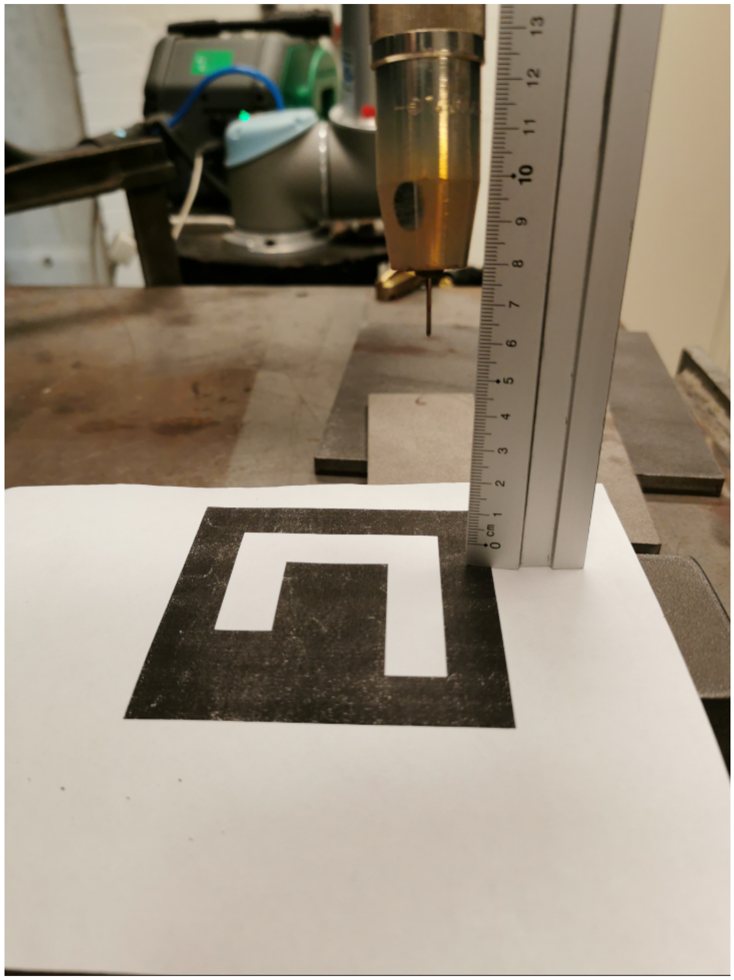


Figure 9.6: Result of the pose estimation test visualised in the real world

The pose estimation result is visualised in RoboDK in Figure 9.5 and in the real world in Figure 9.6. As can be seen, the result is imperfect. The end-effector should be located 10 centimetres above the marker and directly above the centre point. Neither of them are satisfied. There is an offset in all three dimensions with the maximum in the z-direction within five centimetres. With welding applications requiring the error to stay within the millimetre, this results will require improvements if to be used in the welding process. It is however a promising sign with how relative close to the pose estimation using the marker is in location but mainly in orientation as seen in Figure 9.6.

To improve the precision, further tests need to be conducted and the setup troubleshooted to discover where the issues are causing the precision error. This will be discussed further in chapter 10.

Part V

Conclusion

Chapter 10

Evaluation

10.1 AOLP Model

The AOLP model presented in this project puts forth a framework for how to approach pre-production of robotic welding. Robotic welding is a complicated process in and of itself, which also entails the preparation of it to be very complex. However, as discovered in section 2.2 there is a shortage of professional welders, which is a problem only growing larger each year. So even when taking the complexity into account, there is a need for this process to alleviate the pressure that the welder shortage bring. As described in chapter 5 the interest from ETOs and low volume production companies is especially great when it comes to automating the preparations for robotic welding. Manuel welding is often preferred in those cases, since the creation of a robot program for a new product variation is too time consuming compared to benefit of having a robot execute the process. As mentioned, at Sjørring it can take up to a week to create a program, while it only takes seven hours to weld the product. The AOLP model utilises the concept of elementary operations, which are a way of dividing the entire welding problem into sub problems. Each elementary operations describes a small geometric feature of a 3D model of a product and how to weld it with robotics. For parameterised products the potential for reuse of elementary operations arises, by using carry over information to identify similar weld segments, perhaps just longer or shorter, and transfer it automatically to the CAD model when recognised.

However, this concept of elementary operation does require a lot more research into how to implement it as is is still in its infancy. And this might require better software tools or expanded versions of CAD files to hold process information regarding welding parameters that are seamlessly intertwined with CAD geometry, as it is how the start of the automatic flow from CAD model to robot program begins. There is also the issue of many geometries not having a developed solution for robotic welding yet. Robotic welding is mainly used for straight lines and some curves, a maturity stage it has been stuck at for while. To be a more general and flexible solution identifying and performing robotic welding on more complex geometries will be required, if workpieces will have to be welded without some manual welding. Although, it should be taken into consideration that this project does not have access to the state of the art technologies hidden in large companies that prefer to keep their progress to themselves.

As this project is made in collaboration with Sarivan et al.[1], which looks into applying an elementary operation to parameterisable metal box, where the EOPRs follow the parameterisation automatically, this project looked into how to realise this potential, when fully matured, for, ETOs, among others. Considerations has to be taken into account for both parts, the process definition and the execution of it, to find a viable automatic flow. If automatic elementary operation and an appertaining sequence were to be implemented at low volume production facilities, there would still be challenges to overcome if the execution part should run automatically and seamless. Fixtures would still not be a solution for when only one, two, three, etc. of a product variation is put in production. The fixture would be to expensive for the number of products. Therefore, ways of compensating for the abilities of fixtures need development synchronously with the other part of the automatic flow. The main concerns lie in the calibration of the robot to the workpiece and the compensation thermal expansion of it, which both can create a mismatch between the simulated world and the actually world where the welding is applied. Sensor systems might be a solution here. Therefore, this project further focused on being able to dynamically calibrate to a workpiece in changing environments, using a camera and markers with pose estimation features.

10.2 Dynamic Calibration

The use of a camera and an ArUco marker for pose estimation, meant a calibration of the camera was required firstly, described in chapter 6. This provided the intrinsic camera matrix and distortion coefficients for camera which could then be used to estimate a pose of a marker in the camera frame. And using a customised camera holder, placed stationary in the workspace to hold the camera in a calculable pose (covered in chapter 7) meant that the relation from robot to marker and workpiece could be determined using 4x4 transformation matrices.

The calibration itself managed to provide a result that showed the pose estimation using ArUco markers can be used to calibrate a robot to where the marker is located, for example on a workpiece. The resulting pose estimation stayed within five centimetres of the desired location. Although showcasing the concept to be working, the precision leaves a lot to be desired if to be used in a robotic welding scenario, where the precision likely should stay within a millimetre. Some immediate issues, which might be a cause for this error could be the hardware used. The customised camera holder was not created from a CAD model, meaning the measurements of it have been by hand leaving room for imprecision and human error. The camera used is an inexpensive camera from 2007, which could be upgraded and might lead to better precision. Also fixing the marker to be completely stretched out and flat would eliminate any offset caused by distortion from it not being represented in the image as given in the script in terms of the side length of the marker. A rework of the calibration of the camera may also yield better results.

The potential showed in Mikkelsen et al.[47] of 0.35 millimetres are about the precision desired for welding, even with the goal of high quality weld. It shows that dynamically calibrating to a workpiece using ArUco markers might be a solution to the problem ETOs and low volume productions will face when automating pre-production of robotic welding, since fixtures will be financially unwise to acquire. Now, the next challenges then arise in the form of how to implement markers onto workpieces. This again can be difficult to generalise since all product lines are different. But a structure can be put in place. A solution could be to engrave the marker into the workpiece and feed this to the CAD model. Whether this should be done by the material supplier or by the welding company itself is uncertain and likely will be dependent on the local situation for each manufacturer.

Chapter 11

Future Works

There is a lot of work to be done before a fully automatic solution for pre-production of robotic welding is realised. The concept of elementary operations as the common language between CAD software and Robotic Welding software is still in its infancy and requires much more research and development to find its potential for robotic welding. With EOPR technology not yet mature, the automatic sequencing of it is also yet to be seen. This is, however, not the most pressing issue as the time spend on this task is minuscule compared to the time spent on tag creation on a 3D model for controlling the robots movement around it.

Handling the motion planning when it comes to robotic welding is also in and of itself a challenging task. Ways of achieving this with state of the art welding software is not yet publicly understood as these companies expectedly holds their cards close to the chest. Papers regarding the topic have been looking into dividing up the complex task of creating collision free, constraint filled trajectories to multiple planners with one planner having the sparse but complete oversight, but more development is necessary before seeing it implemented in industry. Then there is the challenge of, in general, not just from the perspective of automatic pre-production, being able to weld more geometries on workpieces than just straight lines and curves, for instance part intersections or inside of corners. This will require a merger of competencies from already experienced professional welders and robotics programmers to make this feasible, which would make robotic welding even more viable.

For low volume manufactures, alternatives to fixtures need to be investigated, if they have a desire to utilise robotic welding more and even automated production preparation of it. More research needs to be conducted into the vast amount of sensor systems and solution which could be incorporated into the robotics system to avoid fixtures and compensate for their abilities. Whether it be vision, sound, touch is only to be found out.

For the dynamic calibration specifically, more work has to be put in to find the causes for the imprecision found in this project. Whether it being a better camera, 3D printed camera holder, more testing or better marker conditions will have to be looked into first. To see this as a part of the solution in the industry more research has to be conducted with multiple manufacturers to understand the best way of implementing the markers into the solution, whether it be engraving them into workpieces or a better recognisable feature can be extracted to make the calibration be seamless.

Bibliography

[1] Ioan-Matei Sarivan, Ole Madsen, and Brian Vejrum Waehrens. "Towards Automatic Welding-Robot Programming Based on Product Model". In: *Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems*. Ed. by Ann-Louise Andersen, Rasmus Andersen, Thomas Ditlev Brunoe, Maria Stoettrup Schioenning Larsen, Kjeld Nielsen, Alessia Napoleone, and Stefan Kjeldgaard. Cham: Springer International Publishing, 2022, pp. 174–181. ISBN: 978-3-030-90700-6.

[2] *Bluestar PLM - Engineering inside Dynamics 365/AX*. `https://bluestarplm.com/`. (Accessed on 05/31/2022).

[3] *PDM Technology*. `https://novi.dk/dk/pdm-technology`. (Accessed on 04/05/2022).

[4] *Bluestar PLM - By PDM technology - We create PLM solutions*. `https://bluestarplm.com/about-us/`. (Accessed on 04/05/2022).

[5] *Bluestar PLM by PDM technology: Oversigt | LinkedIn*. `https://www.linkedin.com/company/bluestar-plm/`. (Accessed on 04/05/2022).

[6] *Conversations with CEO Jørgen Schiønning Larsen (1. March - 30. May)*. Personal Communication. Jørgen Schiønning Larsen, CEO of PDM Technology. Mar. 2022.

[7] *Buckets in front of your wheel loader or excavator | Sjørring*. `https://www.sjorring.com/`. (Accessed on 05/17/2022).

[8] *Dialogue with Sjørring Maskinfabrik (1. March - 30. May)*. Personal Communication. Communication via online meetings, visits, and email. Mar. 2022.

[9] *What is Manufacturing? (Definition, Types and Examples) - TWI*. `https://www.twi-global.com/technical-knowledge/faqs/faq-what-is-manufacturing`. (Accessed on 4/05/2022).

[10] Karen Queen. *What is digital manufacturing? And how are companies harnessing it?* `https://www.themanufacturer.com/articles/what-is-digital-manufacturing/`. (Accessed on 4/05/2022). Sept. 2020.

[11] *What is Digital Manufacturing? (A Definitive Guide) - TWI*. `https://www.twi-global.com/technical-knowledge/faqs/what-is-digital-manufacturing`. (Accessed on 4/05/2022).

[12] *What is Welding? - Definition, Processes and Types of Welds - TWI*. `https://www.twi-global.com/technical-knowledge/faqs/what-is-welding`. (Accessed on 4/30/2022).

[13] Brian Reed. *The History of Welding | Fairlawn Tool Inc.* `https://www.fairlawntool.com/blog/history-of-welding/`. (Accessed on 4/30/2022). Mar. 2018.

[14] *The History of Welding | MillerWelds*. `https://www.millerwelds.com/resources/article-library/the-history-of-welding`. (Accessed on 4/30/2022). Apr. 2020.

[15] *List of welding processes - Wikipedia*. `https://en.wikipedia.org/wiki/List_of_welding_processes`. (Accessed on 4/30/2022).

[16] Jeroen De Maeyer and Eric Demeester. "Benchmarking framework for robotic arc welding motion planning". In: *Procedia CIRP* 97 (Jan. 2021), pp. 247–252. DOI: 10.1016/j.procir.2020.05.233.

[17] *Overview of automatic welding | Welding automation | Automated Welding Basics | KEYENCE America*. https://www.keyence.com/ss/products/measure/welding/automation/grouping.jsp. (Accessed on 4/30/2022).

[18] *The Pros and Cons of Automated Welding*. https://www.assemblymag.com/articles/84103-the-pros-and-cons-of-automated-welding. (Accessed on 4/30/2022). Mar. 1998.

[19] Brian Reed. *Complete Guide to Robotic Welding | Fairlawn Tool, Inc.* https://www.fairlawntool.com/blog/complete-guide-robotic-welding/. (Accessed on 05/05/2022). Dec. 2017.

[20] *Robotic Welding Methods*. https://robotsdoneright.com/Articles/robotic-welding-methods.html. (Accessed on 05/06/2022).

[21] Brian Reed. *Robotic Welding Process - How Robotic Welding Works*. https://www.summitsteelinc.com/resources/blog/robotic-welding-processes/. (Accessed on 05/06/2022). Aug. 2019.

[22] Chris Anderson. *Robotic welding issues and challenges*. https://www.thefabricator.com/thefabricator/article/automationrobotics/robotic-welding-issues-and-challenges. (Accessed on 04/25/2022). Feb. 2014.

[23] E. Hiltunen P. Kah M. Shrestha and J. Martikainen. "Robotic arc welding sensors and programming in industrial applications". In: *International Journal of Mechanical and Materials Engineering* (2015). URL: https://ijmme.springeropen.com/articles/10.1186/s40712-015-0042-y#citeas.

[24] *What Is the Welder Shortage? | Tradesmen International*. https://www.tradesmeninternational.com/construction-management/welder-shortage/. (Accessed on 04/25/2022). June 2019.

[25] Mohamed Abderrahim, Alla Khamis, Santiago Garrido, and Luis Moreno. "Accuracy and Calibration Issues of Industrial Manipulators". In: (2006), p. 18.

[26] *What are the Pros and Cons of Robotic vs Manual Welding?* https://aaatoolandmachine.com/what-are-the-pros-and-cons-of-robotic-vs-manual-welding/. (Accessed on 04/26/2022). July 2019.

[27] Nathan Larkin, Andrew Short, Zengxi Pan, and Stephen Duin. "Automatic program generation for welding robots from CAD". In: July 2016, pp. 560–565. DOI: 10.1109/AIM.2016.7576827.

[28] *UR5 Data sheet*. https://www.universal-robots.com/media/50588/ur5_en.pdf. (Accessed on 5/27/2022). 2016.

[29] *What is a cobot (collaborative robot)? | Cobot Webshop | WiredWorkers*. https://wiredworkers.io/cobot/. (Accessed on 5/27/2022).

[30] *Layout 1*. https://www.binzel-abicor.com/uploads/Content/Germany/PDF-Files/PDF_Files_ROBO/English/wh-w500-bestellunterlage_pro.r138.gb_web.pdf. (Accessed on 04/04/2022).

[31] *Advantages of Using Python for Computer Vision - Full Scale*. https://fullscale.io/blog/advantages-using-python-computer-vision/. (Accessed on 5/29/2022). May 2019.

[32] *Stack Overflow Developer Survey 2020*. https://insights.stackoverflow.com/survey/2020/. (Accessed on 05/24/2022). 2020.

[33] *The State of Developer Ecosystem in 2020 Infographic | JetBrains: Developer Tools for Professionals and Teams*. https://www.jetbrains.com/lp/devecosystem-2020/. (Accessed on 05/24/2022). 2020.

[34] *Welcome to Python.org*. https://www.python.org/. (Accessed on 05/24/2022).

[35] Nick Babich. *What is Computer Vision & How Does it Work? An Introduction | Adobe XD Ideas*. https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/. (Accessed on 05/24/2022). July 2020.

[36] *What is Computer Vision? | IBM*. https://www.ibm.com/se-en/topics/computer-vision. (Accessed on 5/29/2022).

[37] *What Is Computer Vision: Applications, Benefits and How to Learn It*. https://www.simplilearn.com/computer-vision-article. (Accessed on 05/24/2022). Oct. 2021.

[38] *OpenCV: Introduction to OpenCV-Python Tutorials*. https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html. (Accessed on 5/29/2022).

[39] *About - OpenCV*. https://opencv.org/about/. (Accessed on 05/24/2022).

[40] *About RoboDK - Team*. https://robodk.com/about. (Accessed on 05/24/2022).

[41] *Robot Offline programming - RoboDK*. https://robodk.com/offline-programming. (Accessed on 5/28/2022).

[42] *UR+ | RoboDK Simulation Software*. https://www.universal-robots.com/plus/products/robodk/robodk-simulation-software/. (Accessed on 05/24/2022).

[43] P.K. Hansen and H. Sun. "A comprehensive view on benefits from product modularization". In: *11th International Design Conference, DESIGN 2010* (Jan. 2010), pp. 233–242.

[44] *Delfoi ARC Software - ARC welding and laser welding*. https://www.delfoi.com/delfoi-robotics/delfoi-arc/. (Accessed on 05/18/2022).

[45] *DTPS III Desktop Programming & Simulation | Panasonic North America - United States*. https://na.panasonic.com/us/factory-automation-solutions/dtps-iii-desktop-programming-simulation. (Accessed on 05/18/2022).

[46] Jeroen De Maeyer, Mark Versteyhe, and Eric Demeester. "Sampling-based Tube Following for Redundant, Planar Robotic Manipulators". In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vol. 1. 2018, pp. 752–758. DOI: 10.1109/ETFA.2018.8502490.

[47] Casper Mikkelsen, Filip Slezak, Jasper Bjerregaard, Peter B. Torrild, Peter T. Sø rensen, and Violeta Kuldaite. *Zero-Point-Calibration 2.0 (ZPC2) -LEGO-*. Project Report. Fredrik Bajers Vej 7B , DK-9220 Aalborg: Aalborg University, 2020.

[48] Rasul Fesharakifard Nahid Banafian and Mohammad Bagher Menhaj. "Precise seam tracking in robotic welding by an improved image processing approach". In: *The International Journal of Advanced Manufacturing Technology* (2021). URL: https://link-springer-com.zorac.aub.aau.dk/article/10.1007/s00170-021-06782-4#citeas.

[49] *Phillips SPC620NC Camera User Manual*. https://www.download.p4c.philips.com/files/s/spc620nc_00/spc620nc_00_dfu_eng.pdf. (Accessed on 5/26/2022). 2007.

[50] *Phillips SPC620NC Camera Detailed Specification*. https://www.download.p4c.philips.com/files/s/spc620nc_00/spc620nc_00_dsd_eng.pdf. (Accessed on 5/26/2022). 2008.

[51] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008.

[52] Elizabeth Gray. *What Is Focal Length in Photography?* https://photographylife.com/what-is-focal-length-in-photography. (Accessed on 04/02/2022). Apr. 2019.

[53] Kyle Simek. *Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix* ←. https://ksimek.github.io/2013/08/13/intrinsic/. (Accessed on 5/31/2022). Aug. 2013.

[54] Anisha Gartia. *Computer Vision Project*. https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj3/html/agartia3/index.html. (Accessed on 5/31/2022). 2016.

[55] Z. Zhang. "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718.

[56] Paul Anderson Minhua Ma Lakhmi C. Jain. *Virtual, Augmented Reality and Serious Games for Healthcare 1*. Springer Berlin, Heidelberg, 2014.

[57] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco Madrid-Cuevas, and Manuel Marín-Jiménez. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47 (June 2014), 2280–2292. DOI: 10.1016/j.patcog.2014.01.005.

[58] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco Madrid-Cuevas, and Rafael Medina-Carnicer. "Generation of fiducial marker dictionaries using Mixed Integer Linear Programming". In: *Pattern Recognition* 51 (Oct. 2015). DOI: 10.1016/j.patcog.2015.09.023.

[59] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. "Speeded Up Detection of Squared Fiducial Markers". In: *Image and Vision Computing* 76 (June 2018). DOI: 10.1016/j.imavis.2018.05.004.

[60] *OpenCV: Detection of ArUco Markers*. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html. (Accessed on 04/03/2022).

[61] Oleg Kalachev. *Online ArUco markers generator*. https://chev.me/arucogen/. (Accessed on 04/03/2022).

[62] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision.* Upper Saddle River, NJ : Prentice Hall, 1998.

[63] *Convert 3-D rotation vector to rotation matrix - MATLAB rotationVectorToMatrix - MathWorks Nordic*. https://se.mathworks.com/help/vision/ref/rotationvectortomatrix.html. (Accessed on 04/04/2022).