NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Master's thesis

Thomas Fredrik Bech Aschehoug

Development of an assembly tool for automatic sealant inspection and offline programming remote control of an industrial robot for automatic assembly of aircraft parts

Master's thesis in Mechanical Engineering
Supervisor: Olav Egeland
June 2019

NTNU
Norwegian University of
Science and Technology

Thomas Fredrik Bech Aschehoug

Development of an assembly tool for automatic sealant inspection and offline programming remote control of an industrial robot for automatic assembly of aircraft parts

Master’s thesis in Mechanical Engineering
Supervisor: Olav Egeland
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of Science and Technology

Preface

This is the concluding master's thesis of the Master of Science degree in Mechanical Engineering at NTNU, Trondheim. The work was carried out between January and June 2019, and is written in collaboration with Kongsberg Defence and Aerospace AS.

The field of industrial automation was presented to me by Alf Pettersen, Technical Manager at Kongsberg Defence and Aerospace AS. While doing an internship for them during the summer of 2017, I was part of a project that explored different ways of automating an assembly process. Not only was I introduced to the many challenges that come with an automation system, but I also experienced first-hand how big potential it has. Thanks to the internship, I knew in which field I wanted to do my thesis.

After working with automation and offline programming of an industrial robot in my specialization project, it was natural for me to further examine these subjects in this thesis. As a result, the topic of "Automatic assembly of aircraft parts with automatic sealant application and offline programming for remote control of an industrial robot" was formulated.

Trondheim, June 11, 2019

Thomas Fredrik Bech Aschehoug

Acknowledgements

First, I must thank my supervisor Professor Olav Egeland for accepting my project and for continuous support and guidance throughout the semester. Thank you for your supervision.

I would like to thank Associate Professor Lars Tingelstad and Senior Engineer Adam Leon Kleppe from the Robotics and Automation Group led by Professor Egeland, for their advice and support. Your contribution has been much appreciated.

I would also like to thank Kongsberg Defence and Aerospace represented by Alf Pettersen and Simen Hagen Bredvold, for providing me with a meaningful thesis. They have aided me with important equipment and essential knowledge and information about the assembly process discussed. Thank you for your assistance and contribution.

Finally, I would like to thank my fellow students for the academic cooperation and my friends and family for their support, especially Marina and Kjersti for motivating and helping me throughout the semester. You guys are the best.

Abstract

The fourth industrial revolution, Industry 4.0, revolves around the technical advancements of industrial automation. The goal of Industry 4.0 is to increase productivity and efficiency in machinery and manufacturing facilities. This thesis explores how to approach this goal by developing an assembly tool and by using offline programming for remote control of an industrial robot to automatically assemble three aluminium aircraft parts.

The developed assembly tool was used for the two subprocesses of picking and placing the aluminium parts and applying sealant. The process was performed automatically by a UR10 industrial robot. To handle the pick and place subprocess of the assembly, a pneumatic vacuum gripper was used. For the sealant application a custom sealant tool was developed, which used linear actuators to push the plunger head of a caulking gun. The actuators were automatically controlled by the robot controller of the UR10. Both the mechanical and mechatronical aspects of the sealant tool are described in the scope of this thesis. For the UR10 to perform both tasks automatically, a mounting mechanism for wielding the vacuum and sealant tools simultaneously was developed. Finally, the overall performance of the assembly tool is evaluated.

The sealant tool part of the assembly tool was used for automatic sealant application testing. The goals of the tests were to figure out how to tune the sealant tool properly to apply the correct amount of sealant and to assess the reliability of the tool.

The second part of the thesis, revolves around the use of offline programming to remote control the UR10. The offline programming software Visual Components was connected to the UR10 through a Real-Time Data Exchange socket connection. The connection exploited general purpose input registers and a Real-Time Data Exchange Synchronization Loop to send joint variables from the simulation in Visual Components to the UR10. The result was that the UR10 could be remote controlled and programmed to perform the assembly process, without the need for programming skills.

Sammendrag

Den fjerde industrielle revolusjonen, Industri 4.0, dreier seg om den tekniske utviklingen av industriell automasjon. Mlet med Industry 4.0 er ke produktiviteten og effektiviteten i maskiner og produksjonsanlegg. Denne oppgaven undersker hvordan man kan nrme seg dette mlet gjennom utvikle et tilpasset monteringsverkty og ved bruke offline programmering for fjernkontroll av en industrirobot for automatisk montering av tre aluminiumsdeler.

Det utviklede monteringsverktyet ble brukt til de to delprosessene for plukke og plassere aluminiumsdelene og pfre tetningsmasse. Prosessen ble utfrt automatisk av en UR10 industrirobot. For hndtere plukking- og plasseringsprosessen, ble det brukt en pneumatisk vakuumgriper. For pfringen av tetningsmasse ble det utviklet et tilpasset tetningsmiddelverkty, som brukte lineraktuatorer til skyve stempelhodet p en fugepistol. Aktuatorene ble styrt automatisk av robotkontrolleren til UR10 roboten. Bde de mekaniske og mekatroniske aspektene ved tetningsmiddelverktyet er deler av oppgavens omfang. For at UR10 skulle kunne utfre begge oppgavene automatisk, ble det utviklet en monteringsmekanisme for bre bde vakuum- og tetningsmiddelverktyet samtidig. Til slutt evalueres den samlede ytelsen til monteringsverktyet.

Tetningsmiddelverktyet ble brukt til automatisk pfring av tettningsmiddel. Mlet med testene var finne ut hvordan man skulle justere tetningsmiddelverktyet riktig til pfre den korrekte mengden tetningsmasse og vurdere verktyets plitelighet.

Den andre delen av oppgaven dreier seg om bruk av offline programmering for fjernkontroll av UR10. Programvaren Visual Components ble koblet til UR10 via en sanntids datautvekslingsforbindelse. Tilkoblingen utnyttet inngangsregistre og en synkroniseringslkke for datautveksling i santid for sende leddvariabler fra simuleringen i Visual Components til UR10. Resultatet var at UR10 kunne fjernstyres og programmeres til utfre monteringsprosessen uten behov for programmeringsferdigheter.

Table of Contents

Preface i

Acknowledgements iii

Abstract v

Sammendrag vii

Table of Contents xi

List of Figures xvi

Abbreviations xvii

1 Introduction 1
1.1 Background 1
1.2 Problem description 2
1.3 Approach 3
1.4 Structure of the thesis 3

2 The assembly process 5
2.1 Today's AS-IS model 5
2.1.1 Frame 5
2.1.2 Cover 6
2.1.3 Lid 6
2.1.4 Application of sealant 7
2.2 Future TO-BE model 8

3 Preliminaries **11**
3.1 The industrial robot: a UR10 11
3.2 Linear actuators 14
3.2.1 How they work 14
3.2.2 Digital output control of the linear actuators 14
3.3 Vacuum gripper theory 15
3.3.1 Suction cup theory 16
3.3.2 The operating principle of the Venturi vacuum generator 17
3.4 Offline programming 18

4 Developing the assembly tool **21**
4.1 The automatic tool changer workaround 21
4.1.1 Developing the dual tool mount 22
4.2 Developing the sealant tool 26
4.2.1 Sealant tool components 26
4.2.2 The mechanical setup:A modified caulking gun with supports . . 31
4.2.3 The mechatronical setup: A DPDT switch system 40
4.2.4 The final design of the sealant tool 48
4.3 Setting up the vacuum tool 50
4.3.1 Vacuum tool components 50
4.3.2 The final vacuum tool setup 54
4.4 The final assembly tool design 56

5 Offline programming of the UR10 with Visual Components **59**
5.1 The motivation behind using VC to control the UR10 59
5.2 Establishing connection between Visual Components and the UR10 . . . 60
5.2.1 An introduction to relevant computer science 60
5.2.2 The connection process 62
5.2.3 Connectivity issues 66
5.3 Connectivity via URSim 68
5.4 The workaround 70
5.4.1 Setting up the registers 70
5.4.2 The RTDE Synchronization Loop 72
5.4.3 Establish the connection 72
5.5 Scientific grounding and rarity assessment 73

6 Results **75**
6.1 Results of developing and testing the assembly tool 75
6.1.1 Results of developing the dual tool mount 75
6.1.2 Results of developing the sealant tool 75

6.1.3 Results of setting up the vacuum tool 82
6.1.4 Results of the finished assembly tool 82
6.2 Results of offline programming of the UR10 with Visual Components . . 84
6.2.1 Results of successfully establishing the workaround connection . 84
6.2.2 Results of the literature review 86

7 Analysis and discussion 89
7.1 Discussion of developing and testing the assembly tool 89
7.1.1 Discussion of the dual tool mount 89
7.1.2 Discussion of the sealant tool 90
7.1.3 Discussion of the vacuum tool 95
7.1.4 Discussion of the finished assembly tool 96
7.2 Discussion of using offline programming with Visual Components to control the UR10 97
7.2.1 Discussion of establishing the workaround connection 98

8 Conclusion 103
8.1 Conclusion of developing and testing the assembly tool 103
8.1.1 The dual tool mount 103
8.1.2 The sealant tool 104
8.1.3 The vacuum tool 107
8.1.4 Future work on the assembly tool 107
8.2 Conclusion of the offline programming for remote control of the UR10 with Visual Components 108
8.2.1 Future work on the remote control connection 109

Bibliography 110

List of Figures

2.1 Frame seen from three different angles. 6
2.2 Cover seen from three different angles. 6
2.3 Lid seen from four different angles. 7
2.4 Complete assembly of the three parts. 7
2.5 Sealant application surfaces. 8
2.6 Sealant squeeze-out area. 8

3.1 The UR10. 11
3.2 The joints of the robot: A: Base, B: Shoulder, C: Elbow, and D, E, F: Wrist 1, 2, 3 respectively. 12
3.3 The control box and the electrical interface. 13
3.4 The robot controller. 13
3.5 Circuit diagrams of the linear actuators extending and retracting. 14
3.6 The DO tab of the I/O panel in the robot controller's GUI. 15
3.7 The electrical interface of the control box. 15
3.8 Actual connections 16
3.9 Suction cup theoretic drawing. The arrows indicate the atmospheric pressure 16
3.10 Operating principle of the vacuum generator 17

4.1 The dual tool mount. 22
4.2 The three initial parts of the dual tool mount. 23
4.3 The first dual tool solution. 23
4.4 CAD replication, showing the plunger to be longer than the vacuum tool when fully pulled back. 24
4.5 Solution to the design complications: Dual tool with extender. 24
4.6 Solution to the rotation complications: Welded models of the dual tool. . 25

4.7 The finished dual tool mount. 25
4.8 The sealant tool. 26
4.9 The main components of the sealant tool. 27
4.10 Modified caulking gun. 28
4.11 The actuator supports. 28
4.12 The linear actuators. 29
4.13 Parallel connection of the two linear actuators. 30
4.14 The DPDT switch. 30
4.15 Early concept model of the sealant gun mounted to the dual tool mount. . 31
4.16 The ring support during manufacturing and finished design with actuators mounted. 32
4.17 The polymer support. 33
4.18 The polymer support. 34
4.19 Broken polymer support. 35
4.20 The silver steel actuator support. 35
4.21 CAD models of and the actual plunger head support. 36
4.22 Plunger head support on the sealant tool. 36
4.23 Creating the polymer support – Vol 1. 37
4.23 Creating the polymer support – Vol 2. 38
4.23 Creating the polymer support – Vol 3. 39
4.24 The entire CAD assembly and the final mechanical setup of the sealant tool. 39
4.25 The DPDT switch system. 40
4.26 The DPDT switch with wire diagrams. 40
4.27 DPDT switch used to reverse a DC motor. 41
4.28 The first DPDT switch solution. 42
4.29 Circuit diagram of the DPDT switch in its two possible states. The top circuit shows the extend state, while the bottom shows the retract state. . . 43
4.30 The modified DPDT switch solution. The two wires to the right go to the motors of the linear actuators, whereas the four wires to the left are connected to the two voltage sources; the robot controller and the battery pack. 43
4.31 Circuit diagrams of the multipurpose DPDT solution. The blue wires are the battery pack circuit. 44
4.32 Development of the cover. Top: CAD model and printing. Middle: Results showing support structure and the inside. Bottom: Battery pack and switch neatly packed inside the cover. 46
4.33 The voltage converter setup. 47
4.34 Closeups of the connection ports of the voltage converter. 47
4.35 The parallel circuit mounted on the assembly tool and the DPDT switch mounted on the robot base structure. 48

4.36 The final design of the sealant tool. 49
4.37 The vacuum tool. 50
4.38 The pressure reduction valve. 51
4.39 The two components of the solenoid valve. 52
4.40 Turning the vacuum gripper off using DO signals from the robot controller. 52
4.41 Turning the vacuum gripper on using DO signals from the robot controller. 53
4.42 The Venturi vacuum generator. 53
4.43 The vacuum gripper and components set. 54
4.44 Chronological process of creating vacuum in the gripper – Vol 1. 55
4.44 Chronological process of creating vacuum in the gripper – Vol 2. 56
4.45 Finished assembly tool setup. 57
4.46 Control panel of the assembly process. 57

5.1 The CAD model of the assembly tool imported into the simulated UR10 virtual environment in VC. Note: The caulking gun is not included in this picture. 61
5.2 Ping command sent from the computer, verifying a connection. 63
5.3 Enabling the Connectivity feature in VC. 63
5.4 The Connectivity tab shows up after enabling it and restarting VC. 63
5.5 Adding an RTDE server to the UR10S under the Connectivity tab. 64
5.6 About tab in PolyScope. 65
5.7 Testing and applying the IP and the RTDE port in VC. 65
5.8 Creating the variable pairs. 66
5.9 The SocketTest connected to the UR10. 67
5.10 The I/O screen on the PolyScope controller before and after sending the DO2 command from the SocketTest on the computer. Notice under "Digital Output", number 2 is turned on. 68
5.11 PolyScope running on Linux in the virtual machine on the same computer as VC is used with Windows. 69
5.12 VC (to the right) and URSim connected with two-way communication. . . 69
5.13 The Create Variable Pairs panel with one variable pair selected. 71
5.14 Successfully created registers. 71
5.15 RTDE synchronization loop script on the PolyScope controller 72
5.16 Connection succeeded. 73

6.1 Stable and sturdy final design. 76
6.2 Caulking gun design issue. 76
6.3 Sealant cartridge issue. 77
6.4 Flat metal ring solution to the caulking gun design flaw. 77
6.5 Broken plunger head support and the quick fix solution to it. 78

6.6 The inside of the DPDT box showing the melted wires. 79
6.7 Results of the initial tests. 79
6.8 Result of the leakage test. 80
6.9 Results of the five application tests. 81
6.10 Sealant curling instead of sticking to the surface. 81
6.11 Results from the paper tests. 81
6.12 Broken and proper sealant cartridges. 82
6.13 The final assembly tool weight. 83
6.14 Volume of the frame. 83
6.15 Remote controlling the UR10 with VC. 85
6.16 Robot postures and joint variables during OLP remote control of the UR10. 86
6.17 Joint variables of the UR10S and UR10 during OLP remote control. . . . 86

7.1 Potential redesign of the sealant tool. 91
7.2 An automatic DPDT solution, using a signal-controlled relay to switch the DPDT switch. 93
7.3 Sealant application errors. 93
7.4 The CAD model of the assembly tool imported into the UR10S virtual environment in VC. Note: For some reason the components of the caulking gun would not import properly. 97
7.5 Exchanging the KUKA KR 120 with the UR10, but keeping the robot program. Notice: all robot position frames turned red; only two of the original positions were within reach of the UR10. 100

8.1 The finished dual tool mount. 104
8.2 Final mechanical setup mounted on the dual tool mount. 105
8.3 An automatic DPDT solution. 106
8.4 Potential redesign of the sealant tool. 106
8.5 Remote controlling the UR10 with VC. 109

Abbreviations

CAD	=	Computer-Aided Design
DO	=	Digital Output
DPDT	=	Double-Pole Double-Throw
GUI	=	Graphical User Interface
IP	=	Internet Protocol
KDA	=	Kongsberg Defence and Aerospace AS
RTDE	=	Real-Time Data Exchange
UR	=	Universal Robots
TCP	=	Transmission Control Protocol
VC	=	Visual Components Premium 4.0

Chapter 1

Introduction

1.1 Background

It seems like everyone is talking about Industry 4.0 these days. Described as the fourth industrial revolution, it revolves around the technical advancements of industrial automation with the goal to increase productivity and efficiency in machinery and manufacturing facilities. Kongsberg Defence and Aerospace AS (KDA) wish to pursue this trend of automation in their production, and the scope of this project thesis has been formulated in cooperation with them.

Within industrial automation, robots have many advantages to humans. They can be used to achieve higher precision, strength, and speed in production; and they can work non-stop, with little to no maintenance. The results are lower production costs, higher productivity and efficiency, and a more flexible solution. The latter could be important for KDA as they produce a wide variety of products, which the automatic production must be able to handle. This is paramount for the transition into automatic manufacturing for such a versatile company. Before KDA can take industrial automation further into use they have requested research into the fields of automatic assembly of aluminium aircraft parts.

1.2 Problem description

The question the thesis is trying to answer is how the assembly of aluminium aircraft parts can be performed automatically with the use of an assembly tool for automatic sealant application, and offline programming of an industrial robot.

The problem description is formally stated as these two problem statements:

Problem statements

1. Develop an assembly tool for testing automatic sealant application.
2. Investigate how to use offline programming for remote control of an industrial robot.

The problem statements are divided into the following objectives:

Objectives:

- Give the reader an introduction to the assembly process of the three aircraft parts, stating the AS-IS and TO-BE models.
- Introduce the UR10 used for the assembly, along with other necessary preliminaries for understanding the implementations of this thesis.
- Develop a mounting mechanism for wielding two assembly tools simultaneously on the UR10.
- Develop a sealant tool for automatic sealant application.
- Test the sealant tool in automatic sealant application experiments.
- Build a vacuum tool setup for picking and placing the aluminium parts of the assembly.
- Activate an RTDE server connection in Visual Components to the UR10.
- Establish an offline programming connection between Visual Components and the UR10.
- Remote control and monitor the UR10 in Visual Components.

1.3 Approach

The approach to answer the problem statement of this thesis has been of both theoretical and practical nature. The theoretical methodology involved researching literature, studying methods, and collecting data to map existing technologies and solutions. Numerous papers and articles about industrial automation, automatic assembly, and offline programming have been studied to provide the knowledge for developing a full assembly tool and an offline programming solution. Chapters 4 and 5 are based on this research and make up the foundation for the following implementations, which are analyzed and discussed later in Chapter 7. The practical aspect of this thesis consisted of developing and testing these implementations. This included the development of an automatic assembly tool, tests of the automatic sealant application, and an offline programming connection to remote control a UR10 industrial robot with Visual Components.

1.4 Structure of the thesis

The thesis revolves around the assembly process of three aluminium aircraft parts. The assembly is wanted automated. The two main tasks of the thesis are developing an assembly tool for testing automatic sealant application and offline programming for remote control of an industrial robot. To do this, and advance with the automation of the assembly, different technologies, methods, and implementations of how to do the assembly are presented.

- Chapter 2 briefly introduces the assembly process and the three aluminium parts. The process of applying sealant is described to provide a better understanding for one of the main objectives of the thesis.
- Chapter 3 provides required preliminaries. First, the UR10 industrial robot used for the assembly experiments is presented. The linear actuators are the motors of the sealant tool and are also given an introduction. Lastly, theory about vacuum grippers and offline programming is presented.
- Chapter 4 describes the development of the assembly tool. The chapter addresses the assembly tool's three main parts; the dual tool mount, the sealant tool and the vacuum tool. Both the mechanical and mechatronical setups of the sealant tool are explained more in depth. Finally, the final assembly tool design is assessed, especially the sealant tool.
- Chapter 5 describes the offline programming of the UR10 with Visual Components. How the connection was established, with the full connection process is given. The

functioning workaround solution is presented including how to set up registers, and the RTDE Synchronization Loop. All the procedures have step-by-step guides.

- Chapter 6 presents the results of the implementations and experiments of Chapter 4 and 5, without interpreting their meaning. Chapter 6, 7, and 8 are twofold and first address the contents of Chapter 4, then finishes with the contents of Chapter 5.
- Chapter 7 analyzes and discusses the results and challenges of the implementations. The results are evaluated compared to the scope of the thesis, the bigger picture and the assembly process at KDA.
- Chapter 8 presents a conclusion of the analysis and discussion. Suggestions of future work is also proposed.
- The digital appendix contains videos of the sealant application tests and the remote control of the UR10 with Visual Components.

Chapter 2

The assembly process

2.1 Today's AS-IS model

To better understand the content of the thesis, a description of the assembly process is presented. Three aluminium parts are assembled by bolting the parts together. Before bolting, sealant must be applied to make sure that no water will leak in between the parts. This is an important step of the assembly process. The full process is currently done manually.

The three parts are described in the following section along with the sealant application process. Computer-Aided Design (CAD) models of all parts are also given. All three parts and specifications given are fictitious to keep the original models and tolerances secret, however, they still represent the geometric properties of the original parts well enough to imitate the original process.

2.1.1 Frame

The first part is the frame. It is the main part of the aluminium structure, and the two other parts are mounted on this one. It is hollow to be as light as possible and it is predrilled with holes. The frame can be seen from three different angles in Figure 2.1.

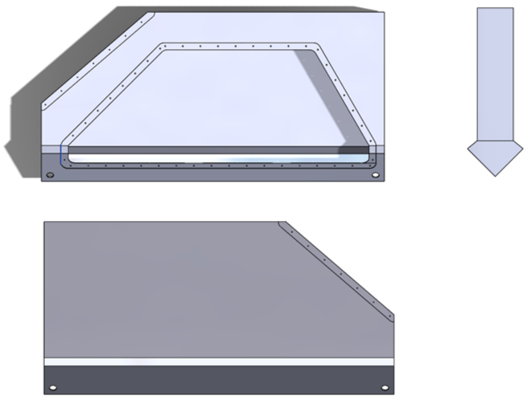


Figure 2.1: Frame seen from three different angles.

2.1.2 Cover

The second part is the cover. The cover is a thin lid that makes it possible to hollow out all the unnecessary weight from the frame. The part is predrilled and counterbored. The cover can be seen from three different angles in Figure 2.2.

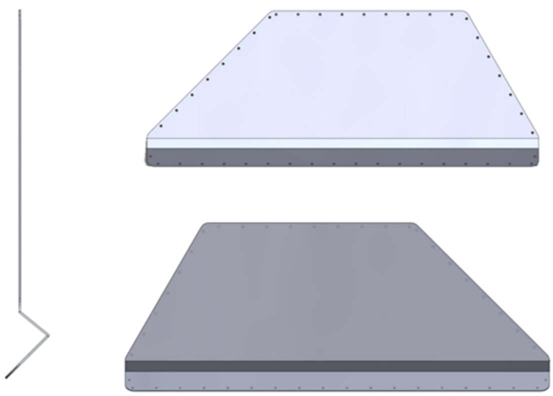


Figure 2.2: Cover seen from three different angles.

2.1.3 Lid

The third part is the aerodynamic lid. This is a lid, which is assembled so it points in the flight direction of the aircraft. It has hollow profile to be aerodynamic. It is both predrilled and counterbored. The lid can be seen from four different angles in Figure 2.3. The complete assembly of the three parts can be seen in Figure 2.4.

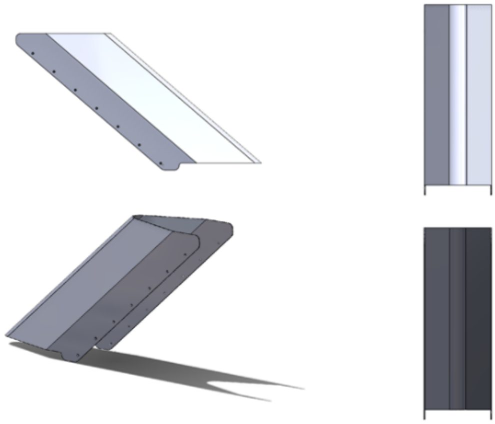


Figure 2.3: Lid seen from four different angles.

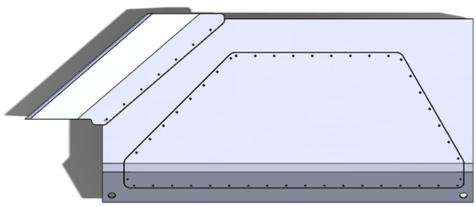


Figure 2.4: Complete assembly of the three parts.

2.1.4 Application of sealant

The sealant is a product that resembles silicone. It seals the parts together and keeps the water out. The sealant application areas are on the joining surfaces of the three parts, marked in blue in Figure 2.5.

There must be applied sufficiently with sealant to form a 360 degree squeeze-out. This means it must escape an equal amount of sealant in all directions along all joining surfaces after assembly of the three parts. The reason behind this is to verify that enough sealant has been applied. If there is a break in the squeeze-out, there is no way to guarantee that the assembly is waterproof. If this happens, the parts must be disassembled to apply more sealant, and then be reassembled again. This is costly and time-consuming, and it is

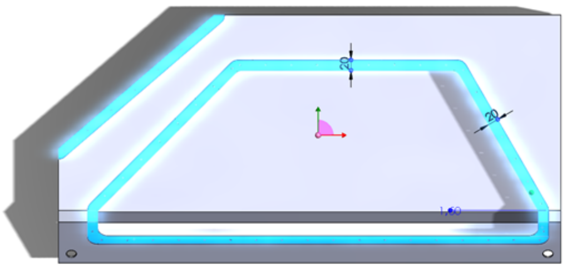


Figure 2.5: Sealant application surfaces.

therefore important to guarantee that the sealant has been applied properly. An example of a 360 degree squeeze-out is shown in Figure 2.6.

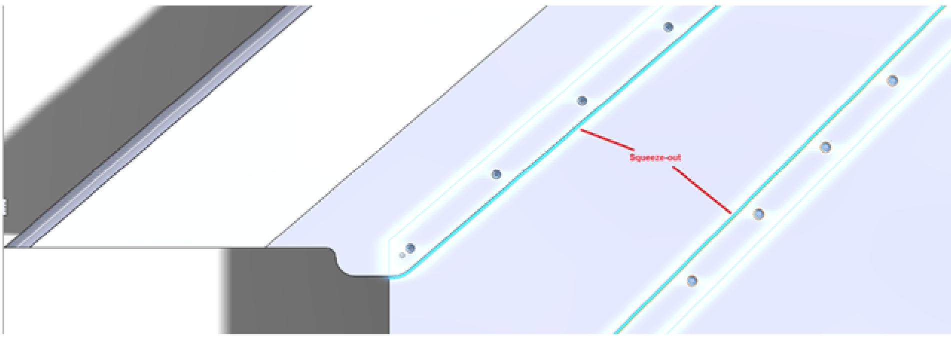


Figure 2.6: Sealant squeeze-out area.

2.2 Future TO-BE model

The aim is to make the manual assembly process automatic. The automatic process would consist of picking and placing the frame on the workbench, applying the correct amount of sealant on the joining surfaces as seen in Figure 2.6, and mounting the cover and the lid to the frame. To do this, an industrial robot must be implemented to carry out the physical handling operations of the aluminium parts. Additionally, a way to be sure that sufficient sealant has been applied must be implemented to guarantee a waterproof assembly.

During the project thesis, this was worked with and an offline programming simulation of the robot cell was created along with two image processing algorithms to automatically inspect the sealant application.

This master's thesis further focuses on making the manual assembly process automatic. An industrial robot is taken into use to test the steps of the assembly process. To realize the assembly, a custom assembly tool was developed, with a new approach to guaranteeing a proper sealant application. The offline programming software of the project thesis is also implemented with the industrial robot. Preliminaries of all three processes of developing the tool, testing the sealant application, and offline programming simulation are described in the following chapter.

Chapter 3

Preliminaries

3.1 The industrial robot: a UR10

To get to the future TO-BE model and perform the assembly process automatically, an industrial robot from Universal Robots was used: a UR10. The UR10 is Universal Robots' biggest robot. It is able to lift payloads of up to 10 kg, with a reach of 1300 millimeters from the base joint [1]. The UR10 used in this thesis was a CB3-series. The robot is mounted on a metal base structure, and can be seen in Figure 3.1.

Figure 3.1: The UR10.

The UR10 consists of six rotational joints. The joints from the bottom and out are called:

base, shoulder, elbow, and wrist 1, 2, and 3. The tool flange of the robot is at the end of wrist 3. All joints have joint ranges at ±360 degrees. The joints can be seen in Figure 3.2.

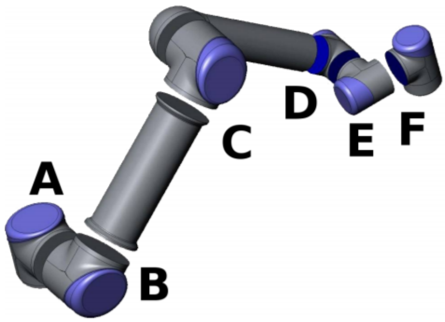


Figure 3.2: The joints of the robot: A: Base, B: Shoulder, C: Elbow, and D, E, F: Wrist 1, 2, 3 respectively.

The control box

The UR10 is powered by the control box, which in this cell layout is positioned underneath the UR10 inside the metal base frame. The box has three main cables going out from it: the power cable to the outlet in the robot cell, a robot cable to the UR10, and the last cable go to the teach pendant.

Inside the control box, the electrical interface can be found. The control box and its inside, showing the electrical interface, can be seen in Figure 3.3. The other cables going out of the control box are for the safety switch and the assembly tools explained in the next chapter, Chapter 4.

(a) The control box. **(b)** The electrical interface.

Figure 3.3: The control box and the electrical interface.

The robot controller

The robot comes with a teach pendant. It is connected to the control box and has a touch screen panel. The software running on the teach pendant is called PolyScope. The PolyScope teach pendant is throughout this thesis called the robot controller or the PolyScope controller. The robot controller can be seen in Figure 3.4.

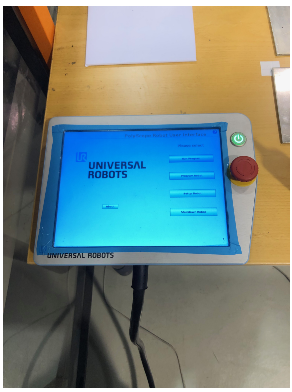
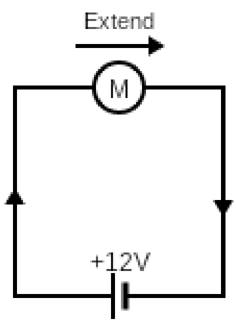


Figure 3.4: The robot controller.

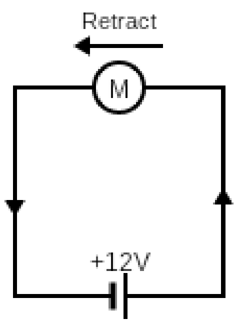
3.2 Linear actuators

3.2.1 How they work

Two linear actuators are used to create an automatic sealant tool. The development of the tool is described in Chapter 4.2. The linear actuators used have three operating states: extending, retracting and standing still. By applying voltage and reversing the polarity on the actuators, thus changing the flow direction of the current, the actuators can be controlled to extend or retract. To make them stand still, the voltage is cut off. The simplest way to connect an actuator to the robot controller would be to connect the power wire (the red wire) of the actuator to one of the digital output (DO) ports (DO1 in this case), and the ground wire (the black wire) to 0V. The actuator will then receive power through its power wire from the robot controller whenever the DO port is set to high. The power will run through the motor of the actuator the normal way, causing it to extend and exit through the black ground wire into the 0V ground port of the robot controller. The circuit diagram of this connection can be seen in Figure 3.5a. Connecting the actuator the other way around effectively reverses the electricity conducted through the motor, which causes the actuator to retract, as seen in the circuit diagram of Figure 3.5b.



(a) Extending.



(b) Retracting.

Figure 3.5: Circuit diagrams of the linear actuators extending and retracting.

3.2.2 Digital output control of the linear actuators

For the robot to apply the sealant autonomously, it must be able to control the linear actuators through sending digital output signals via the robot controller. The digital outputs of the controller are binary, and can be set to either high (True) or low (False). High will turn the power on and low will turn it off. A picture of the input/output (I/O) panel of the robot controller's graphical user interface (GUI), showing the DO tab with all ports set to low, can be seen in Figure 3.6.

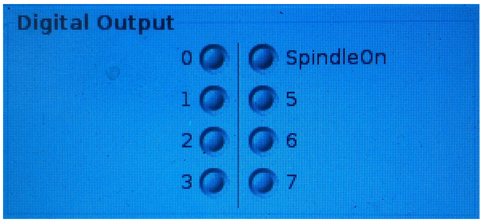
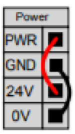
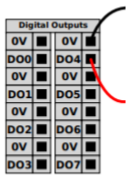


Figure 3.6: The DO tab of the I/O panel in the robot controller's GUI.

The digital output ports are internally connected to the robot controller's own power source in the control box. This means that even if a DO port is set to low it will not work as 0V ground – it will simply be turned off and not conduct electricity. However, all the ports marked 0V in the robot controller are connected to ground, so these were used instead. The internal connections of the control box can be seen in Figure 3.7a, whereas the DO connections can be seen in Figure 3.7b.



(a) The internal connections of the control box.



(b) The DO connections.

Figure 3.7: The electrical interface of the control box.

The actual connections for digital output control from the control box, can be seen in Figure 3.8. The black and red wires connected to 0V and DO1 are to control the linear actuators of the sealant tool, and the grey wire connected to 0V and DO3 go to the solenoid valve of the vacuum gripper.

3.3 Vacuum gripper theory

A pneumatic vacuum gripper with suction cups was used to pick and place the aluminium parts of the assembly process. The vacuum tool setup is explained in Chapter 4.3. To understand how it works basic vacuum theory is provided in this section.

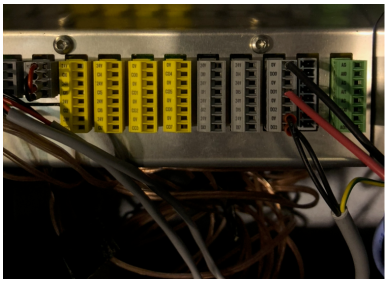


Figure 3.8: Actual connections

3.3.1 Suction cup theory

Unfortunately, the suction cups do not automatically attach themselves when they come in contact with the surface of the aluminium parts. However, by creating a pressure difference between the inside and the outside of the suction cups, it is possible to create a suction effect, which allows the suction cups to grip the parts [2]. This pressure difference occurs when the atmospheric pressure outside the suction cups is greater than the pressure between the suction cups and the workpiece.

To lower the pressure inside the suction cups below the atmospheric pressure outside, the air between the suction cups and the workpiece can be sucked out. If the suction cups are in contact with the surface of the workpiece when the air inside it is being sucked out, and no air can enter from the sides, a vacuum is generated inside the suction cups and they are able to attach to the workpiece. The drawing in Figure 3.9 shows how a suction cup works. On the right side, all the air between the suction cup and the surface of the workpiece has been sucked out, causing a vacuum, which creates the pressure difference that allows the suction cup to grip the workpiece.

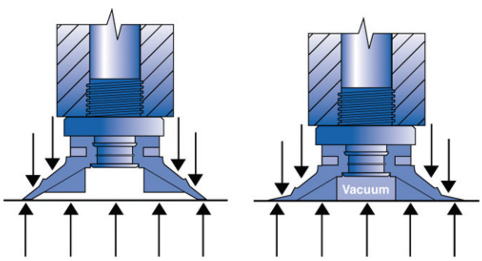
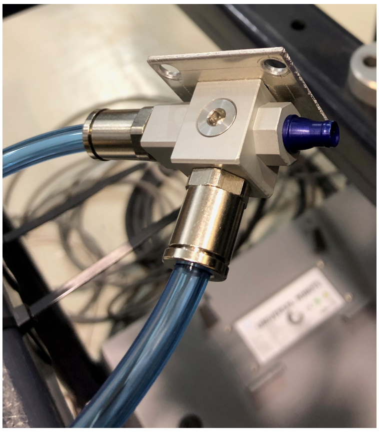


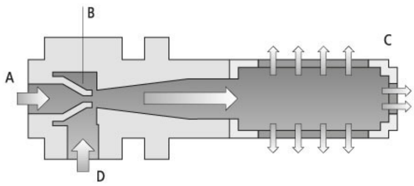
Figure 3.9: Suction cup theoretic drawing. The arrows indicate the atmospheric pressure

A bigger difference between the atmospheric pressure outside and the vacuum pressure inside the cups, or a larger effective area of the cups acting on the workpiece, results in greater gripping force [3]. The pressure difference needed can be created with a vacuum generator.

3.3.2 The operating principle of the Venturi vacuum generator



(a) The vacuum generator.



(b) Drawing of the inside of the vacuum generator.

Figure 3.10: Operating principle of the vacuum generator

The pneumatic vacuum generator used in this thesis was a vacuum ejector from the robot lab, which functioned based on the Venturi principle. It is explained in the following with references to Figure 3.10b [4].

- Definition on Wikipedia: The Venturi principle is "the reduction in fluid pressure that results when a fluid flows through a constricted section (or choke) of a pipe".
- Compressed air flows into the ejector (A).
- Due to the reduced cross section of the Venturi nozzle (B), the compressed air is accelerated.
- Because of this acceleration, the dynamic pressure increases, while simultaneously the static air pressure decreases in the nozzle. Once the compressed air has passed the Venturi nozzle, the accelerated air expands again and a vacuum is generated.
- This causes air to be sucked in through the vacuum gripper connection (D) into the ejector – effectively sucking out the air from the vacuum gripper and the suction cups. As mentioned previously, this causes vacuum in the suction cups, given that they are in contact with the workpiece surface).

- The compressed air and the air sucked out of the vacuum gripper, escapes from the ejector through the silencer (C).

3.4 Offline programming

The use of offline programming software was studied during the project thesis of last semester (autumn 2018). The case of Visual Components was examined in detail. As some knowledge of offline programming is essential for understanding the content of Chapter 5, some theory from the project thesis is provided. It summarizes what offline programming is, as well as mentioning it's advantages and limitations compared to online programming.

Chapter theory: Brief summary on offline programming

Whereas online programming is the method of jogging the robot around to acquire and register waypoints, offline programming (OLP) is the method of controlling the entire robot cell through a virtual reality 3D simulation. Many robot manufacturers have their own proprietary software for this, which is compatible with their own hardware. However, generic OLP software can be more flexible as they often are compatible with multiple hardware manufacturers. Visual Components (VC) is an example of a generic OLP software, which is compatible with several manufacturers such as Fanuc, Kuka, ABB, etc.

The biggest differences between online programming and OLP are first and foremost that the latter does not require the actual robot for the programming, because it happens in the virtual environment. This means that production downtime and the cost of change between programs can be reduced, because the robot can continue to produce while the simulation is being used to design and test new robot programs. OLP is therefore more cost-efficient than its counterpart.

Additionally, OLP is more flexible. It can handle workpieces with more complex geometries, or processes that require advanced handling and may be physically difficult for a human operator to perform with online programming.

Moreover, the 3D simulations of entire work cells can be tested. Robot reachability limitations or collision violations can be proved to see if the workspace layout combined with the robot programs are actually physically feasible – already before they are implemented. For instance, it is possible to verify if all motion statements have robot joint configurations that are within their respective range of motion, or if the UR10 at any time is close to any unwanted singularities. If the testing fails, the robot paths must be changed, and it is both quicker and cheaper to correct the robot programs in a simulation, compared to online programming where the robot programs must be implemented to be tested. This flexible, pre-implementation testing minimizes errors and guarantees productivity and safety already at the design phase of programming.

Lastly, programming offline means that the operator is moved away from the robot environment. This makes it safer to create new robot programs for hazardous operations or operations in dangerous environments.

The advantages of OLP are many, yet it is not as intuitive as online programming and does require some programming skills. CAD models of both the robot cell and the workpieces are needed, and the quality of the process relies heavily upon their designs. If they are not modeled accurately enough, the OLP will result in wrong outcomes in the physical world. Because of this, it is often necessary to perform some form of post-processing or touch-up. This is to assure that the process will behave in the real world as it does in the simulation and ensures high quality of the results.

Chapter 4

Developing the assembly tool

To perform the assembly process automatically, the proper tools are needed for the sub-processes of picking and placing the parts and applying the sealant. A sealant application tool and a vacuum gripper were used in this thesis. This chapter explains how these two tools were mounted to the UR10 robot, how the sealant tool was designed and manufactured, and how the vacuum tool was put together.

The assembly tool was used for automatic sealant application tests as part of the assembly process of the aluminium parts. The assembly process has been simplified, however, and the aluminium parts from Chapter 2 have been changed with small, flat, aluminium plates. The aluminium plates worked well in representing the aluminium parts for the tests of automatic sealant application.

4.1 The automatic tool changer workaround

A way of performing both sub-processes of the assembly automatically was needed. This could either be done by using two robots with one tool each, or by having one robot using both tools e.g. with an automatic tool changer. During this thesis just one UR10 could be used, and investing in another robot would be outside the budget of the project, so the first solution was not an option.

For the UR10, companies like Zimmer Group and Universal Robots have existing automated tool changer solutions [5][6]. From Universal Robots, a SmartShift automated tool changer starter set, would at the time of writing (April, 2019) cost 30.000 NOK. This would only allow the UR10 to change between two tools. This thesis only experimented with two tools, but the total assembly process requires more than this. Instead, a cheaper workaround was thought out and a custom dual tool mount was developed for the experiments.

4.1.1 Developing the dual tool mount

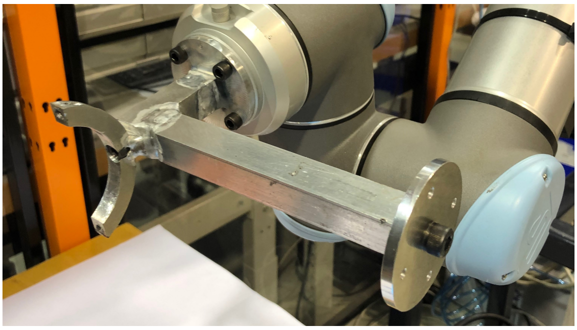
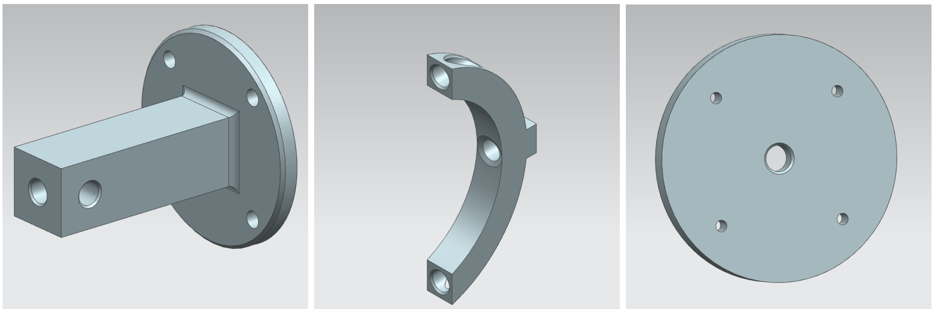


Figure 4.1: The dual tool mount.

The workaround solution was to mount both tools on a dual tool mount, with 180 degrees relative rotation of the tool center point of each end-effector. The UR10 could then operate both tools by simply rotating the wrist 3 joint (seen in Figure 3.2) 180 degrees.

Initial designs

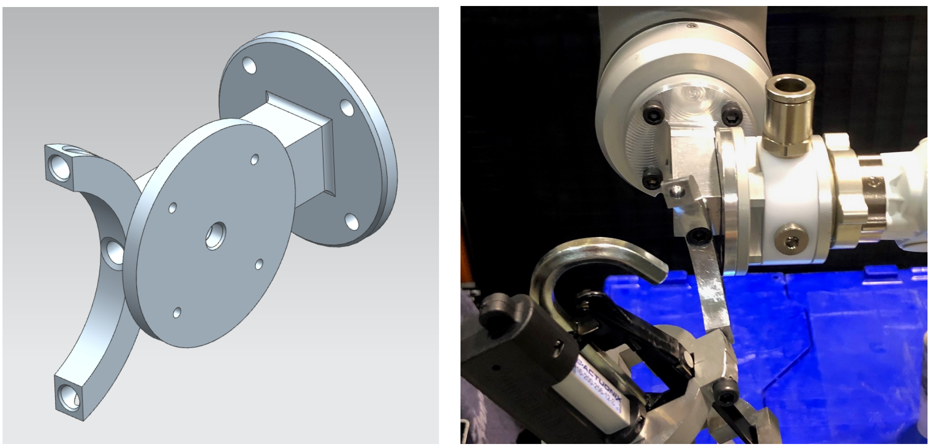
The initial CAD model of the dual tool mount consisted of three parts. A mount between the dual tool and the robot's tool flange (with an extension for mounting of the two other parts), a tool mount to the sealant tool, and a tool mount to the vacuum tool. All parts can be seen in Figure 4.2.



(a) Mount to the UR10's tool flange. **(b)** Mount to the sealant tool. **(c)** Mount to the vacuum tool.

Figure 4.2: The three initial parts of the dual tool mount.

An assembly of the three parts can be seen in Figure 4.3a. The UR10 has a limitation of maximum payload of 10 kg. Therefore, the dual tool was manufactured in aluminium. Keeping the total weight of the assembly tools down to a minimum, is a recurring issue throughout this chapter. The dual tool mount assembly was mounted on the UR10 and tested with both tools attached, shown in Figure 4.3b.



(a) CAD design of the three pieces assembled. **(b)** Dual tool mounted on the UR10 with tools attached (sealant tool not completely attached).

Figure 4.3: The first dual tool solution.

Design complications

It was discovered that when the sealant tool had a loaded sealant cartridge, the plunger of the tool would extend beyond the vacuum gripper. The plunger thus hindered the vacuum gripper from reaching the surface of the workpieces to pick them up. This is demonstrated in a CAD replica of the situation in Figure 4.4.

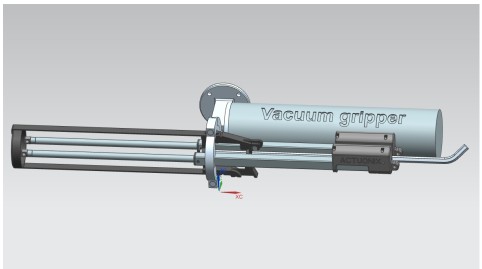


Figure 4.4: CAD replication, showing the plunger to be longer than the vacuum tool when fully pulled back.

This could be solved by rotating the sealant tool 90 degrees so it would be perpendicular to the vacuum tool, however, an extension was manufactured instead. The extension was attached between the pieces in Figure 4.2a and 4.2c, and can be seen in Figure 4.5.

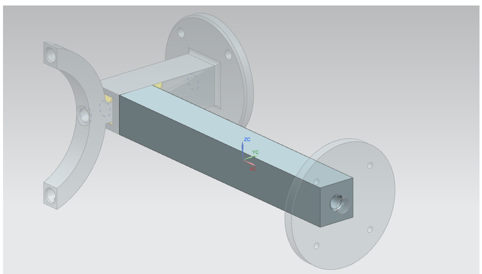
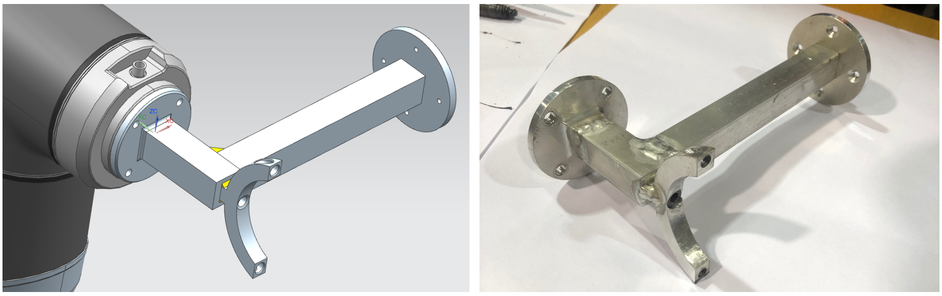


Figure 4.5: Solution to the design complications: Dual tool with extender.

Rotation complications

The extended solution worked fine for a while, however, complications of unwanted rotations appeared. All three connecting surfaces between the dual tool mount parts eventually became a bit loose and risked rotating the tools out of position. An example of this can be seen with the half-attached sealant tool in Figure 4.3b, showing the sealant part rotated out of position. The cause of this was that the dual tool parts were designed with single screw fastening mechanisms. Another type of fastening design should have been developed, i.e.

using two smaller screws instead and thus removing the possibility of rotation. Wanting to use the already-made components, it was decided to weld the pieces together. The welded design can be seen in both CAD model and physical shape in Figure 4.6. Welding aluminium is fairly difficult, so this was performed by professionals from the workshop at NTNU Valgrinda, not the authir if this thesis.



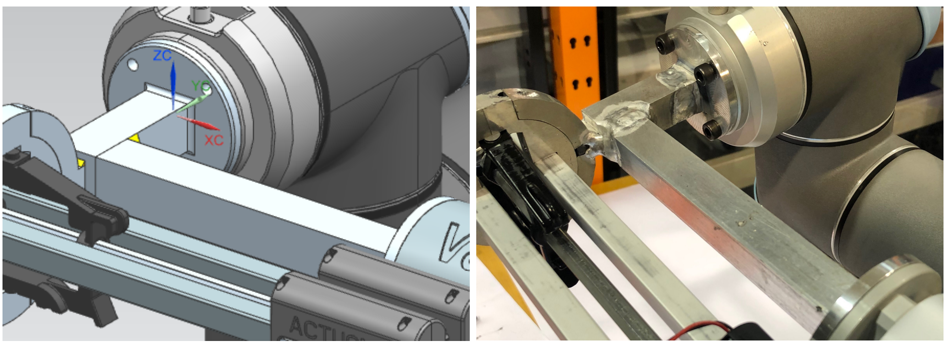
(a) CAD model of the welded design, with the extender mounted.

(b) Physical model of the welded design.

Figure 4.6: Solution to the rotation complications: Welded models of the dual tool.

The final dual tool mount design

Two pictures of the final design with the two assembly tools mounted, can be seen in Figure 4.7.



(a) CAD model of the final design.

(b) Physical model of the final design.

Figure 4.7: The finished dual tool mount.

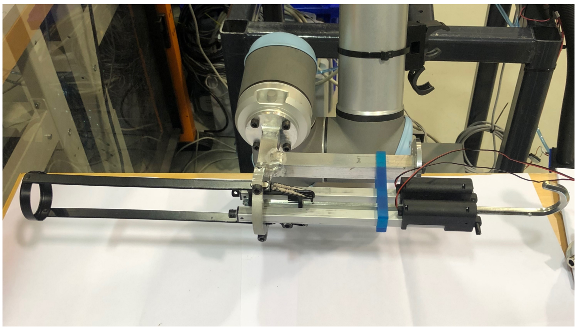


Figure 4.8: The sealant tool.

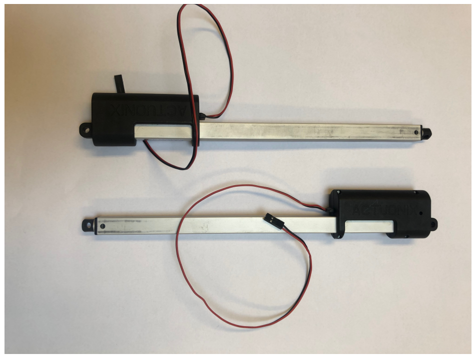
4.2 Developing the sealant tool

To perform the sealant application a sealant tool was needed. The robot lab at IPK did not have a sealant application tool. A brand new sealant tool was either way outside the budget scope or had too long delivery time. Therefore, a custom, automatic sealant application tool had to be developed before work on the automation of the assembly process could begin.

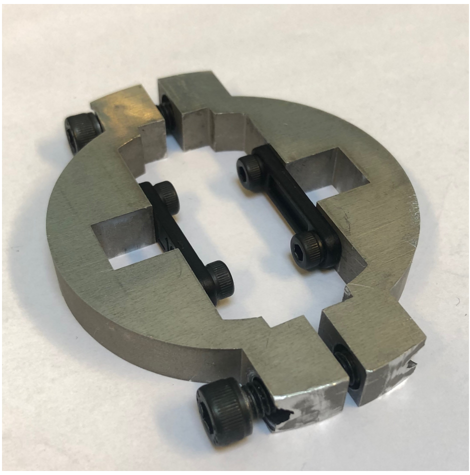
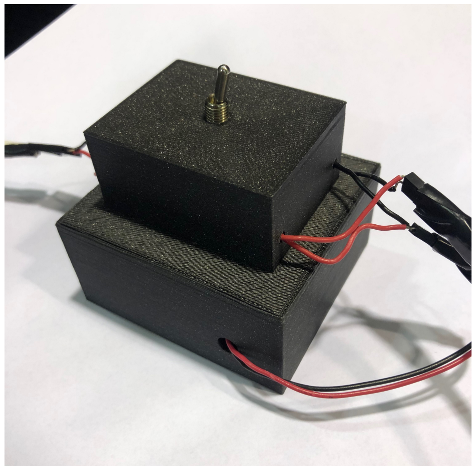
The automatic sealant tool was created with a modified, electric version of a standard, mechanical caulking gun. The purpose behind the tool was to automatically apply the sealant when the robot had its end-effector at the correct position. This meant that the tool would have to be able to respond to signals from the robot controller. To solve this, it was invested in two linear actuators. Their purpose was to physically push the plunger of the caulking gun on demand. The actuators were connected to the robot controllers DO ports. The linear actuators where ordered from the US, whereas the remaining parts were bought in a local hardware store or manufactured in the workshop at NTNU Valgrinda.

4.2.1 Sealant tool components

The main components of the sealant tool were a caulking gun, two linear actuators, some actuator supports, and a DPDT (Double-Pole Double-Throw) switch solution. The parts can be seen in Figure 4.9.

(a) A caulking gun. **(b)** Two linear actuators.

(c) One of the actuator supports. **(d)** A DPDT switch.

Figure 4.9: The main components of the sealant tool.

The mechanical parts

The caulking gun

The caulking gun was the skeleton of the sealant tool, all other parts of the tool was attached to it. The cartridge with sealant was loaded and kept in place in the front of the gun. The red caulking gun seen in Figure 4.9a is not the same as the one used in the sealant tool, which can be seen in Figure 4.10. However, it is the exact same model and gives a picture of how the black caulking gun looked before being modified. For the modified tool, the handles and spring/brake system were cut away to reduce friction on the plunger and minimize the total tool weight.

Figure 4.10: Modified caulking gun.

The actuator supports

The actuator supports were made to attach the linear actuators to the caulking gun and to mount the sealant tool to the dual tool mount. In total, five different supports were designed, four of them are shown in orange in Figure 4.11. The supports were mainly manufactured in aluminium or polymers to give high support at a low weight cost. The smallest pieces were machined in steel to be sufficiently strong. Because these pieces were so small the triple density of steel compared to aluminium was acceptable. The process leading to the final actuator supports design is described in section 4.2.2.

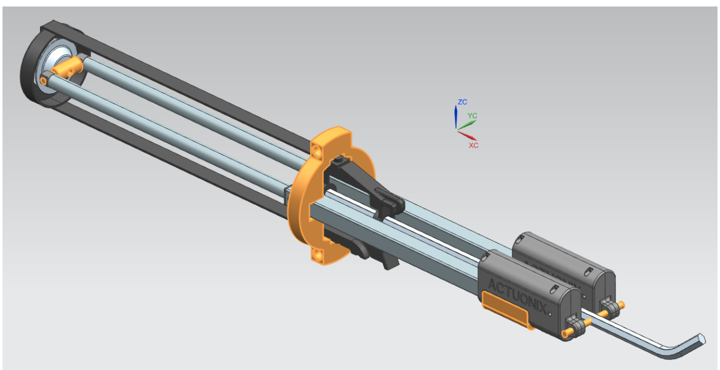


Figure 4.11: The actuator supports.

The mechatronical parts

The linear actuators

The linear actuators were of the electro-mechanical type. This meant that they consumed electrical energy to create rotary motion in the motor, which was transformed into mechanical pushing force. The actuators were mounted to the caulking gun to push the plunger through the cartridge and squeeze out sealant. The actuators came with different gear and stroke options. For the different gear options, higher power came at the cost of lower speed. In this case, power was more important than speed and the highest possible gear ratio of 256:1 was chosen. This ratio resulted in a maximum force of 300 N per actuator [7].

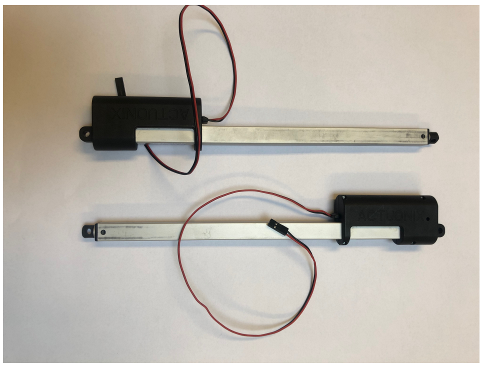


Figure 4.12: The linear actuators.

Stroke length was also an important factor, as the actuators had to be able to push out as much sealant as possible. The distance between the end of the caulking gun and the start position of the actuators was 220 mm, so the longest available stroke length of 200 mm was chosen. Each actuator was internally powered by a 12 V DC motor connected to the digital output ports of the robot controller. To effectively control the actuators a DPDT switch system was designed. This design is described in section 4.2.3.

To control both actuators simultaneously while maintaining the same voltage over both of them, the actuators were connected in a parallel circuit. It was important to have the same voltage as the voltage controlled the speed of the actuators.

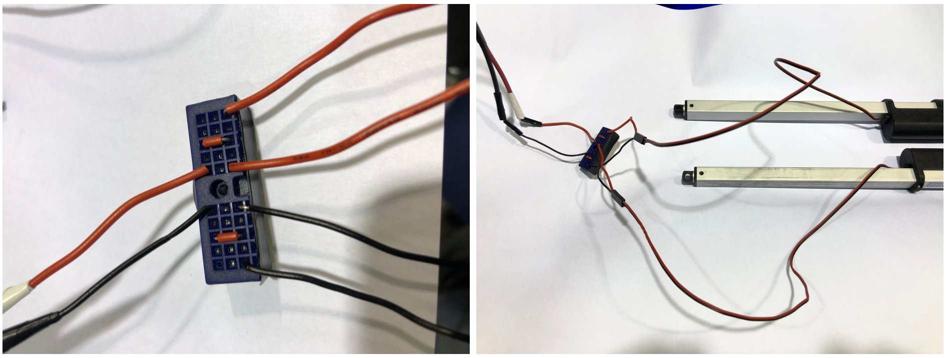


Figure 4.13: Parallel connection of the two linear actuators.

The DPDT switch

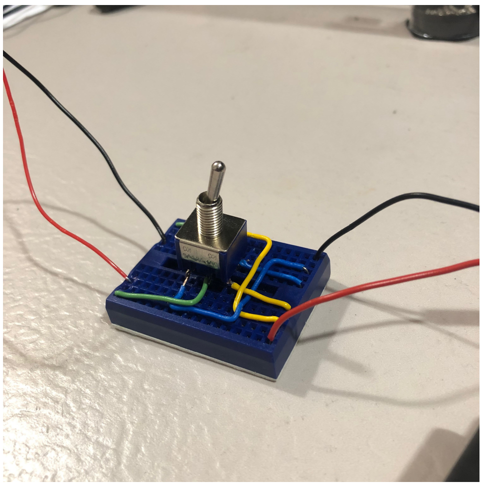


Figure 4.14: The DPDT switch.

The switch itself was mechanical, however, in order to make the process autonomous, the switch was used in an electrical system connected to the robot controller and the linear actuators. The end result of the DPDT switch system is therefore considered mechatronical. The total system is described further below in 4.2.3.

4.2.2 The mechanical setup:A modified caulking gun with supports

Even though the components of the sealing gun are quite simple, the final design becomes somewhat complex. As mentioned, the caulking gun had to be heavily modified and several actuator supports were needed. This part of the thesis describes how the mechanical design developed throughout the thesis and how design challenges were handled. It involves the modification of the caulking gun and the manufacturing of the actuator supports.

The caulking gun and the ring support

To begin with, the sealant tool design was fairly simple. It consisted of the caulking gun and a ring support. The caulking gun had its spring and brake systems and handles cut off as shown in Figure 4.10. Otherwise, the caulking gun was used as normal – the linear actuators were used to help push the plunger rod, with the plunger head attached onto the rod.

The ring support can be seen in Figure 4.9c, with actuator tracks and fastening braces screwed onto it. The tracks with the braces created two 12 mm squares just big enough for the actuators to fit. The purpose of the ring support was not only to hold both the linear actuators, but also to anchor the sealant tool to the dual tool mount. The ring was cut in two along the middle, to facilitate the mounting and dismounting of the sealant tool from the dual tool mount. An early concept CAD model of the ring support mounted with the sealant tool, which still has its spring/brake parts intact, can be seen in Figure 4.15.

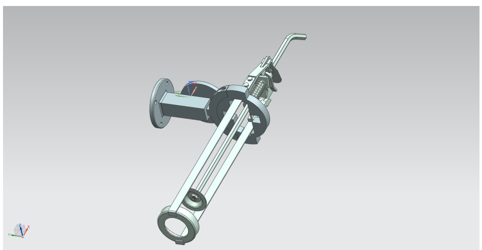
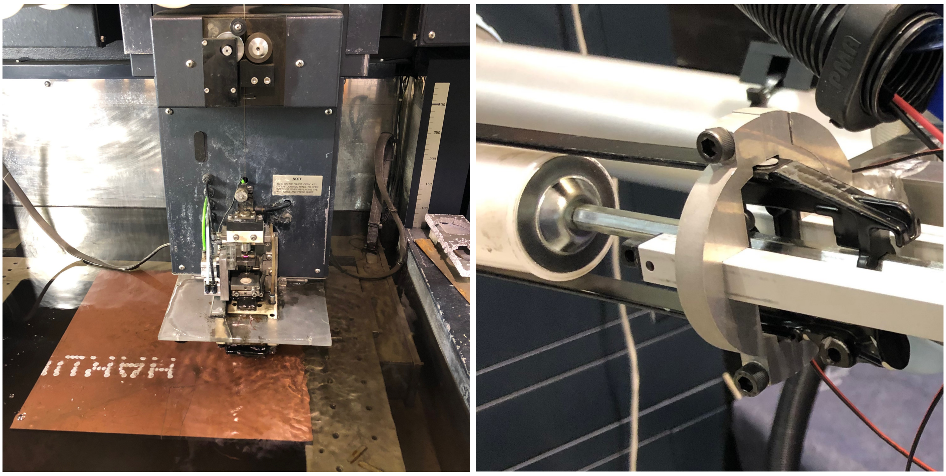


Figure 4.15: Early concept model of the sealant gun mounted to the dual tool mount.

The ring support was manufactured in aluminium using wire Electrical Discharge Machining (wire EDM), which uses electrical current to cut through the metal. Wire EDM is efficient for producing complex part geometries that are difficult to machine with other methods, and worked great for cutting the actuator tracks with a precision down to the millimeter. The wire EDM cutting process can be seen in Figure 4.16a, whereas the finished part on the sealant tool can be seen in Figure 4.16b.

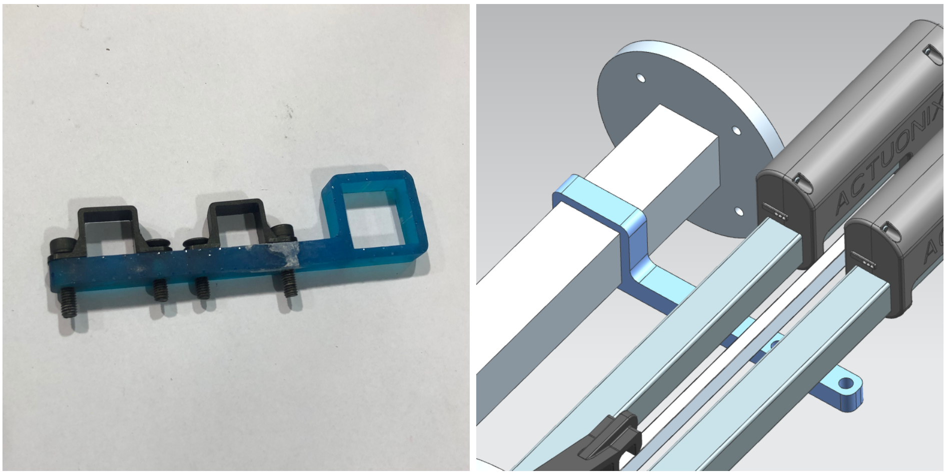


(a) The wire EDM process. **(b)** The ring support with actuators mounted.

Figure 4.16: The ring support during manufacturing and finished design with actuators mounted.

The polymer support

The actuators were only attached in their front to the ring support. To further restrict movement, a polymer support was designed to fasten the actuators at the other end. One of the biggest challenges of designing the polymer support was that the plunger rod had to be able to move between the actuators, whereas the actuators were standing still. The polymer support solution seen in Figure 4.17 was developed.



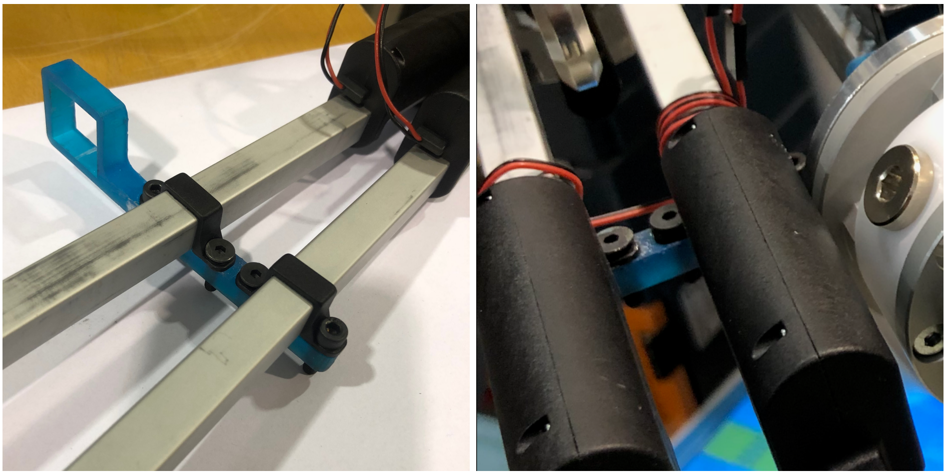
(a) Polymer support with actuator brackets. **(b)** CAD model of the polymer support.

Figure 4.17: The polymer support.

The polymer support was 3D printed in a Formlab Form 2 printer using inverted stereolithography. Stereolithography is a technology used for creating production parts in a layer by layer fashion using photochemical processes by which light causes chemical monomers to link together to form polymers [8]. Inverted means that the part was built upside-down. The photopolymer resin used was produced by Formlab and called Tough v5.

After being printed, the part was washed in isopropyl alcohol to remove excess liquid resin, and cured with 405 nm light at 60 °C to improve its mechanical properties. The properties improve because of two reasons. Firstly, the exposure to light triggers the formation of additional chemical bonds in the printed part, making the material stronger and stiffer. Secondly, heating the curing chamber to 60 °C accelerates the process and enables even more completed bond formations [9].

The design took advantage of using the hardware that came with the actuators – i.e. the black braces and brackets – and using the dual tool mount as the foundation for the polymer support. The polymer support with actuators attached, threaded onto the dual tool mount can be seen in Figure 4.18.



(a) With the actuators mounted by two fastening props.

(b) Just barely seen holding the two actuators, threaded onto the dual tool mount.

Figure 4.18: The polymer support.

Design complications

This first design was not enough to support the actuators. The supports functioned in holding the actuators sturdy, but not tightly. The actuators were not sufficiently fastened against displacement along their axis of motion (the x-axis, marked with a red arrow in Figure 4.11). During testing of the design, when the actuators met the start of the cartridge and began to push out sealant, the resistance in the cartridge became too high due to the sealant's high viscosity resulting in both actuators sliding backwards instead of pushing the sealant forward. It was later discovered that the sealant had solidified, which meant that the actuators tried to compress a rigid block of sealant

Additionally, it was difficult to make the polymer support low enough in its middle to not crash with the plunger rod, while still high enough to support the actuators. Note that in Figures 4.17 and 4.18, the screws used in the middle were flat headed, yet the plunger rod barely managed to move above the support, adding pressure on, and slightly bending, the polymer support downwards. The material was not as though as expected, and the screws cut through the threads in the polymer breaking, effectively loosening the brackets. Eventually, the polymer support broke entirely, already before testing. These two major flaws initiated a complete redesign, focusing on back support.

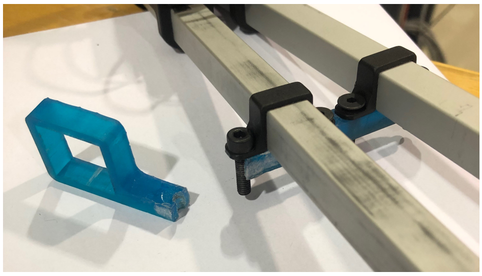


Figure 4.19: Broken polymer support.

Modification of the caulking gun and development of the silver steel support

The redesign modified the caulking gun. Limited mounting possibilities for back support of the actuators lead to a repositioning of the plunger rod. The plunger head was separated from the rod and the rod was pulled to the back of the caulking gun, where it was used statically as a support structure. A hole, aligned with the end-holes of the actuators, was drilled through the plunger rod and a silver steel rod was pinned through all three holes. The silver steel support mounted on the sealant tool can be seen in Figure 4.20. Its ends were covered in electrical tape to keep it in place.

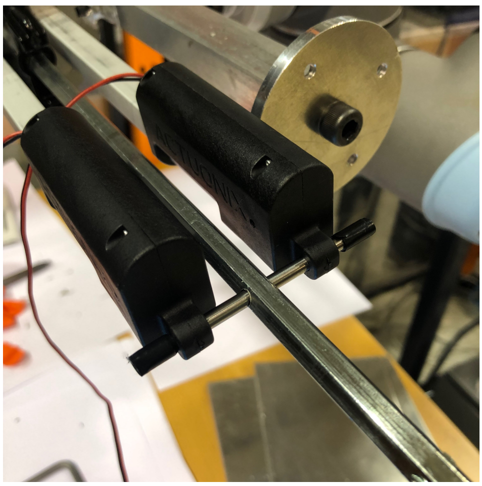


Figure 4.20: The silver steel actuator support.

The plunger head support

A support for the separated plunger head was developed. The plunger head support was designed to be held between the moving ends of the actuators, and to attach and aim the plunger head. It was drilled through the middle and threaded to fasten the plunger head with a screw. Both CAD models and pictures of the physical part can be seen in Figure 4.21.

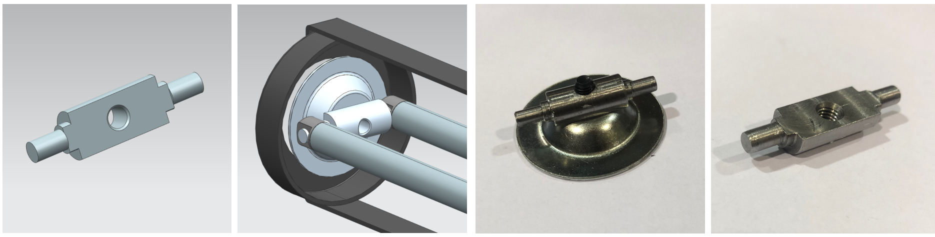
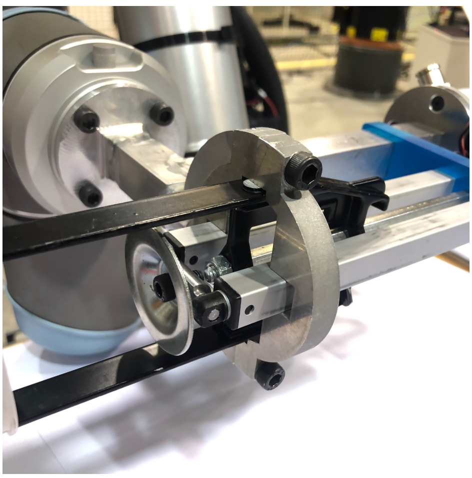
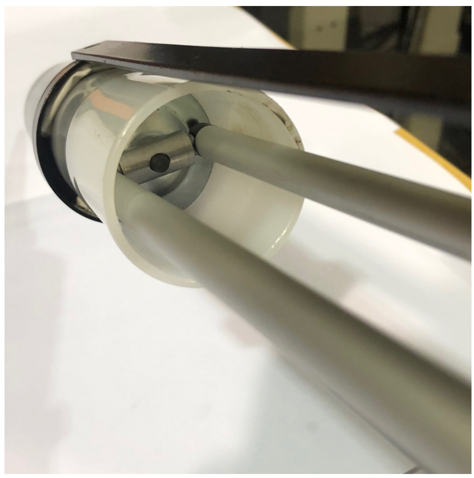


Figure 4.21: CAD models of and the actual plunger head support.

The final design of the plunger head support was lathed in a lathe turning machine. The tiny part, seen to the right of Figure 4.21, was only 38 millimeters long. In Figure 4.22a, the plunger head support attached to the sealant tool, as well as the new position of the plunger rod at the back of the caulking gun, can be seen. In Figure 4.22b the plunger head support can be seen pushing sealant inside the sealant cartridge.



(a) The plunger head support and new plunger rod position.

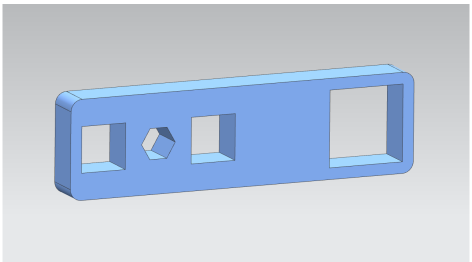


(b) The plunger head support inside the sealant cartridge.

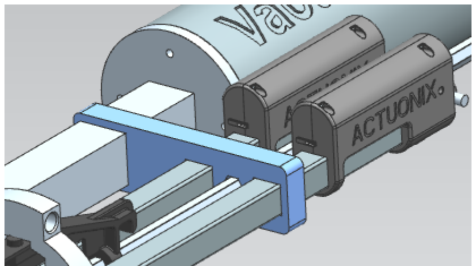
Figure 4.22: Plunger head support on the sealant tool.

The new polymer support

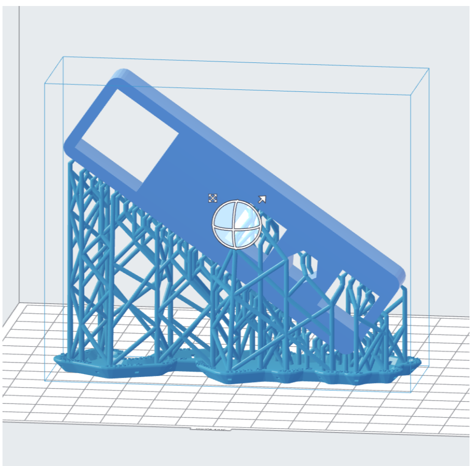
The first polymer support broke during testing. To reinforce the back of the plunger rod and actuators, another polymer support was additively manufactured. Again, the Though v5 material was used with the inverted stereolithography method. The first polymer support warped when printed horizontally, causing a small deformation. Because the polymer support had to be precise down to one tenth of a millimeter (to accurately fit the actuators), the print was done vertically this time. Consisting of 802 layers, the printing time spanned five hours and 25 minutes. After printing, the polymer support was washed and cured on 60 °C for one hour. The development and manufacturing of the new polymer support is summarized in Figure 4.23.



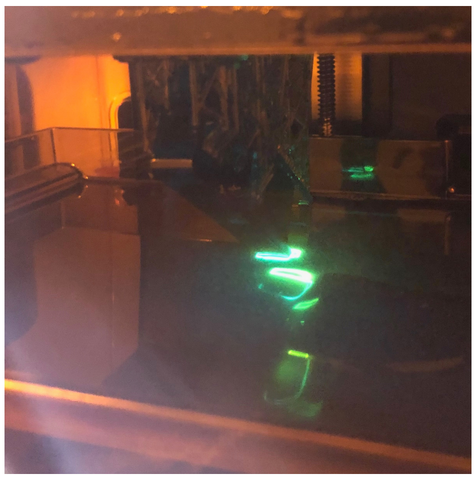
(a) CAD model.



(b) CAD model on complete tool.

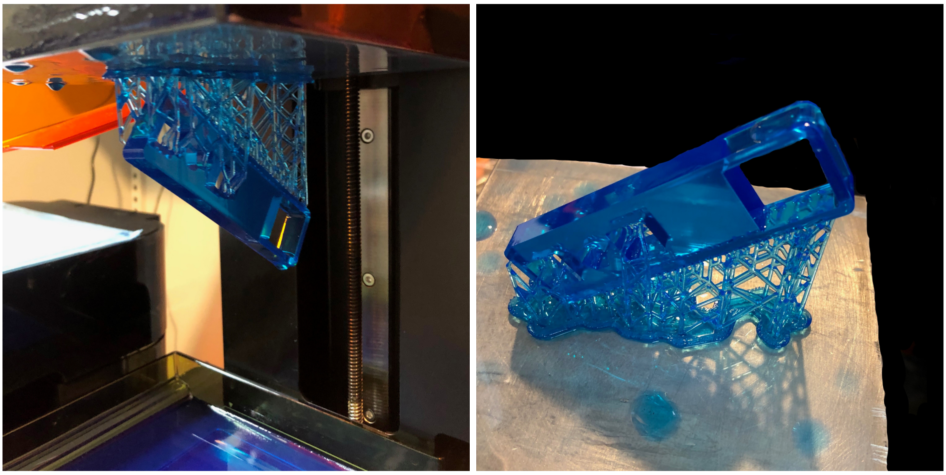


(c) 3D print plan.

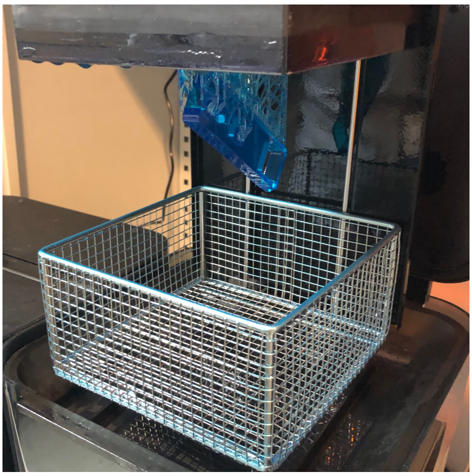


(d) During print.

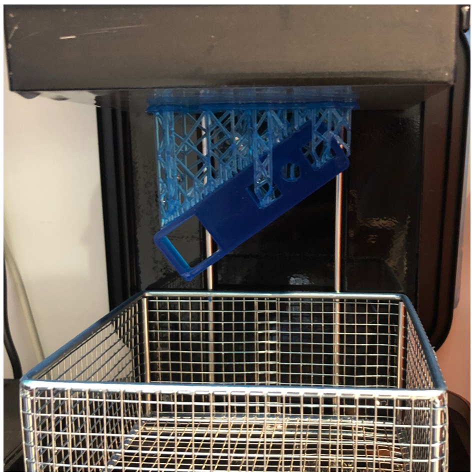
Figure 4.23: Creating the polymer support – Vol 1.



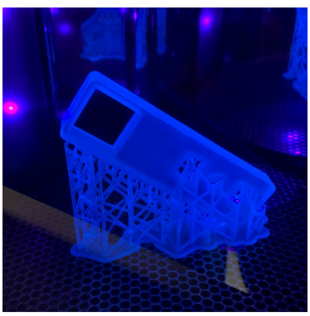
(e) After print.



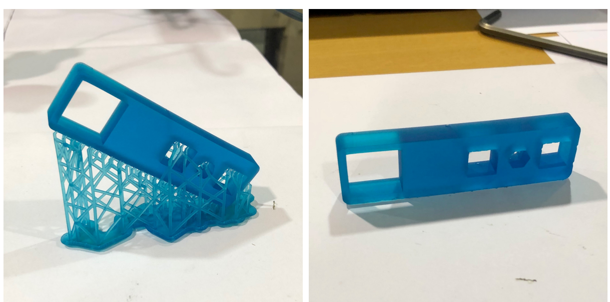
(f) Before wash in isopropyl alcohol.



(g) After wash – removed uncured resin.

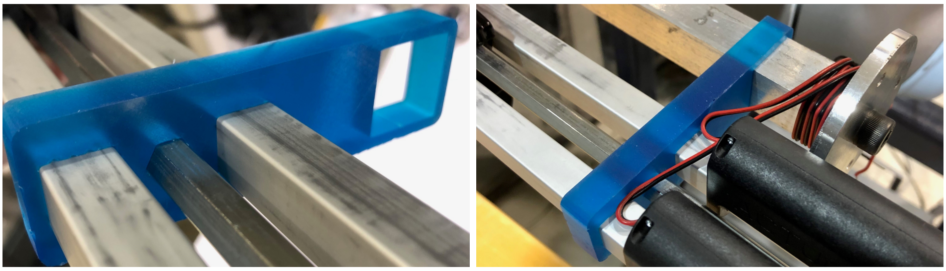


(h) Cured in 405 nm light.



(i) Finished part with and without supporting structure.

Figure 4.23: Creating the polymer support – Vol 2.



(j) With actuators and plunger rod mounted. **(k)** Mounted on the dual tool mount.

Figure 4.23: Creating the polymer support – Vol 3.

The final mechanical setup

After the caulking gun was modified and the required actuator supports were manufactured, the mechanical setup of the sealant tool was complete – including the CAD assembly of the entire tool (with the vacuum tool part simplified). The final CAD assembly was implemented in the offline programming solution explained later, in Chapter 5.1. The final mechanical setup is shown in Figure 4.24.

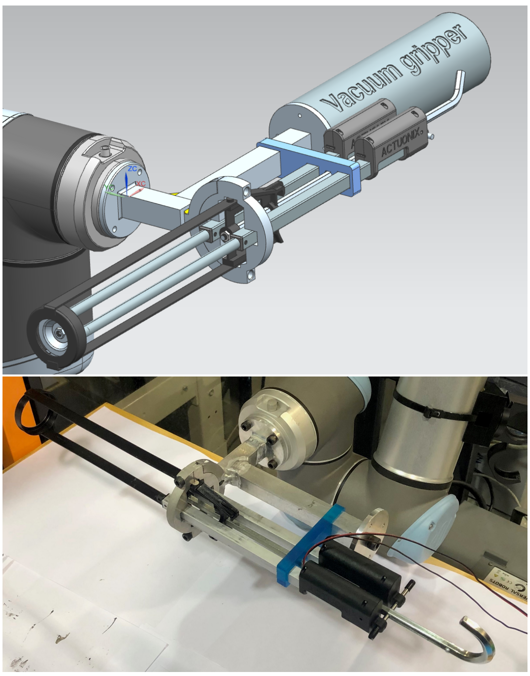


Figure 4.24: The entire CAD assembly and the final mechanical setup of the sealant tool.

4.2.3 The mechatronical setup: A DPDT switch system

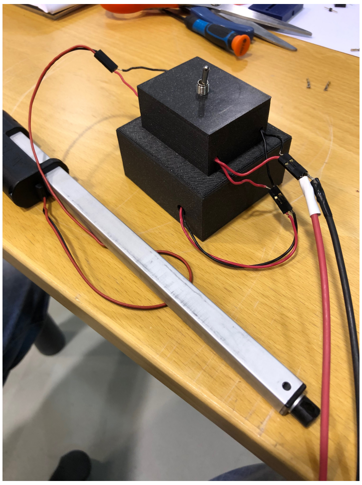
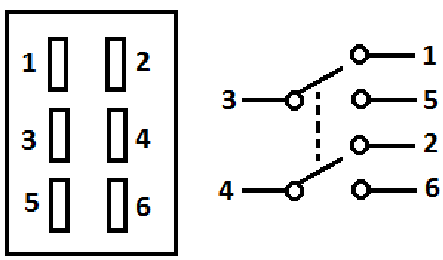


Figure 4.25: The DPDT switch system.

A solution to both extend and retract the linear actuators was developed for automatic control of the sealant tool. This can be accomplished several ways, i.e. with a DPDT switch or relay, or by using an H-Bridge. A DPDT switch solution was chosen.

The DPDT switch solution

A DPDT switch solution was used to alternate the current flow and control the motion of the actuators. A picture of the type of DPDT switch used in this thesis, can be seen in Figure 4.26, along with two possible ways of depicting the wiring diagram of the switch.



(a) The physical switch. **(b)** Wire diagrams of the switch.

Figure 4.26: The DPDT switch with wire diagrams.

An example of how a DPDT switch can be used to reverse the current flow through the DC motor of a linear actuator, can be seen in Figure 4.27.

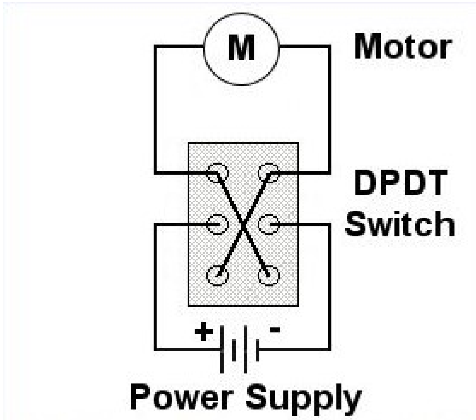


Figure 4.27: DPDT switch used to reverse a DC motor.

The figure above was used to design the DPDT switch solution, where a linear actuator was connected to a battery pack of 12V for initial testing. The design can be seen in four pictures in Figure 4.28. The circuit diagrams can be seen in Figure 4.29, showing the two states of the DPDT switch causing the actuator to extend or retract. Note that this is the same connection as in Figure 4.27, but with the other type of wiring diagram used for the switch (as shown in the right part of Figure 4.26b).

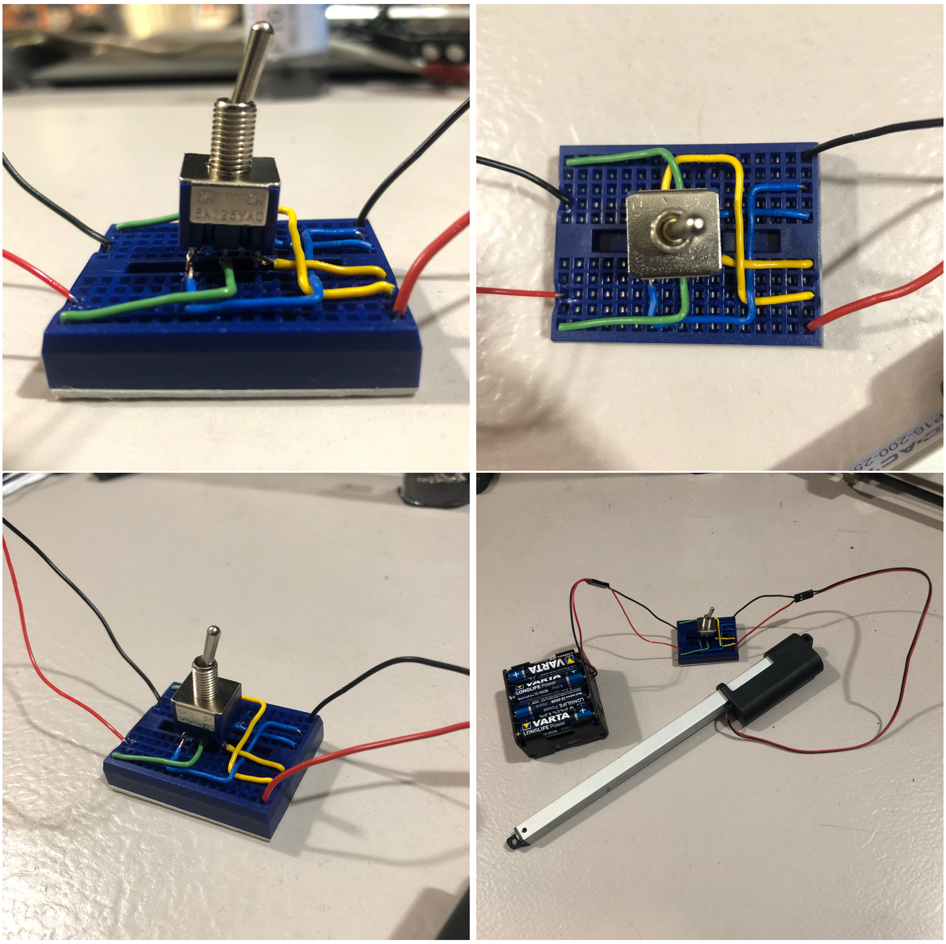


Figure 4.28: The first DPDT switch solution.

This connection enabled the robot to control the actuators to both extend or stand still, or to retract or stand still depending on the position of the DPDT switch. Note that this is a semi-automatic solution. To change from extend to retract, the switch must be flipped. However, while the robot program runs, the sealant tool only needs to extend to squeeze out sealant of the cartridge, or stand still to hold and wait for the robot to get into a new position for further sealant application. The only scenario when the actuators need to retract is when the sealant cartridge is empty and needs to be changed. Because of how the sealant had to be mixed prior to being loaded and applied, manual interaction was unavoidable. The solution thus served it's purpose autonomously when the DPDT switch was on extend mode – until a cartridge reload was necessary.

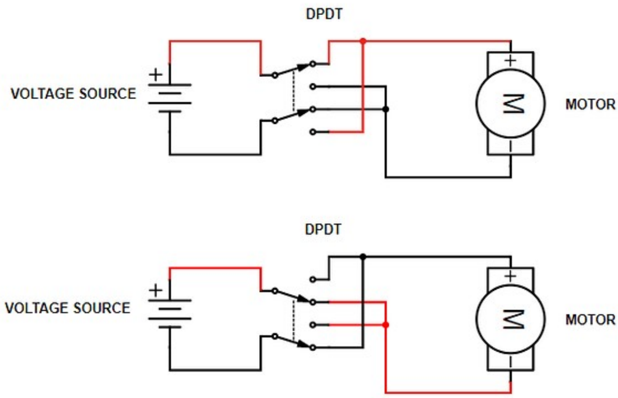


Figure 4.29: Circuit diagram of the DPDT switch in its two possible states. The top circuit shows the extend state, while the bottom shows the retract state.

Modified DPDT solution

The tool was not able to handle automatic cartridge reloads, so an option to retract the actuators when the robot was powered off was wanted. This solution would allow for manual reload and handle of the actuators during testing. To incorporate this feature in the previously explained DO solution in section 4.2.3, a multipurpose system was designed, which connected the DPDT switch system to the 12V battery pack from the initial testing, in combination with the robot controller's digital output voltage source.

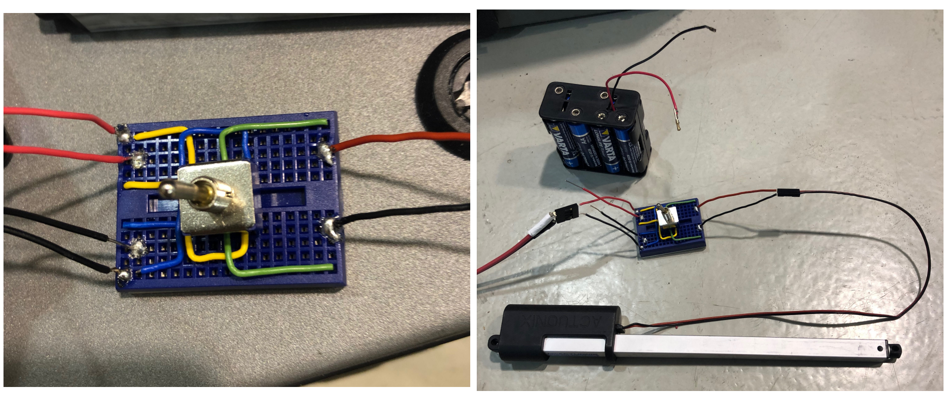
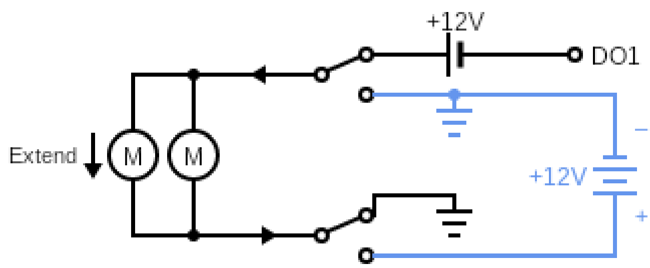
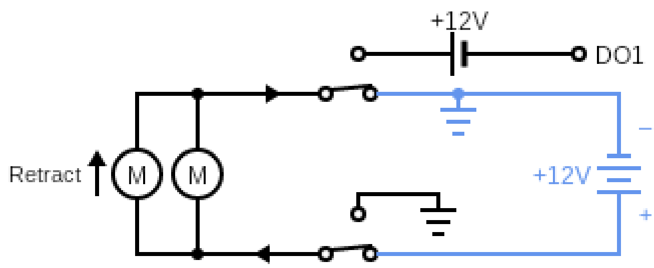


Figure 4.30: The modified DPDT switch solution. The two wires to the right go to the motors of the linear actuators, whereas the four wires to the left are connected to the two voltage sources; the robot controller and the battery pack.

In the new design the motors of the linear actuators are instead connected to the middle pins of the switch (pins 3 and 4 in Figure 4.26b). This effectively makes the switch change between the two different power sources instead to reverse the voltage over the actuators this way. The physical design can be seen in Figure 4.30, with the connections soldered in place. The circuit drawings can be seen in Figure 4.31. Switching the flip up in ”extend mode” (Figure 4.31a) makes the actuators draw power from the DO1 port of the robot controller. Setting the DO1 signal high extends the actuators, whereas setting the signal low controls the actuators to stand still. By reverse-connecting the battery to the actuators, the DPDT switch retracts the actuators whenever the switch is flipped down in ”retract mode” (Figure 4.31b).



(a) Extend mode.



(b) Retract mode.

Figure 4.31: Circuit diagrams of the multipurpose DPDT solution. The blue wires are the battery pack circuit.

Protective plastic cover

Finally, a plastic cover was designed and additively manufactured. The method used was Fused Deposition Modeling (FDM) with a material called Prusament Galaxy Black. The total printing process took five hours and 30 minutes. Pictures from the development process and the result can be seen in Figure 4.32. The purpose of the cover is to keep all the wires, the battery pack, and the switch packed together neatly, and to protect the entire switch system.

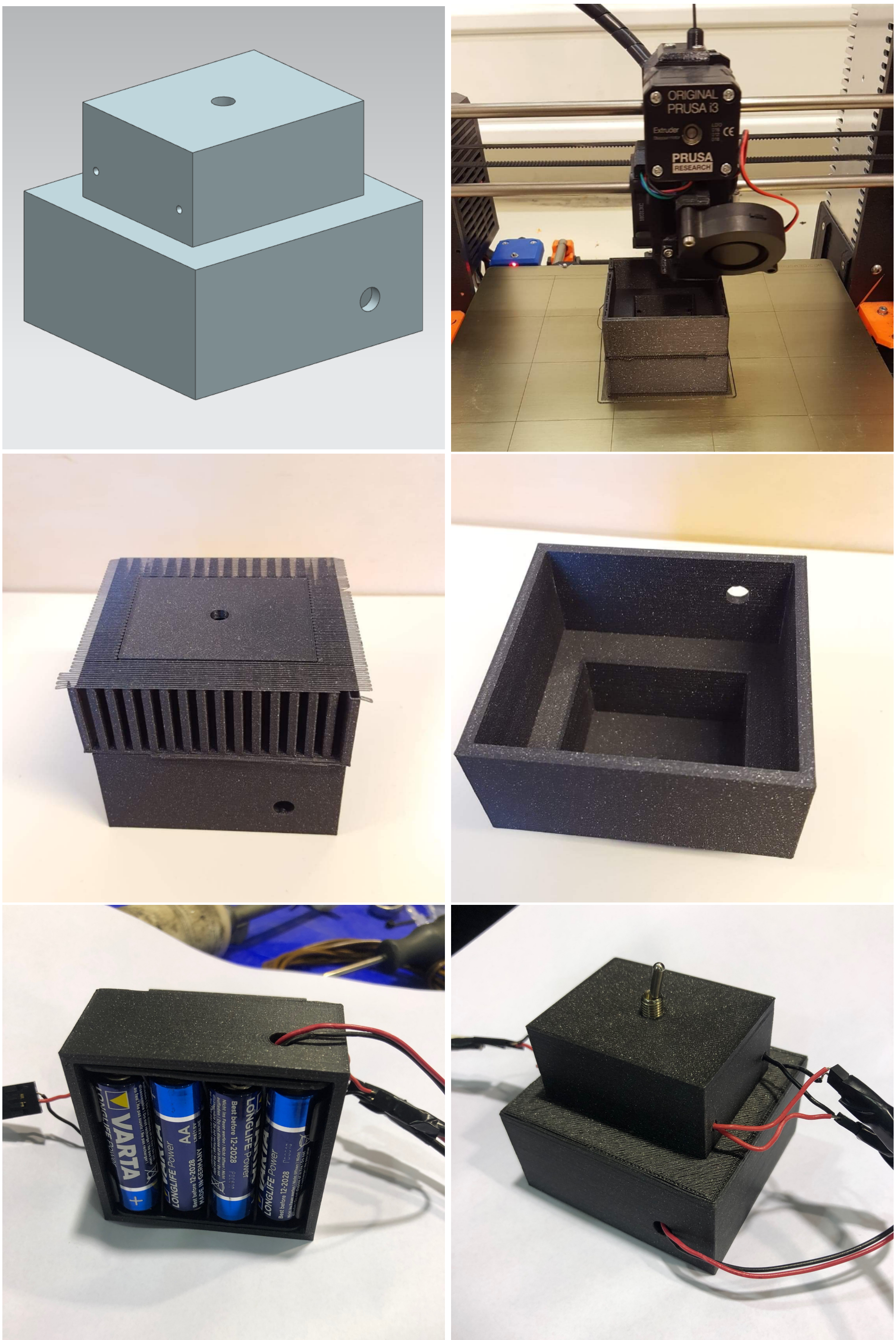


Figure 4.32: Development of the cover. Top: CAD model and printing. Middle: Results showing support structure and the inside. Bottom: Battery pack and switch neatly packed inside the cover.

Voltage converter

Finally, a voltage converter was installed. It reduced the voltage sent out from the control box to the actuators from 24V to 12V. The voltage converter setup can be seen in Figure 4.33, with closeups of the connection ports shown in Figure 4.34.

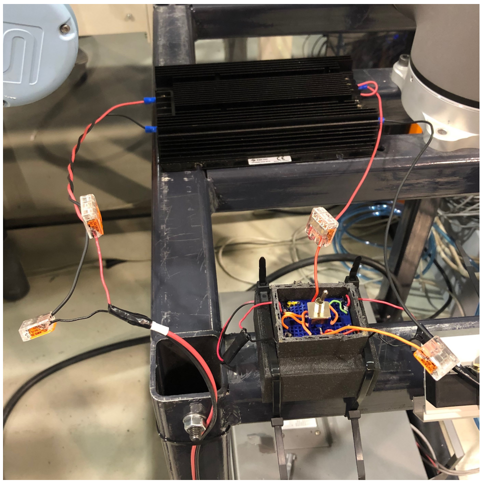
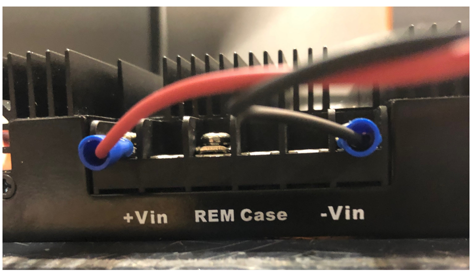
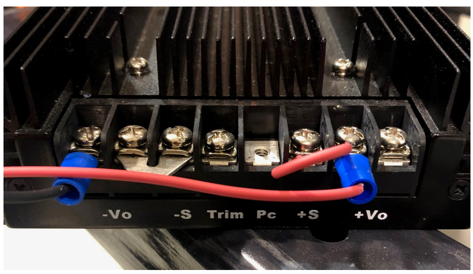


Figure 4.33: The voltage converter setup.



(a) The input to the converter: 24V.

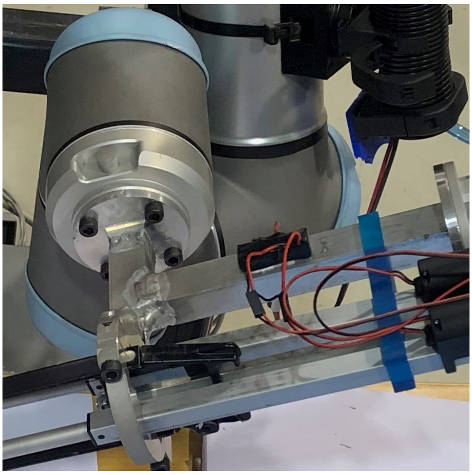


(b) The output from the converter: 12V.

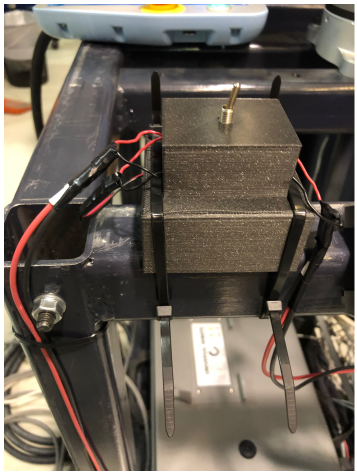
Figure 4.34: Closeups of the connection ports of the voltage converter.

The final mechatronical setup

To reduce the maximum weight mounted on the UR10, only the parallel circuit part of the DPDT switch system was mounted directly on the assembly tool, shown in Figure 4.35a. Therefore, the switch itself was ultimately mounted on the robot base structure (Figure 4.35b), to what would later become the overall control panel of the assembly process. The rest of the control panel consists of components from the vacuum tool, which is explained in the next section.



(a) Parallel circuit mounted on the assembly tool.



(b) DPDT switch mounted on the robot base structure.

Figure 4.35: The parallel circuit mounted on the assembly tool and the DPDT switch mounted on the robot base structure.

4.2.4 The final design of the sealant tool

The final design of the sealant tool with both the mechanical and the mechatronical setups complete, can be seen in Figure 4.36.

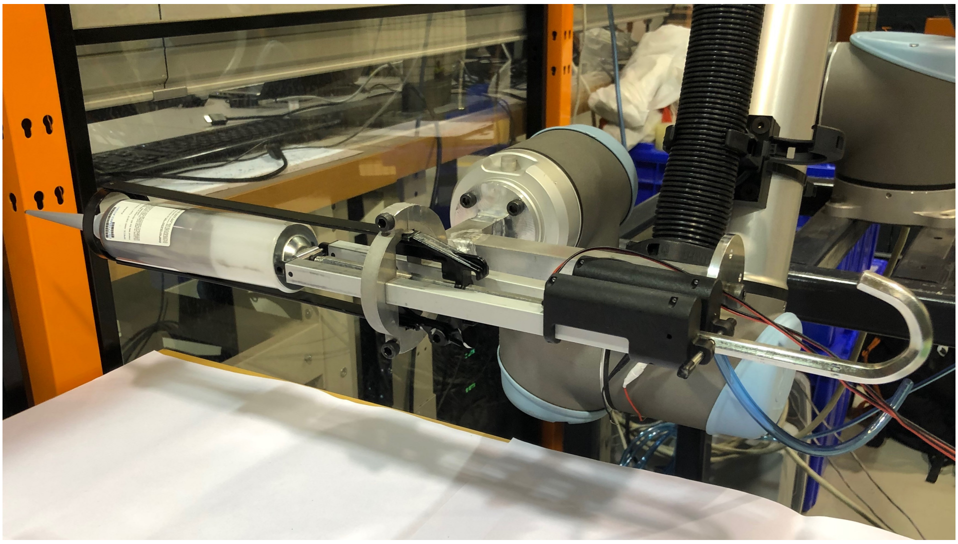


Figure 4.36: The final design of the sealant tool.

4.3 Setting up the vacuum tool

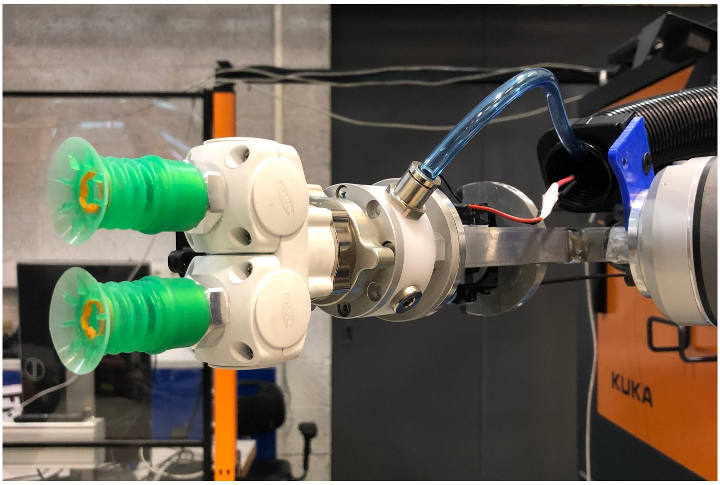


Figure 4.37: The vacuum tool.

A pneumatic vacuum gripper with suction cups is used to pick and place the aluminium parts of the assembly process. Suction cups were chosen because the assembly parts are mostly flat and do not have any distinct gripping possibilities, which a mechanical gripper with fingers would need. Additionally, a vacuum gripper is relatively cheap and the vacuum effect works well with the smooth, solid, metal surfaces of the aluminium parts. A disadvantage with a vacuum gripper is that compressed air supply is needed (or a vacuum pump), which contributes to extra energy costs. Also, the vacuum gripper might be sensitive to dusty or dirty environments [10].

4.3.1 Vacuum tool components

The vacuum tool used in this thesis consists of a pressure reduction valve, a solenoid valve with a cord connection to the robot controller, a vacuum generator based on the Venturi principle, and the vacuum gripper with suction cups, as described in section 3.3.2. In addition, the system is connected to NTNU's pressure supply through the output port in the robot lab.

The pressure reduction valve

Figure 4.38: The pressure reduction valve.

The pressure reduction valve is used to set a specific output pressure because the pressure from the supply port in the robot cell is higher than what is necessary. It is close to eight bar, and the system requires only two. Not regulating the pressure causes excess consumption of compressed air and results in waste of energy [11].

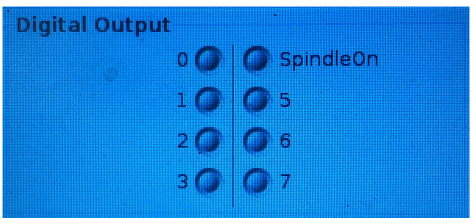
The solenoid valve

(a) The solenoid valve.

(b) Cord connection to the robot controller.

Figure 4.39: The two components of the solenoid valve.

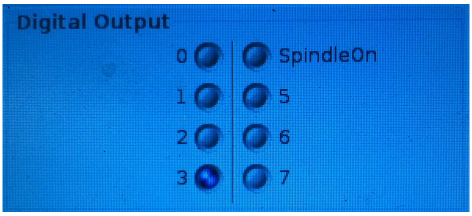
The solenoid valve is an electromechanical device, which can send an electric current through its solenoid to generate a magnetic field [12]. This magnetic field is used to operate the mechanism which regulates the opening or closing of the valve to control the flow of compressed air. The solenoid valve is thus used for turning the vacuum gripper on or off. The valve is connected to the robot controller and controlled through digital output signals.



(a) DO3 set low.

(b) Solenoid valve off.

Figure 4.40: Turning the vacuum gripper off using DO signals from the robot controller.



(a) DO3 set high.

(b) Solenoid valve on. Notice the small yellow diode to the right.

Figure 4.41: Turning the vacuum gripper on using DO signals from the robot controller.

The Venturi vacuum generator

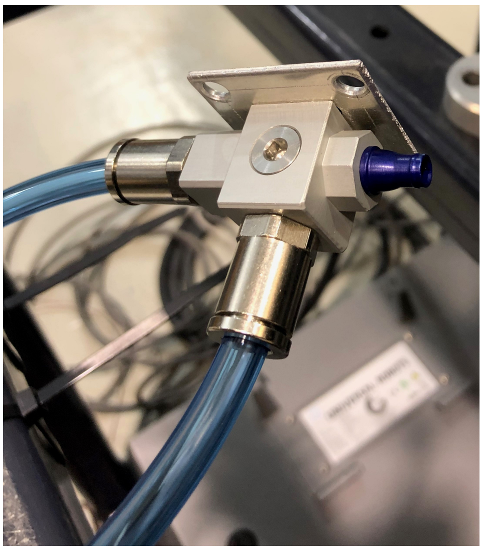


Figure 4.42: The Venturi vacuum generator.

The pneumatic vacuum generator used in this thesis is a vacuum ejector, which operates based on the Venturi principle explained in Chapter 3. The generator can be seen in Figure 4.42.

The vacuum gripper

(a) The vacuum gripper.

(b) Gripper components set.

Figure 4.43: The vacuum gripper and components set.

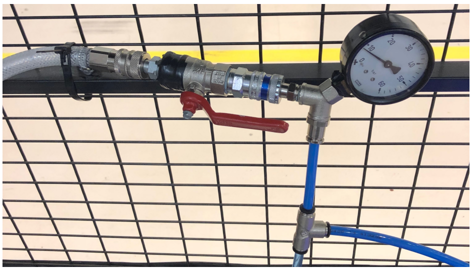
The robot lab provided a vacuum end-effector set. The gripper used in this thesis was built from this set of components made by Schmalz.

4.3.2 The final vacuum tool setup

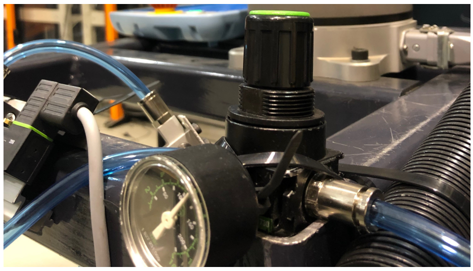
To sum up, the chronological flow of air to make the vacuum gripper able to pick and place the assembly parts is presented: The compressed air is supplied from the robot lab output port and flows to the pressure reduction valve – the pressure gets reduced – then enters the solenoid valve. If the solenoid valve is on, the compressed air flows into the vacuum generator and exits through its silencer. In the vacuum generator, because of the Venturi nozzle, the air from the vacuum gripper gets sucked out. This causes a vacuum in the gripper that allows the suction cups to attach and pick up workpieces. Pictures of the whole process in chronological order can be seen in Figure 4.44. When the robot is ready to place the workpieces, the air is shut off in the solenoid valve. When the solenoid valve is closed, no vacuum gets created in the vacuum generator, so air is no longer sucked out from the vacuum gripper. Instead, the gripper gets filled with air because of its lower air pressure. The suction cups releases the workpieces when the pressure difference is too low to hold onto the surface.

Robot cable management

Additionally, a cable management set is mounted on the UR10. The set consists of a black tube stretching from the control panel at the base (Figure 4.44e), to the end-effector at the tool flange (Figure 4.44g). The tube carries and protects both the pneumatic tube for the vacuum gripper and the electrical wires for the linear actuators on the sealant tool. A rotational joint for the cable tube is mounted on the upper arm of the UR10. The joint provides stability, while still ensuring flexibility, of the tube while the robot is moving (Figure 4.44f).

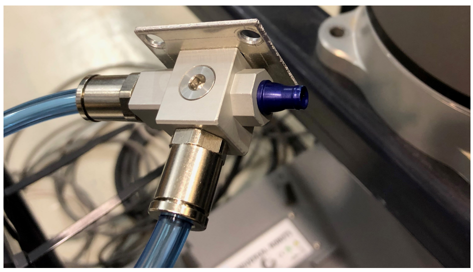


(a) Starts from the output port.

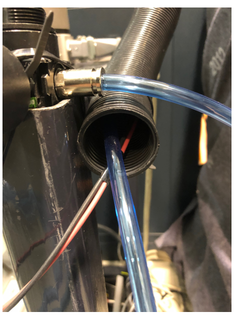


(b) Through the pressure release valve.

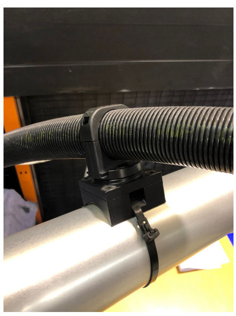
(c) On or off with the solenoid valve.



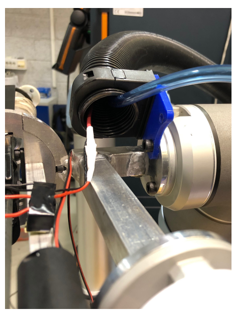
(d) Venturi nozzle vacuum generator.



(e) Support tube entrance.

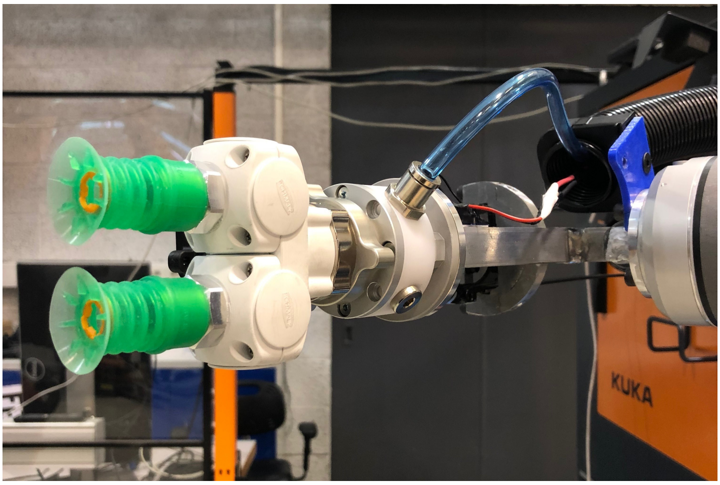


(f) Rotational joint.



(g) Support tube exit.

Figure 4.44: Chronological process of creating vacuum in the gripper – Vol 1.



(h) Vacuum gripper.

Figure 4.44: Chronological process of creating vacuum in the gripper – Vol 2.

4.4 The final assembly tool design

The final design of the assembly tools consist of a sealant tool and a vacuum tool, both mounted on a dual tool mount. The complete tool was mounted on the UR10. The finished assembly tool setup is shown in Figure 4.45.

The assembly process control panel

Similarly to the DPDT switch box, both the pressure reduction valve and the solenoid valve were mounted on the robot base structure. Together the three components completed the control panel of the assembly process, seen in Figure 4.46.

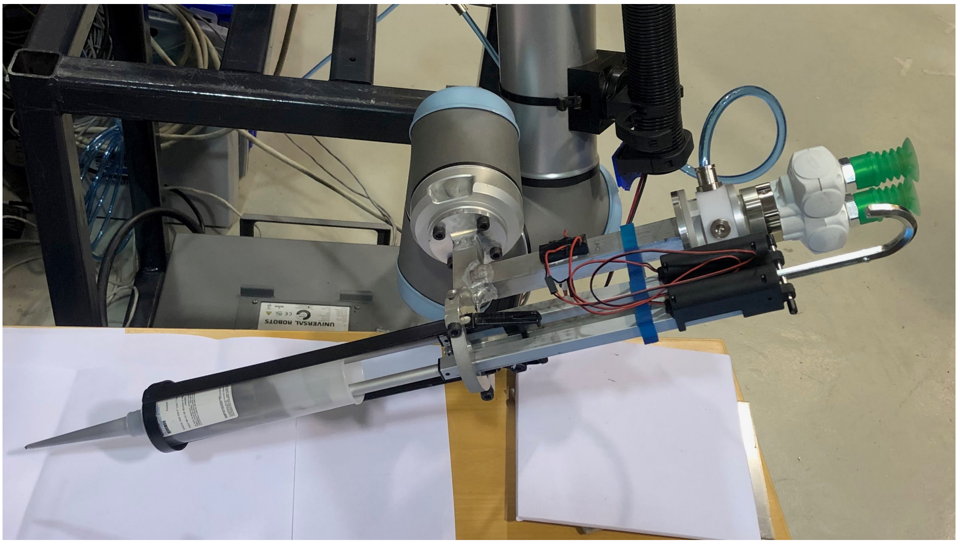


Figure 4.45: Finished assembly tool setup.

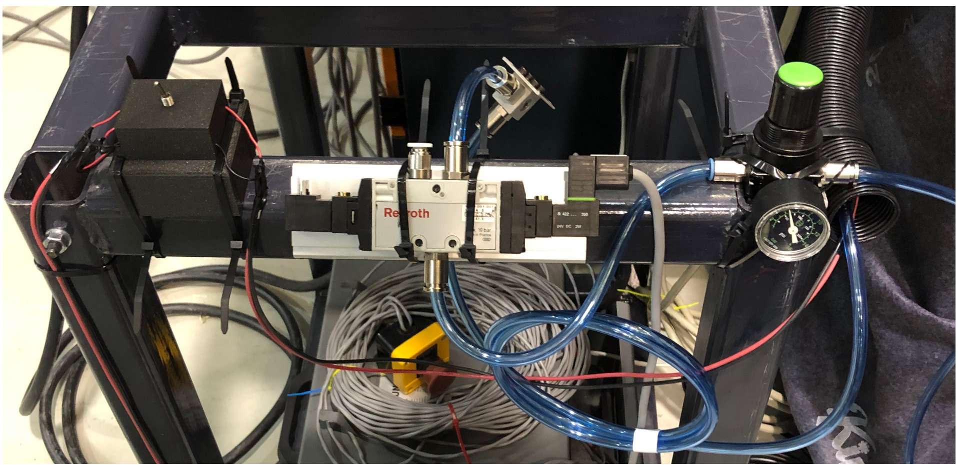


Figure 4.46: Control panel of the assembly process.

Chapter 5

Offline programming of the UR10 with Visual Components

A brief summary about OLP was given in Chapter 3.4. The use of OLP software was studied during the project thesis of last semester (autumn 2018) and the case of Visual Components was examined in detail. A natural continuation of the project thesis was to establish a connection between the offline programming (OLP) software Visual Components and the UR10 robot.

Successfully creating the connection could result in a very elegant solution of remotely controlling and monitoring the UR10 in real time. The robot could then be controlled without scripting any programming code for it, e.g. in Python urx. Instead, the robot programs would be created through jogging or creating paths in Visual Components, which is more intuitive and quicker than coding robot scripts. It is also an offline simulation solution which has plenty of advantages mentioned in Chapter 3.4.

The procedure on how this connection could be accomplished is explained in Chapter 5.2.2 and 5.4.

5.1 The motivation behind using VC to control the UR10

Having a remote control connection directly to the UR10 from Visual Components (VC) would be an elegant way of controlling the robot. Especially because of the possibility to program the robot without the actual robot being present. This was helpful to counter the limitation of the workshop's restricted availability. Only having access to the robot lab from 8:00 to 15:30 during the weekdays, was a huge obstacle. Therefore, being able to establish an OLP connection meant that robot trajectories could be created and tested at the office after closing hours, for later to be implemented to the physical robot when the workshop was open again.

On the downside, VC does require CAD models of everything involved in the assembly process to recreate the virtual environment in the simulation, i.e. parts, tools, the robot itself, and the components of the work cell. Although creating CAD models of everything can be very time consuming, there are several open source communities, which share CAD models of complete tools, support structures, and even robots – VC already contains more than 2000 models, including robots from the most popular industrial robot brands. This avoids having to reinvent the wheel and design all models from scratch – unless the parts are distinctively unique as in Chapters 2 and 4 with the aluminium parts and the assembly tools. However, when the parts are distinctively unique, CAD models of the parts have most likely been made to develop the parts after all.

The upside of implementing custom CAD models is that it makes the development of robot trajectories even more tailored. The tool center point of the robot can be moved from the tool flange of the robot to match the proper tool center point of the end-effector. Figure 5.1 shows the assembly tool from Chapter 4 imported and implemented to the virtual environment in VC. In effect, everything that can be done physically with the UR10 in the robot work cell can with the correct CAD implementations be simulated with the UR10 in VC.

5.2 Establishing connection between Visual Components and the UR10

5.2.1 An introduction to relevant computer science

Enabling the connection between VC and the UR10, involved setting up a remote control via TCP/IP hooked up to the robot controller data streams. The goal was to make a real-time client for controlling and monitoring the UR10 remotely with VC.

This section will explain some relevant terminologies from the field of computer science needed to understand this chapter. The section after that, describes the connection process, however, some complications were met, and troubleshooting, potential other solutions, and workarounds were performed. These unsuccessful attempts are also explained in this section. They are important to include in case any potential following students or others

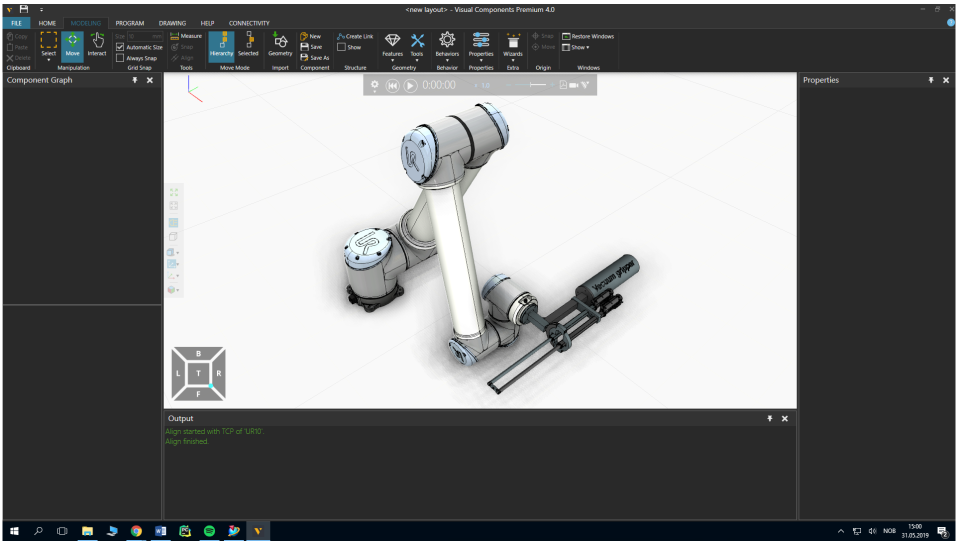


Figure 5.1: The CAD model of the assembly tool imported into the simulated UR10 virtual environment in VC. Note: The caulking gun is not included in this picture.

may want to continue the work of establishing the connection without doing the same mistakes.

Relevant terminologies:

- **TCP (Transmission Control Protocol):** is a protocol for one-to-one connections. It defines how VC and the UR10 can create channels of communication across the network. TCP manages how messages are compressed into smaller packets, transmitted, and recompiled at the destination address. In other words, TCP keeps track of data (packets) for efficient routing through the network.
- **IP (Internet Protocol):** is a protocol of how to handle the actual delivery of the data. It defines how to address and route each packet to make sure it reaches the right destination.
- **Internet protocol suite:** consists of TCP and IP – often referred to as TCP/IP. The suite was used as the communications protocols in the network connection between VC and the UR10. It is also the set of communications protocols used in the World Wide Web [13].
- **TCP/IP socket connection:** is associated with a specific socket address, namely the IP address and a port number for the local node (VC). There is also a corresponding

socket address at the foreign node (UR10), which itself has an associated socket, used by the foreign process [14].

- **RTDE (Real-Time Data Exchange):** The RTDE interface provides a way to synchronize VC with the UR10's robot controller over a TCP/IP socket connection, without breaking any real-time properties of the UR controller. This functionality is needed for interacting with drivers (e.g. Ethernet/IP), manipulating robot I/O and plotting robot status (e.g. robot trajectories) [15].

5.2.2 The connection process

In this subsection, a description of the connection process is presented. This part of the thesis shows what has been done and is essential for potential following students or others who want to replicate the connection process. The description is written as a step-by-step guide with pictures in order to communicate the information as intuitively as possible.

Prerequisites

The connection setup requires a computer with VC, a UR10 robot, and an Ethernet cable to connect the two. After setup, the computer and the robot both need their own static IP-address configured in the same subnet mask to be able to communicate with each other. The configurations of the computer and the robot can be seen in Table 5.1. The robot can then be pinged from a command prompt of the computer to assure that the computer and the robot are successfully connected (see Figure 5.2).

Table 5.1: IP and Subnet of the PC and the UR10.

	PC	UR10
IP address	192.168.0.215	192.168.0.101
Subnet	255.255.255.0	255.255.255.0

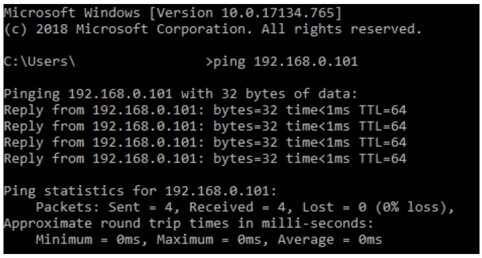


Figure 5.2: Ping command sent from the computer, verifying a connection.

Before starting, the ”Connectivity” feature in VC needs to be enabled. This is done by enabling it in the options menu of VC, under ”Add On”, as seen in Figure 5.3. A restart of the program is necessary after activating this feature. After restart the Connectivity tab will show up next to the Help tab, as seen in Figure 5.4, in the top, right corner.

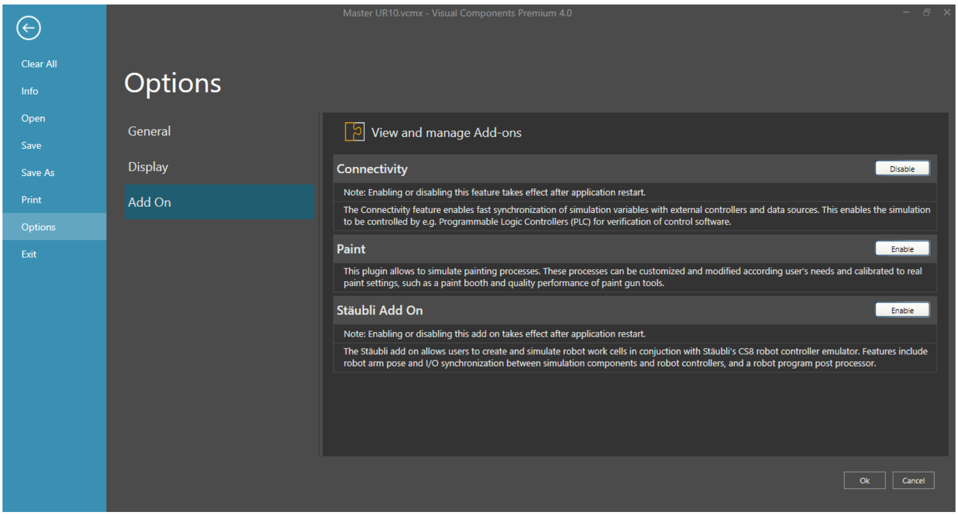


Figure 5.3: Enabling the Connectivity feature in VC.

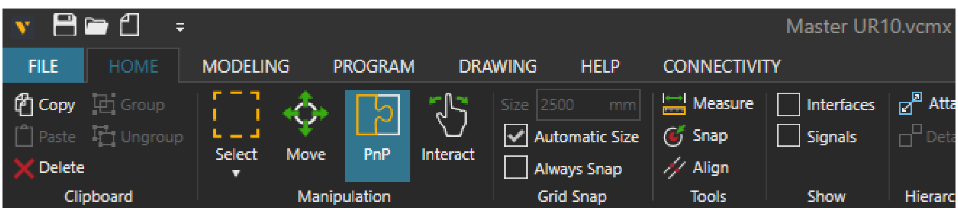
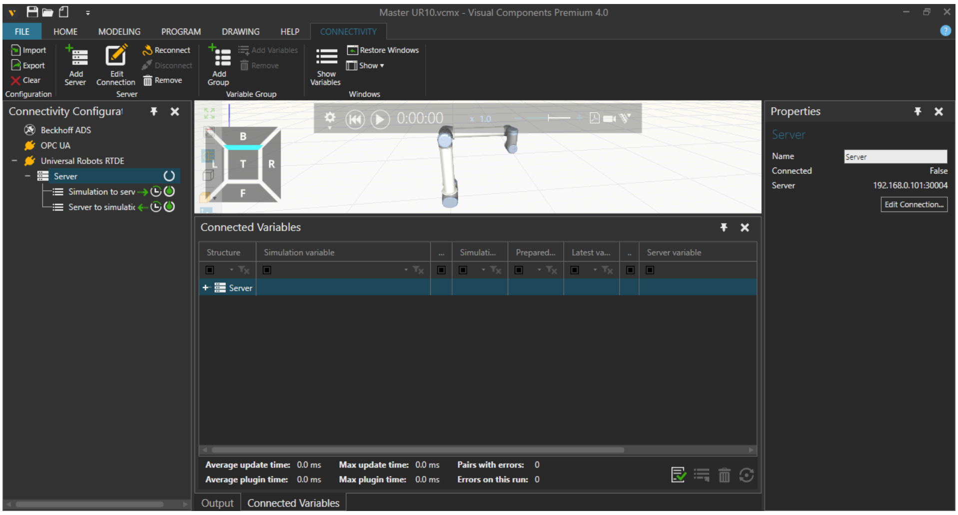


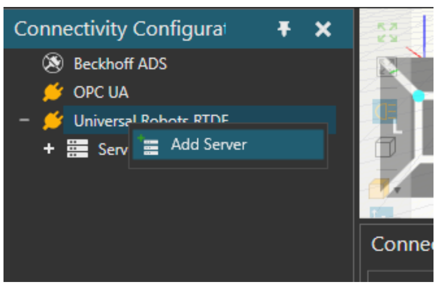
Figure 5.4: The Connectivity tab shows up after enabling it and restarting VC.

Step-by-step guide:

1. Power on, start, and initialize the UR10.
2. Open up VC and add a UR10 to the 3D world. To avoid confusion, this simulated UR10 in VC will from now on be referred to as UR10S, whereas the physical UR10 continues to be called UR10.
3. Under the Connectivity tab, select the UR10S, and add a server for RTDE connection, as shown in Figure 5.5b.



(a) The Connectivity tab with a UR10S in the 3D workspace.



(b) Add server for RTDE connection.

Figure 5.5: Adding an RTDE server to the UR10S under the Connectivity tab.

4. To activate the server connection, find the IP address of the UR10 robot controller and the RTDE port number.

(a) The IP can be found by checking the "About" tab in the PolyScope controller (see Figure 5.6) or checking the Network Connections folder in the control panel of the computer and looking at properties of the Ethernet connection under TCP/IPv4.

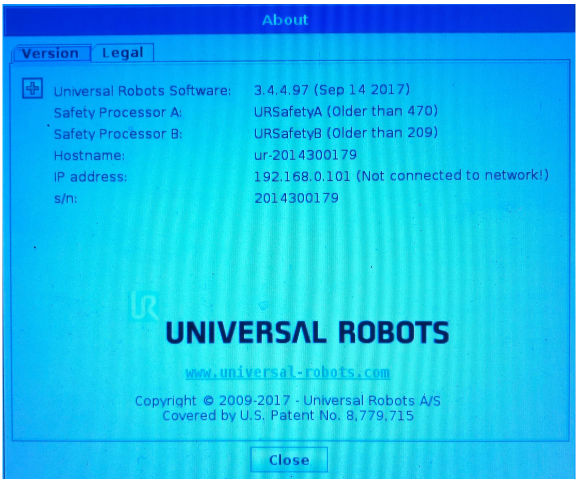


Figure 5.6: About tab in PolyScope.

(b) For the Universal Robots CB3-series the RTDE port is by default available on port number 30004.

5. When the IP address and the RTDE port are found, test and apply the connection between VC and the UR10, as seen in Figure 5.7.

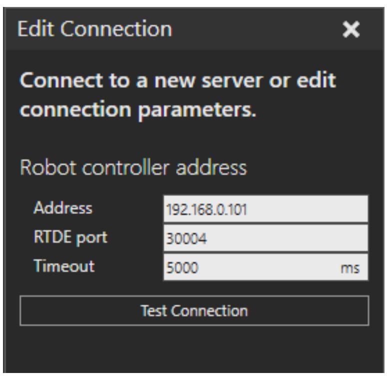


Figure 5.7: Testing and applying the IP and the RTDE port in VC.

Linking simulation and server joint variables

Following the step-by-step guide, an RTDE socket connection was successfully established over TCP/IP. The connection enabled the communication between VC and the UR10. To control the parameters of the UR10, the simulation variables of the UR10S had to be linked to the joint variables of the UR10. This was done in VC under the Connectivity tab after the RTDE server was added to the UR10S:

1. Right click the ”Server to simulation” and select ”Add Variables”. The ”Create Variable Pairs” panel should show up. Check the ”Signals” box and uncheck all others.
2. On the left side: Expand UR10 ⟶ RTDEInterface ⟶ Choose ”RTDEJoints”, which is a string signal.
3. On the right side: Expand Output ⟶ Joint Variables ⟶ Choose ”target_q”, which is a string vector containing all joints of the robot.
4. Click ”Pair Selected”. The process is shown in Figure 5.8.

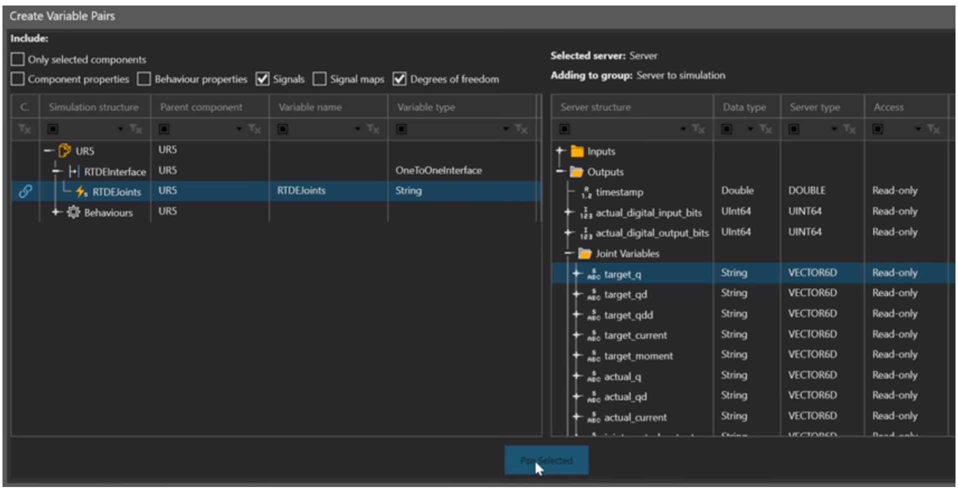


Figure 5.8: Creating the variable pairs.

5.2.3 Connectivity issues

The variables were paired, however hooking up to the robot controller data streams was harder than expected. All further work failed in setting up a remote control of the UR10. It seemed like the connection only went one way. The UR10S in VC would mimic the movements of the UR10, but the UR10 would not register movements done by the UR10S.

The goal was to establish two-way communication, or even just one-way communication, but the other way around, with VC writing positions for the UR10 to read. To figure out what the error could be, a troubleshoot was performed on the system. The reason for the mistake was found later and is presented in Chapter 6.2.

Troubleshooting and socket testing

The UR10 Client-Server was tested by sending script commands via a TCP socket connection from the host (PC with VC) to the UR10. The program SocketTest was used [16] along with this guide: [17].

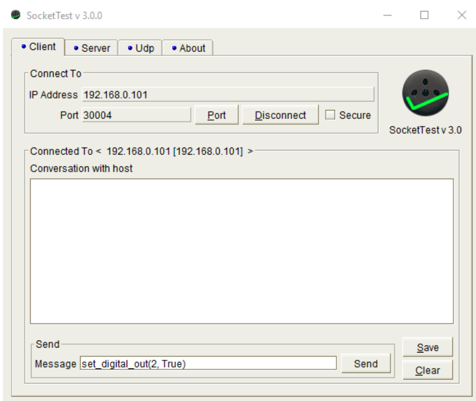


Figure 5.9: The SocketTest connected to the UR10.

Figure 5.9 shows the SocketTest connected to the UR10 at address 192.168.0.101 and port number 30004.

When SocketTest is connected to the UR10, through the correct IP and RTDE port, a simple test of sending script commands to the UR10 was performed. In the "Message" box the command "set_digital_out(2, True)" was sent to the robot via the TCP socket connection (See the bottom of Figure 5.9). The command went through, sat DO2 to high, which on purpose did nothing physical, but proved that the connection worked (see Figure 5.10) [18].

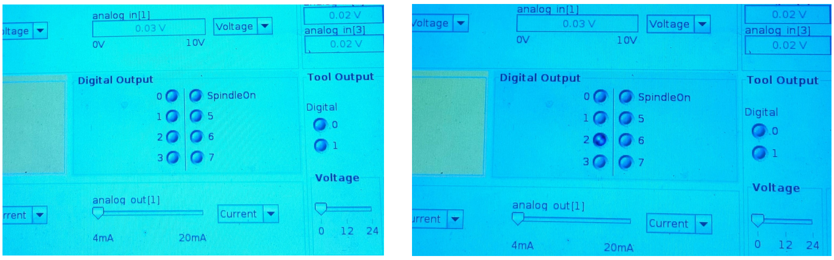


Figure 5.10: The I/O screen on the PolyScope controller before and after sending the DO2 command from the SocketTest on the computer. Notice under ”Digital Output”, number 2 is turned on.

The test proved that the Client-Server and the socket connection works. The computer successfully sent the UR10 simple digital output commands, which means that VC could be used to control the assembly tools. However, VC still had no way of controlling the motion of the robot.

Because there were no problems with the RTDE socket, the problem was assumed to have been because VC was a generic OLP software, and that it because of this might not have been compatible with sending script commands to the UR10 directly. This sparked an idea of creating an indirect connection – to go from VC to UR10 via URSim.

5.3 Connectivity via URSim

This section explains the process of establishing a connection from VC through URSim to control the UR10. It was believed that the communication problems directly between VC and the UR10 were because VC was a generic OLP software, and therefore not 100 % compatible with the UR10. It was also believed that this could thus be solved with a detour via Universal Robots own proprietary OLP software: URSim. The goal was to find a solution where VC controls URSim, which further controls the UR10.

URSim is a simulation software that is used for offline programming and simulation of robot programs. It is only created for Linux, so in order to run URSim on another operating system (Windows in this case), a virtual machine was needed. A virtual machine is a program where multiple operating systems can be installed, including Linux. In this thesis, the freeware VirtualBox was used to run Linux. The steps of this guide [19] were followed for the installation of VirtualBox, which can be seen in Figure 5.11.

The steps of connecting from VC to URSim were done by following the same step-by-step process from Chapter 5.2.2, only with the virtual robot controller in URSim instead. The connection was once again successfully established, and both cases of VC managing to

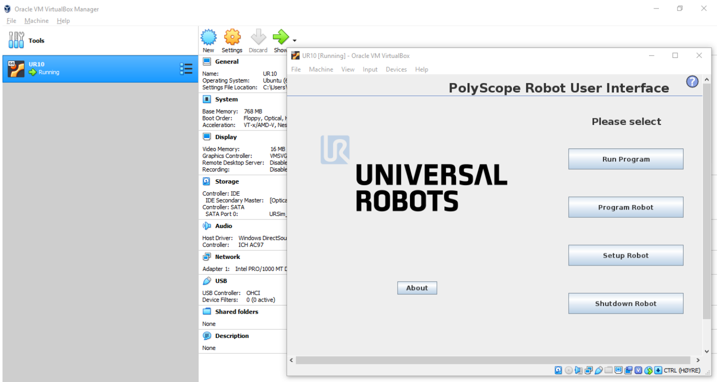


Figure 5.11: PolyScope running on Linux in the virtual machine on the same computer as VC is used with Windows.

control the simulated UR10 in URSim, and URSim managing to control VC worked – proving a two-way communication was established. The result can be seen in Figure 5.12. Notice that the two robots have identical postures and joint rotation values.The two robots were linked, so moving the UR10S in VC would cause the same movement with the UR10 in URSim.

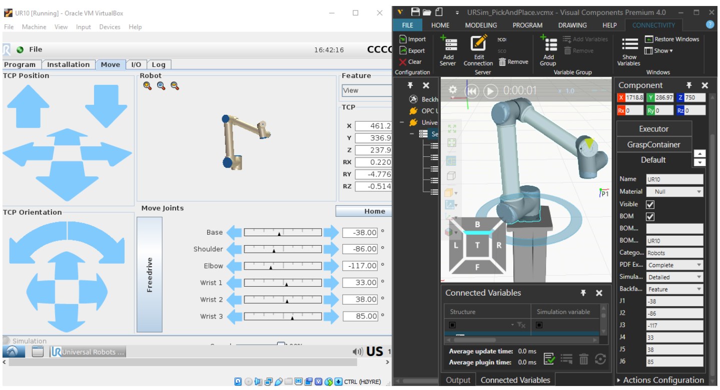


Figure 5.12: VC (to the right) and URSim connected with two-way communication.

Even with the successful connection between VC and URSim, a way through to the UR10 was still not achieved. URSim was not able to send motion commands to the real UR10 through the socket connection. After an extensive forum search and emailing back and forth with VC Customer Support, it turned out that there is no possibility to control outputs from simulation (VC or URSim) directly to the real robots, and furthermore, that this is a limitation set by Universal Robots themselves. This was a major disappointment, as a lot of time had already been invested in trying to find an OLP solution. It was not expected that not even URSim could connect with the UR10, given the fact that they were both developed by the same company. Eventually, some more emailing with VC Customer Support resulted in finding a possible workaround for the problem!

5.4 The workaround

A workaround for acquiring direct one-way communication from VC to the UR10 was found. The workaround solution wrote the VC simulation variables to registers. These registers were sent to the UR10 where a robot program then read the VC variables from those registers. The data synchronization protocol of the UR controller, which was mentioned earlier (Chapter 5.2.1) was exploited. The RTDE synchronization was in fact configurable and could for example involve the following data [15]:

- Output: robot-, joint-, tool- and safety status, analog and digital I/O's and general purpose output registers
- Input: digital and analog outputs and general purpose input registers

This shows that, the UR10 could take digital and analog outputs and general purpose input registers as input. This explains why it earlier was possible to set the DO2 high, but not to send motion output commands. It is the registers that were capable of connecting the simulation variables of VC to the UR10 – including robot joint orientation variables.

5.4.1 Setting up the registers

To prepare the registers, the simulation variables of the UR10S has to be linked to the corresponding joint variables of the UR10. As earlier, this is done in VC under the Connectivity tab after an RTDE server has been added to the UR10S:

1. Right click the "Simulation to server" and select "Add Variables". The "Create Variable Pairs" panel should show up.
2. On the left side: Expand UR10 → Behaviours → RobotController (or CB3) → The "J1" to "J6" options should appear. Expand J1 → Choose "J1".
3. On the right side: Expand Inputs → Double Registers → Choose "input_double_register_1".
4. Click "Pair Selected". An example is shown in Figure 5.13.
5. Repeat steps 2-4 with J2 to J6.

The registers are then complete. Under "Connected Variables" it should look like in Figure 5.14. Note: when the simulation is running on a correctly applied RTDE connection, the

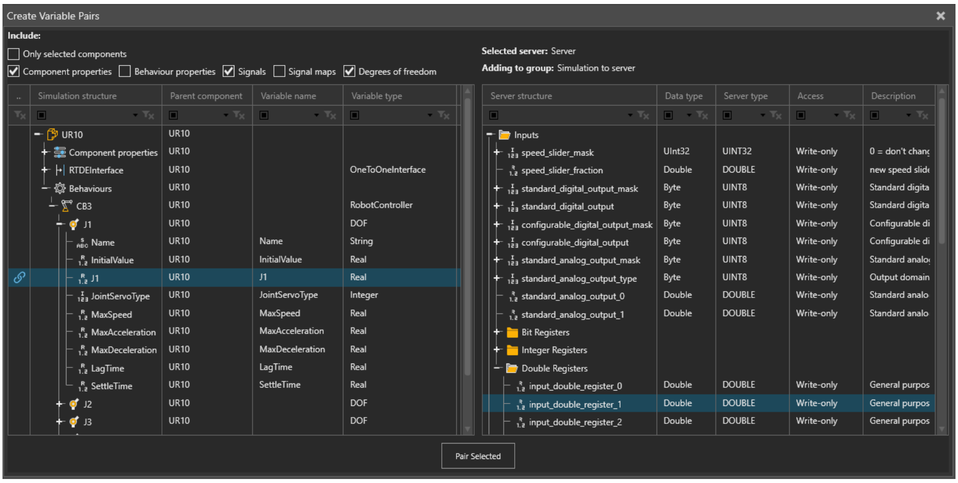


Figure 5.13: The Create Variable Pairs panel with one variable pair selected.

statuses of the links turn into green checkmarks. Before this they are gray question marks, indicating a pending state.

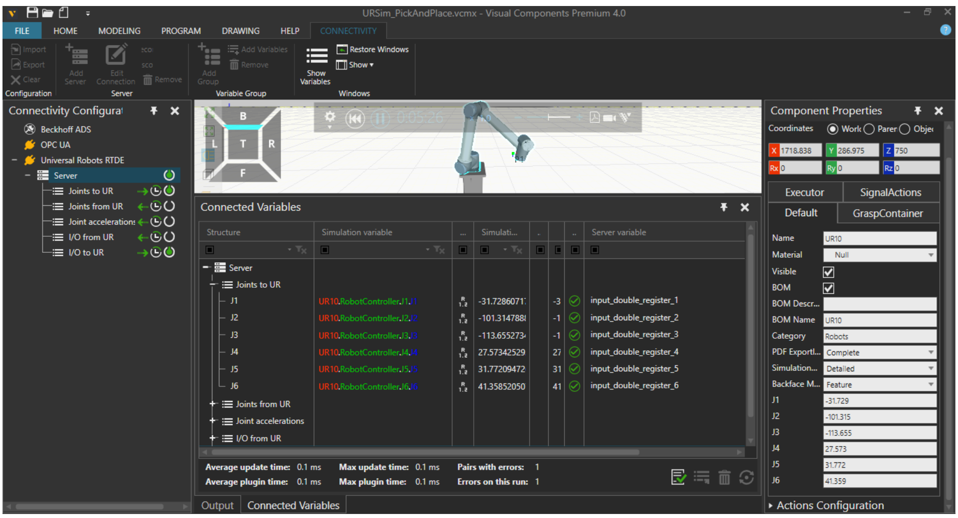


Figure 5.14: Successfully created registers.

VC continuously send these registers containing the UR10S joint variables to the UR10, whenever the simulation in VC is running. The UR10 receives these registers via the TCP/IP connection of the RTDE socket connection. For the UR10 to be able to read the input registers, a synchronization loop program must be created.

5.4.2 The RTDE Synchronization Loop

The RTDE functionality was divided into two stages: a setup procedure and a synchronization loop. By creating an RTDE Synchronization Loop program in the robot controller of the UR10, the robot continuously look for updates, i.e. general purpose input registers from VC. This is accomplished with a while loop that connects the joints of the UR10 to the read input registers. The script implemented in the PolyScope robot controller can be seen in Figure 5.15. Notice that in PolyScope the while loop makes the robot joints (i.e. j1, j2,...) read "input float registers". These are the same registers called "input double registers" in VC in Figure 5.14. Why Universal Robots uses different names for the group in Polyscope and RTDE is unknown, but the different names do not cause any trouble. For students and others who wish to replicate the connection and have access to the UR10 of NTNU, this program can be loaded in the robot controller (called "TA RTDE Synchronization Loop"). To activate the while loop, the first line must be set to "True".

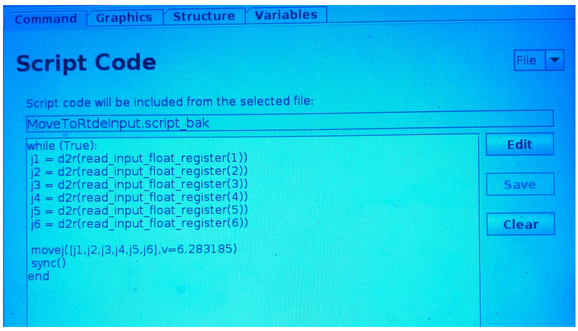


Figure 5.15: RTDE synchronization loop script on the PolyScope controller

5.4.3 Establish the connection

After all the simulation variable and input register links have been set up, and the RTDE Synchronization Loop script has been programmed, perform the following procedure to establish the connection between VC and the UR10:

Steps in the PolyScope controller:

1. Start and initialize the UR10.
2. Load the program "TA RTDE Synchronization Loop.urp".
3. Select the first statement "Script: MoveToRtdeInput.script_bak".

4. Modify the code, by changing ”False” to ”True” on the first line, then press save.
5. Run the program.
6. Auto move into position ⟶ press OK ⟶ Press play.

While the while loop is running, the robot is continuously looking for RTDE input registers. To enable VC to start sending input registers follow these steps:

Steps in VC:

1. Open the saved layout containing all the variables linked to registers.
2. Go to the Connectivity tab.
3. Activate the server connection as in Step 4 of 5.2.2: Edit Connection ⟶ Define the IP and RTDE port ⟶ Test the connection ⟶ Accept.
4. Connect the server by clicking the circle next to ”Server” (should go from gray to green).
5. Enable the group ”Simulation to server” (should also go from gray to green).
6. Run the simulation.
7. Jog the robot in VC and see if the UR10 is moving.

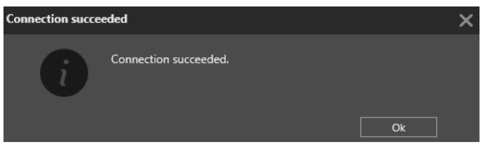


Figure 5.16: Connection succeeded.

5.5 Scientific grounding and rarity assessment

During the project thesis, an extensive literature review of OLP was done. There are thousands of articles on the subject. However, the case of OLP with VC is not as common. A literature review was performed to find scientific grounding and to assess the rarity of the OLP connection with VC. Different combinations of the words TCP, IP, remote control, offline programming, UR10, and such; in combinations with Visual Components; gave very few hits on scientific search engines (e.g. Scopus, Google Scholar, IEEE Xplore, etc.) – most of which only mentioned Visual Components once as an example of an OLP software. Three papers were found, which to some extent are related to this thesis.

In the two first papers [20][21], both written by the same author, other Visual Components simulation solutions are found. to be intuitive solutions. The findings reveal the same as stated in this thesis: ”Removing complexity from end-users. (...) Even beginners will be able to quickly design robot cells and benefit from the virtual world of robot simulation.”

In this third paper [22] a UR10 was used in VC for a simulation-based feasibility study of the possibility of using cobots: ”The objective in the experiments with the Visual Components software simulation was loading and unloading of CNC machines. (...) The RoboDK libraries include robots from tens of manufacturers and Visual Components from hundreds of manufacturers. Visual Components provides visual, easy-to-use software with a huge variety of robot and accessory libraries and simulation features. (...) We found that the cobots and the simulation and programming software used in the experiments can be applied with non-experts and are well-suited for small-scale industries.”

The third paper also explains how to design robot work cells in VC the same way it would be done in this thesis, and justifies why it worked with a UR10: ”First, the robot and the gripper were selected from the huge variety in the software library. The work cell was constructed and modeled using SolidWorks 3D CAD software. Then the work cell was imported to the Visual Components simulation software, and the components were positioned in the correct places. After the simulation model was completed, it was used for the feasibility study where, for example, robot reach, payload, work cycle, possible collisions, and safety issues were analyzed. The simulation showed quickly that the reach and payload of UR3 robots are not sufficient; therefore, UR10 robots were chosen instead. With UR10 robots, the loading and unloading cycles were simulated successfully after various numbers of iterations.”.

The last paper could potentially have taken it one step further and used the simulated UR10 robot programs with a TCP/IP remote control connection to perform the feasibility study physically with a real UR10.

Chapter 6

Results

6.1 Results of developing and testing the assembly tool

In Chapter 4, the process of developing the total assembly tool was described. The tool consisted of the dual tool mount, the sealant tool and the vacuum gripper. It was used for picking and placing the aluminium parts and automatic sealant application. In this section the results is presented.

6.1.1 Results of developing the dual tool mount

To make an autonomous solution, the possibility to perform both sub-processes of the assembly with one robot was needed. Parts for a dual tool mount were modeled, machined, and assembled. The resulting dual tool mount could mount both the required assembly tools to one UR10. The final design was shown in Figure 4.7. The aluminium extender solved the problem of the two mounted tools colliding with each others work space and the weld work fixed the rotation problem.

6.1.2 Results of developing the sealant tool

The sealant tool went through several iterations before the final design. Some of the results of these design iterations have already been mentioned. For example, that the ring support was cut in half, the modified caulking gun was modified further, the first polymer support broke, and additional actuator supports were developed. These results were included earlier in Chapter 4 to not ruin the continuity of the thesis. This section focuses on the performance and results of the final version of the sealant tool used during sealant application testing. This includes the biggest issues with the mechanical and mechatronical designs, as well as the main results from the sealant application tests.

Results of the mechanical setup

The final mechanical setup, with the modified caulking gun and the proper actuator supports worked well and resulted in a stable and sturdy sealant tool (Figure 6.1). The silver steel support successfully restricted any unwanted movement of the actuators, while the second polymer support held the actuators in position to precisely aim the caulking gun with help of the plunger head support.

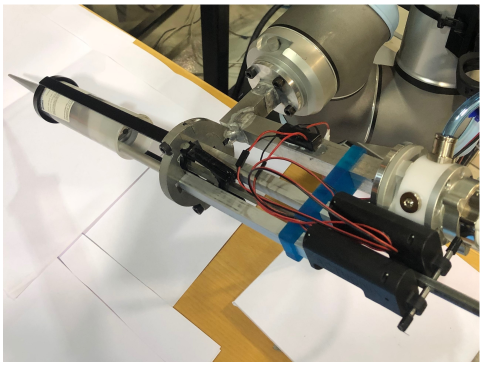


Figure 6.1: Stable and sturdy final design.

Initial testing of the mechanical setup revealed a design flaw of the caulking gun. When the linear actuators made contact with the sealant cartridge, instead of pushing out sealant the entire cartridge was pushed through the end of the caulking gun. The problem is shown in Figure 6.2.

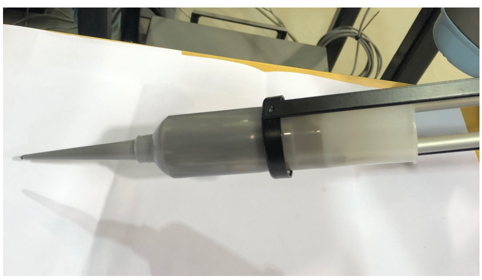


Figure 6.2: Caulking gun design issue.

This had not been a problem during other tests with the same caulking gun prior to creating the sealant tool. Suspecting the problem could be because of that particular sealant cartridge, another caulking gun was tested. The result, which can be seen in Figure 6.3, showed that the sealant had completely solidified and become incompressible. The caulking gun design issue, was in fact a sealant cartridge issue. Still, a solution to the caulking gun design flaw was machined to hinder it from happening again. A flat metal ring was made, and is shown in Figure 6.4.

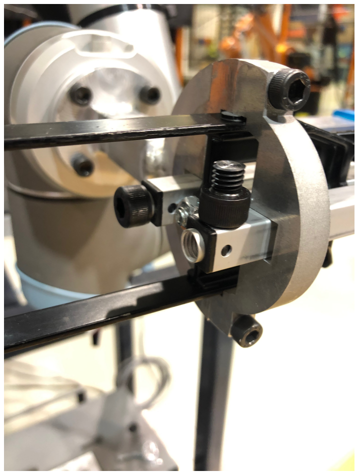
Figure 6.3: Sealant cartridge issue.

Figure 6.4: Flat metal ring solution to the caulking gun design flaw.

During the next round of testing, the plunger head support broke. The plastic tips of the linear actuators could not withstand the friction of movement inside the sealant cartridge and fractured (see Figure 6.5a).

The result of the fractures was that the plunger head support no longer had something to attach to. A quick fix was performed by grinding down two M8 screws and attaching them to the actuators instead, seen in Figure 6.5b.

(a) Fractured actuator tips.



(b) Plunger quick fix: two shortened M8 screws.

Figure 6.5: Broken plunger head support and the quick fix solution to it.

Results of the mechatronical setup

For the mechatronical setup, initial tests consisted mostly of getting the actuators to work both ways. The main iterations of the DPDT system design were presented in Chapter 4.2.2. Creating the multipurpose DPDT solution and sealing it inside the protective plastic cover worked well. The semi-automatic solution served it's purpose when the DPDT switch was on extend mode. The actuators functioned as wanted and were controlled automatically by the robot controller via the digital output signals.

The functionality of the DPDT switch eventually became unreliable. Switching the power source to the battery pack to retract the actuators did not work consistently. The top of the box was sawed off to discover the issue. The wires inside the DPDT box connected to the battery pack had melted – not completely, but enough to weaken the functionality of the switch (see Figure 6.6). The wires probably had too small cross sections to handle the current from the batteries. The DPDT switch system was rebuilt with thicker wires and delivered stable results throughout the remaining experimental period.

Results from the sealant application tests

The sealant tool with the plunger head support had a precise tool center point with good accuracy. After the plunger head support broke and was fixed with two M8 screws, the cartridge was less supported and became imprecise. Nevertheless, for the sealant application tests the tolerances were not too strict, and the sealant tool was precise enough. The

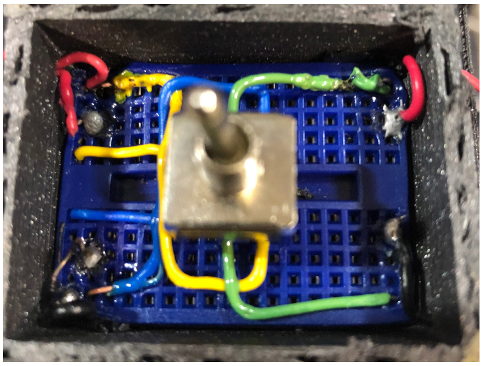


Figure 6.6: The inside of the DPDT box showing the melted wires.

tests were simple: the UR10 was programmed to 1) slowly approach the start point of the sealant strip, 2) apply a straight stripe of sealant on the aluminium plate until the endpoint of the strip is reached, and 3) turn off the sealant and move a couple of centimeters away from the endpoint. The goal was to figure out how to tune the actuators to apply the perfect amount of sealant and to check the reliability of the tool. Videos of the tests are appended in the digital appendix.

Initial tests: The result of the first tests was that the sealant application worked smoothly and produced continuous lines (see Figure 6.7). However, the sealant strips were too thin i.e. the nozzle exit was too small, so the tip of the nozzle was cut off to increase its cross section area.

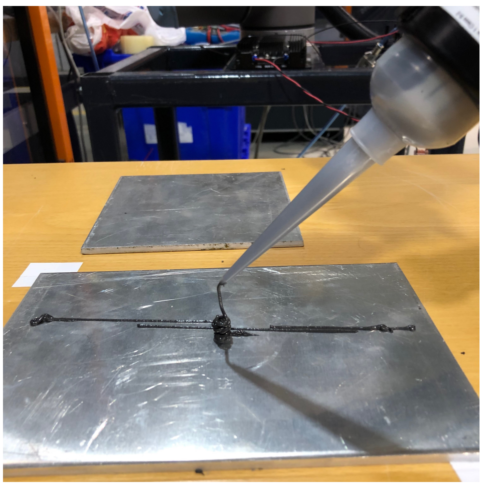


Figure 6.7: Results of the initial tests.

Another result from the initial sealant test is shown in the same picture above. Notice the pile of sealant in the middle of the picture. The actuators worked well in pushing out sealant, but there was no way of automatically stopping the sealant from pouring out after the actuators had compressed the cartridge. A leakage test was performed to see how severe the leak was.

Leakage test: The sealant application robot program was turned DO1 high to activate the actuators at the starting point of the strip, and turned DO1 low to cut them off at the end point. It took approximately three seconds to make a 20 cm long strip. A test of exactly two seconds of push from the actuators was performed, which resulted in leakage for more than two minutes from the nozzle. The amount of sealant squeezed out during the test can be seen in Figure 6.8. Note that it is not a huge quantity because the sealant was so viscose, but it looks like less than it actually is because the sealant slumped together over the time span of two minutes. A later test showed that the leakage stopped if the actuators were pulled back – releasing the pressure inside the sealant cartridge.

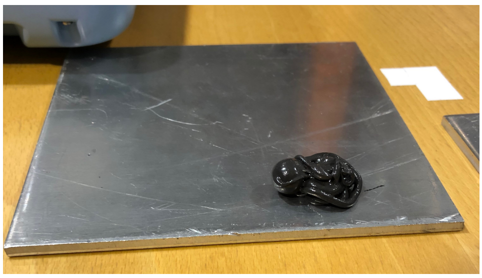


Figure 6.8: Result of the leakage test.

Application tests: In Figure 6.9, five tests of applying sealant in straight stripes are shown. The first test is pictured on the bottom, while the last test is on the top of the picture. The tests showed that the sealant struggled to stick to the aluminium, but rather curled up and twisted around the nozzle instead. This happened even when the nozzle was sliding along the surface, touching the aluminium plates. Results from the tests can be seen in Figure 6.10. Also, piles of sealant can be seen near the start and endpoints of the sealant stripes because of leakage.

Figure 6.9: Results of the five application tests.

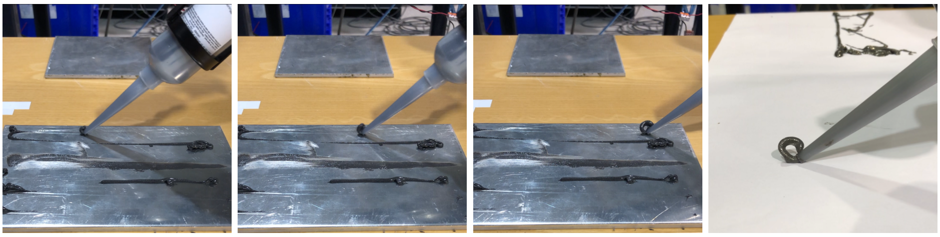


Figure 6.10: Sealant curling instead of sticking to the surface.

Paper tests: Paper tests were performed to test if the sealant was not sticking to the aluminium during the automatic application because of its material properties. The sealant curled on the paper too, which can be seen to the right of Figure 6.10. Yet, the paper worked better than the aluminium in creating continuous sealant strips. The paper results were not optimal as the assembly parts never would be covered in paper, but it saved a lot of time with the cleanup for further testing. The test results were still relevant and helpful for tuning the actuators. Results from the paper tests can be seen in Figure 6.11.

(a) Continuous strip of sealant, sticking nicely to the paper.

(b) Background separated to hide sealant spill from other tests. Notice the loop on the left which was caused by curling.

Figure 6.11: Results from the paper tests.

End of testing: After the paper tests it was discovered that the sealant cartridge was destroyed internally. The end cap, which was being pushed by the two M8 screws, had twisted and bent out of position, causing sealant to escape through the wrong end of the sealant cartridge. The problem is shown in Figure 6.12. This might explain why the sealant flowed so slowly out from the nozzle. This was the last remaining cartridge of sealant, which made further testing impossible.

(a) Broken cartridge. **(b)** Proper cartridge.

Figure 6.12: Broken and proper sealant cartridges.

6.1.3 Results of setting up the vacuum tool

The pneumatics worked flawlessly. The pressure reduction valve managed to reduce the flow of compressed air, to save energy, and the solenoid valve turned the vacuum gripper on/off on demand with negligible delay. As long as the vacuum gripper picked up the parts approximately above their center of gravity, no problems were encountered. Picking up a part further away from its center of gravity caused the part to lean over on its heavy side, which sometimes resulted in imprecise placements, but, for the small-case pick and place scenarios of this thesis, the vacuum tool setup worked perfectly.

6.1.4 Results of the finished assembly tool

A big focus point throughout Chapter 4 was the total weight of the assembly tool. With everything mounted, the complete tool finally weighed in at 1492 grams. The maximum payload the UR10 could lift was thus 8.508 kilograms. Additionally, the robot cable management system was mounted on the robot contributing to the total weight, but it was still more than enough lifting power to handle the small aluminium plates used during testing of the autonomous system.

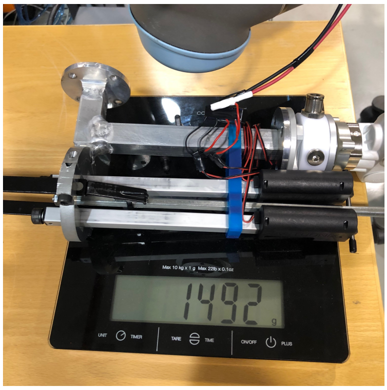


Figure 6.13: The final assembly tool weight.

To see if the UR10 was strong enough to carry the aluminium parts of the assembly, the masses of the assembly parts were found. The volume of the CAD model of the frame from Chapter 2, can be seen in Figure 6.14 to be 2537.640 cm^3. With the density of aluminium at 2.7 g/cm^3, the total mass of the frame equaled approximately 6.850 kilograms. The masses of the cover and lid were 1.126 and 0.566 kilograms respectively. The full assembly therefore weighed in at 8.542 kilograms.

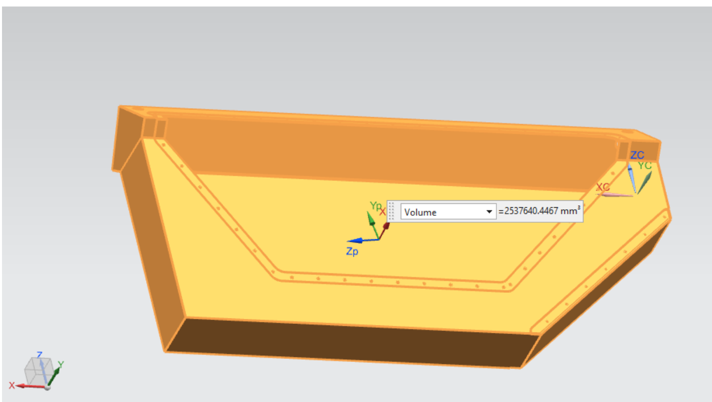


Figure 6.14: Volume of the frame.

6.2 Results of offline programming of the UR10 with Visual Components

Enabling the connection between VC and the UR10, involved setting up a remote control via TCP/IP sockets hooked up with the robot controller data streams. The goal was to make a real-time client with VC for controlling and monitoring the UR10 remotely.

It was discovered in Chapter 5, that the UR10 actively worked against receiving direct motion output commands, and furthermore, that this was a limitation set by Universal Robots themselves. To bypass this, connectivity from VC to URSim was achieved, but further connection to the UR10 was not. The VC to URSim two-way communication indeed had exciting opportunities with many advantages, unfortunately none of them were particularly relevant for the scope of this thesis.

A workaround for acquiring direct one-way communication from VC to the UR10 was found and successfully implemented in Chapter 5.4. Without knowing it at the time, the procedure of subsection "Linking simulation and server joint variables" of Chapter 5.2.2, was close to a working solution. A keen eye might have noticed the grand mistake: The first connection with the linked variables was in fact not an output from the simulation to the server, but an input from the server to the simulation.

6.2.1 Results of successfully establishing the workaround connection

The workaround connection was an output from the simulation to the server. The solution this time, wrote the VC simulation variables to registers instead, and it were these registers that were capable of connecting the simulation variables of VC to the UR10 – including robot joint orientation variables. After successfully linking the variable pairs of the UR10 and the UR10S, the connection was established: In the PolyScope controller the RTDE input loop was created and initiated, and in VC the server was setup and the simulation was ran. With the setup complete, the UR10 was able to read and mimic the motion commands of the VC simulation.

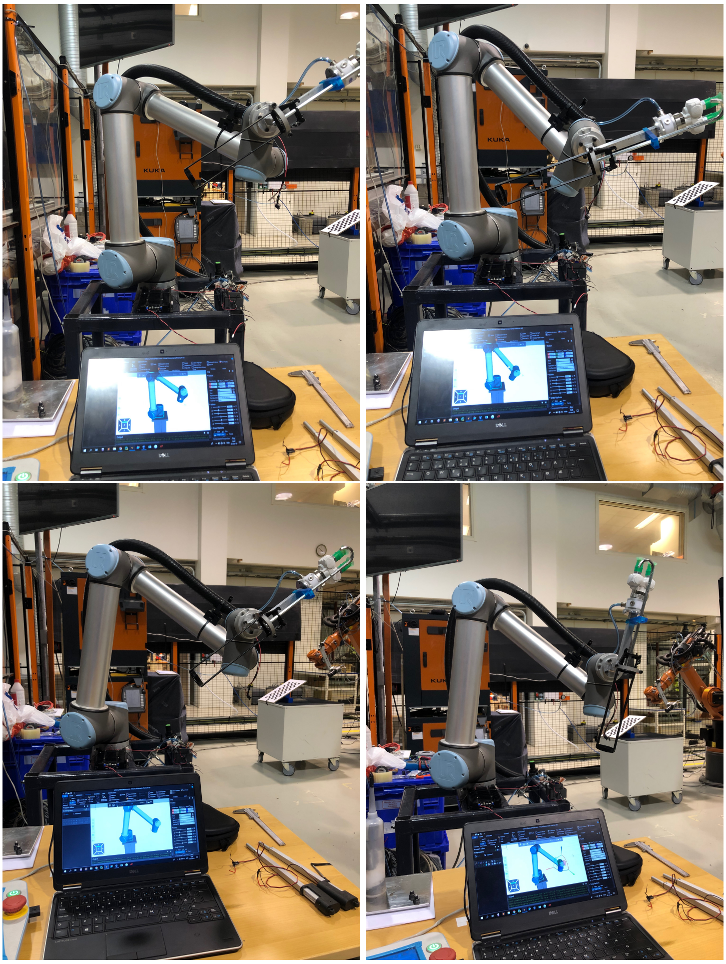
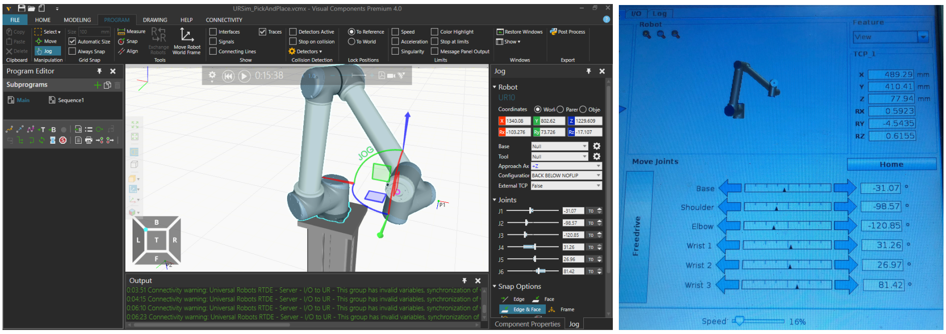


Figure 6.15: Remote controlling the UR10 with VC.

Results from using VC to remote control the UR10 can be seen in Figure 6.15. It may be hard to notice, but the UR10S (seen on the computer screen in VC) and the UR10 have the exact same postures, with identical robot joint variables. The UR10S was jogged around inside the simulation in VC. The UR10 was remote controlled in real-time and moved while the UR10S was being jogged. Screenshots of VC and the PolyScope robot controller are shown in Figure 6.16.



(a) The UR10S shown in VC. It has the same posture as the UR10 seen down to the right of Figure 6.15

(b) The UR10 shown in the PolyScope robot controller.

Figure 6.16: Robot postures and joint variables during OLP remote control of the UR10.

(a) The UR10S in VC.

(b) The UR10 in PolyScope.

Figure 6.17: Joint variables of the UR10S and UR10 during OLP remote control.

In Figure6.17, the robot joint variables are zoomed in on. A slight deviation in Wrist 2 (J5) can be seen. This was because the UR10 continuously sent updates, and even small vibrations or draft in the workshop next to the robot lab affected its displayed values. The robot base structure was not completely in balance either. These factors had no effect on the overall performance of the UR10, but affected the PolyScope representation of the joint variables, which were quite precisely displayed with two decimals. Videos of the remote control are appended in the digital appendix.

6.2.2 Results of the literature review

In Chapter 5.5 a literature review was performed to find scientific grounding and to assess the uniqueness of the OLP connection with VC. Searches for scientific articles on OLP with VC, gave scarce results. Three articles were highlighted, but were only vaguely relevant to the scope of this thesis. The only viable results found were that all three articles support the use of VC for different simulation solutions and that the GUI of VC makes the solutions so intuitive that non-experts with low skill levels can implement them.

Chapter 7

Analysis and discussion

7.1 Discussion of developing and testing the assembly tool

Chapter 6.1 presents the results from the process of developing the total assembly tool. The following discussion of the assembly tools revolves around its three main parts: the dual tool mount; the sealant tool, with the automatic sealant application tests; and the vacuum gripper. In the end, the total assembly tool as a whole is evaluated.

7.1.1 Discussion of the dual tool mount

The dual tool mount was invented as an automatic tool changer workaround. The development of the mount is described in Chapter 4.1, with its results being presented in Chapter 6.1.1. In this section, the dual tool mount solution is discussed from three different points of view: in regards to the experiments of this thesis, the bigger picture, and the full assembly process at KDA.

For the thesis: The dual tool mount worked fine in dealing with the automatic tool changer problem. It could be argued that two robots with one tool each, or an automatic tool changer would be a better solution, although, it ultimately depends on the use case scenario of the process, but in this case just one UR10 was going to be used. To carry out the experiments of this thesis, the dual tool mount worked perfectly. Both the sealant tool and vacuum tool were able to perform their tests while mounted to the dual tool mount.

For the bigger picture: For processes that only require two tools, a dual tool mount solution, which rotates between two tools can be quicker and cheaper than investing in an automatic tool changer system. A system utilizing a tool changer base station, where the robot has to stop by every time it needs to change between tools, would probably be slower than just rotating the outermost joint 180 degrees. This may be the case for processes of more tools too – as mentioned; it depends on the use case scenario.

Possible limitations of a multi tool mount is that some tools may not be capable of being mounted on a multi tool mount. For instance, thee tool might be huge, or have a complicated geometry, which could make the design of a dual tool mount difficult. Additionally, if more complex tools are needed, e.g. several tools with hoses like the vacuum gripper or a pneumatic/electric bolting gun, it is easy to visualize lots of hoses, tubes and wires entangled in a big, messy knot after some revolutions. In these cases, an automatic tool changer may be the best option. A conveyor belt system with several robots along the belts, each assigned with their own task and corresponding tools could also be possible solutions.

For the full assembly process: For the full assembly process of the aluminium parts discussed in this thesis, it would be challenging to perform everything with only one robot and a multi tool mount. For the subprocesses of picking and placing the parts, applying the sealant, and mounting the parts, one robot can be used with a dual tool mount wielding a gripper and a sealant tool. For the additional subprocess of bolting the parts together, a triple tool mount could perhaps be used, but another robot or an automatic tool changer are recommended instead. It is also likely that one robot is needed to hold the parts still after mounting, while another tool bolts them together.

7.1.2 Discussion of the sealant tool

The development of the sealant tool is described in Chapter 4.2, with its results being presented in Chapter 6.1.2. In this section, the performance of the mechanical and mechatronical setups are discussed, as well as the results from the sealant application tests. Finally, the overall performance of the sealant tool is evaluated.

Discussion of the mechanical setup

The initial mechanical design was a disaster, but after creating the necessary actuator supports and modifying the caulking gun, the mechanical setup of the sealant tool became robust and ready for testing.

The silver steel support and the second polymer support worked well – the vertical print made the polymer support as precise as required. The ring support worked nicely for mounting everything to the dual tool mount, however the actuator tracks became somewhat obsolete with the new actuator positionings after the caulking gun modifications and the new design for the plunger head support.

The two biggest issues of the design during the tests were connected to third party hardware. The sealant cartridges were not held properly inside the caulking gun and the plastic tips of the actuators broke off with the plunger head support.

The cartridge gun issue could easily be fixed. Either with another gun suited for the cartridges or fixing a blocker like the flat metal ring to the end of the caulking gun.

The result with the most impact was that the plunger head support broke. Initially, this was thought to be a trifle, but during the sealant application tests it became evident that the plunger head was essential for pushing out sealant correctly. The M8 screws pushed fine, but because they did not cover the entire cartridge cross sectional area, the back cap skewed out of position. This caused sealant to escape through the wrong end of the cartridge, which led to the sealant not properly coming out of the nozzle. Additionally, this affected further testing of sealant application, which was desperately needed to tune the tool properly.

A functioning plunger head design would have solved this problem. Figure 7.1 shows a potential solution using only one linear actuator with the plunger head directly attached to the actuator with an M8 screw. This redesign would have had a stable plunger head to push out sealant steadily. The ring support should be remodeled more effectively to solely focus on anchoring the caulking gun, and the caulking gun would have to be remodeled to hold tightly on to and support the actuator. This design is arguably simpler, cleaner, and more elegant than the solution presented in Chapter 4.2.2.

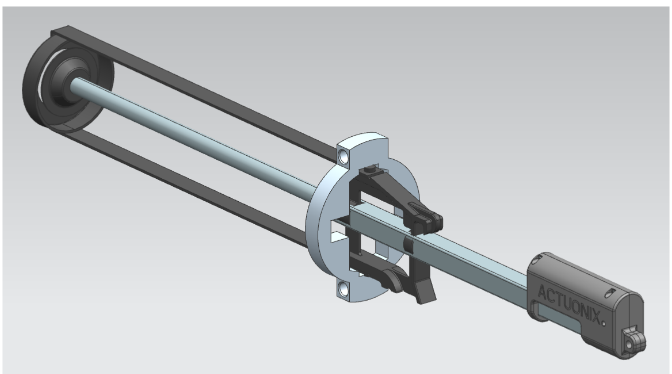


Figure 7.1: Potential redesign of the sealant tool.

Discussion of the mechatronical setup

The mechatronical setup became unnecessarily complex and cumbersome. Firstly, with 300N pushing power, one linear actuator would have been enough to push out the sealant. This would result in a simplified cabling system – making the parallel circuit redundant – and the redesign of Figure 7.1 would have been possible. Secondly, and more importantly, the DPDT switch system did not allow for a fully automatic solution.

The overall objective of the thesis was to find an automatic solution. As mentioned in Chapter 6.1.2, many design iterations revolved around finding a solution to both extend and retract the linear actuators. Eventually deciding to use a physical DPDT switch was a huge mistake. The semi-automatic solution served it's purpose when the DPDT switch was on extend mode, and originally this was considered a convenient solution. However, as the results of the leakage test showed; the leakage could have been stopped by pulling the actuators slightly back, releasing the pressure inside the sealant cartridge. A fully automatic solution, able to both extend and retract the actuators should have been pursued from the beginning of this thesis and might have improved the results from the sealant application tests.

Utilizing the digital output signals of the robot controller was an ideal solution to control the actuators. It could also have had the potential of being used in a fully automatic solution. The solution would require the DPDT switch to be flipped with a signal instead of a physical switch. In Figure 7.2, a relay switch solution is shown. The relay switch can be signal-controlled and thus presents an automatic solution. Using the robot controller, this circuit would need a voltage converter to supply 12V to the actuator and to be connected to two DO signals and 0V/ground. One DO signal to trigger the DPDT switch and reverse the voltage over the actuator, and another DO signal to turn the power on/off – this way the actuator could be controlled to extend, retract, or stand still.

The plastic cover worked well in packing the DPDT system together neatly, however, the DPDT system was initially intended to be placed on the assembly tool and the cover was developed for this and to protect the DPDT system while the robot was in motion. The redesigned system with the DPDT box mounted on the robot base structure made the cover superfluous – a ”nice to have”-function, but not really necessary – specially for an automatic solution where the battery pack and physical switch are not needed.

Discussion of the sealant application test results

The goals of the sealant application tests were to figure out how to tune the sealant tool to apply the perfect amount of sealant and to check the reliability of the tool. Factors that

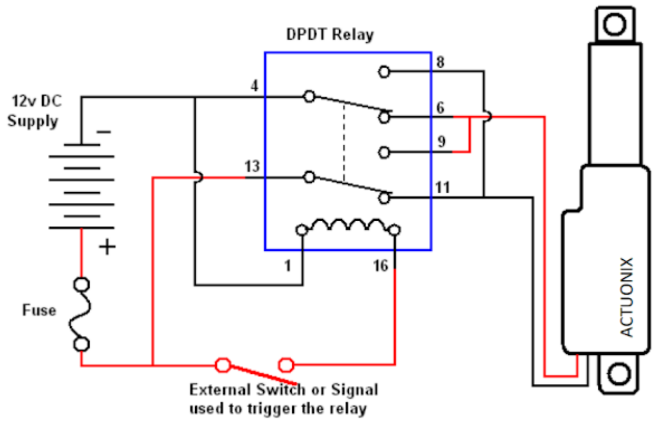


Figure 7.2: An automatic DPDT solution, using a signal-controlled relay to switch the DPDT switch.

could be tuned were; the pushing speed of the linear actuators, the movement speed of the robot, and the angle of the nozzle to the surface.

The results of Chapter 6.1.2 shows that the sealant curled and leaked during the sealant tool tests. Figure 7.3 shows both these issues summarized in one picture. The starting point of the nozzle movement was further to the left in the picture. At the start point the sealant curled instead of stick to the surface. The curling created a loop (Figure 6.10 shows how it developed) that eventually became too heavy and fell down onto the surface. This loop was dragged out by the motion of the nozzle towards the right, which can be seen in the sealant to the left. A continuous strip was made until the endpoint, where the nozzle moved away horizontally. The sealant did not stop leaking and amounted to the excess sealant seen in the top right of the picture.

Figure 7.3: Sealant application errors.

Leakage test: The sealant tool definitely managed to apply sealant. The biggest problem was to make it stop. The leakage test showcased the high viscosity of the sealant, causing the sealant to flow out of the cartridge very slowly. When the actuators pushed on the cartridge and created pressure inside the tube, the sealant was forced out through the nozzle. But, because the sealant flowed so slowly, there was still pressure remaining inside the cartridge, which continued to force out sealant after the application was stopped.

The actuators should have been retractable by command of the robot controller to enable a robot program where the pressure could be released after reaching the endpoint of the application surface. If the actuators could have retracted, the sealant would have been able to expand inside the confined cartridge instead of pouring out through the nozzle.

Application tests: The application tests, performed on both aluminium and paper, showed that it was not because of the material that the sealant did not stick to the surface of the parts. Based on the fact that the initial sealant test produced a straight, continuous (too thin) strip of sealant, it could be the cutting of the nozzle that ruined the sealant application. The cut altered the nozzle exit, which might have caused the sealant to curl instead of applying itself, as shown in Figure 6.10 of Chapter 6.1.2.

Possible solutions to make the sealant stick could be slower robot movements and a sharper tool angle. Slower robot movements would have given the sealant more time to fall down onto and stick to the surface at the start point. A sharper sealant tool angle (closer to being perpendicular to the surface), could also have helped preventing the curling effect and improved the application process. Another solution could be to cut a wider cross sectional area for the nozzle exit to mitigate the curling effect, allow more sealant to flow out, and rather adjust the robot's movement speed accordingly. These hypotheses were never tested and can only be seen as speculations.

Furthermore, the sealant tool had expired one and a half year prior to testing (January 2018 and June 2019). As the author had no prior experience with the sealant before the project thesis of last semester, the sealant might have acted unusual during testing compared to sealant that has not expired – or it might not, hard to know without a reference.

Finally, too few tests were performed to statistically draw any sound conclusions. Only two cartridges were available for testing, which in theory should have been enough, but it was not foreseen that the last cartridge was destroyed almost right after it was opened. Moreover, searches for scientific articles containing ”Light Fueltank Sealant application” and such, did not provide further relevant scientific grounding for the discussion.

Discussion of the overall performance of the sealant tool

The final assessment of the performance of the sealant tool is summed up here in regards to the experiments of the thesis and the full assembly process at KDA.

For the thesis: During testing of the sealant tool the goal was to figure out how to tune the sealant tool to apply the perfect amount of sealant and to check the reliability of the tool. The sealant application testing completely failed to tune the sealant tool. Nevertheless, it is still believed that either with mathematical calculations or by fine-tuning, it is possible to find an automatic solution that applies the exact amount of sealant needed to create a 360 degree squeeze-out. The sealant is not expensive (500 NOK per cartridge) so it could be possible to verify that enough has been applied by applying in abundance. However, all excess sealant has to be removed, so this would not have resulted in an elegant solution.

Although the tool was never tuned, it was both mechanically and electrically quite reliable. With the plunger head support, the sealant tool was stable and accurate. The actuators behaved reliably too, and only had problems when the DPDT switch wires were melted by the battery pack. With the proposed design solutions given in the discussions of the mechanical and mechatronical setups (Figures 7.1 and 7.2), the sealant tool could be implemented in a fully automatic solution. While it is true that there exist an automatic sealant mixer for the sealant used in this thesis, the solution would have to solve the problems of handling the sealant cartridges automatically to be completed.

For the full assembly process: It is hard to argue that the sealant tool of this thesis can compete with a proper sealant tool made by a professional producer. Certainly, because the sealant tool was not capable of loading and unloading sealant cartridges automatically. For KDA, there are several sealant tools that automatically loads/reloads cartridges, which rather should be invested in. A tool that also can do the automatic sealant mixing would be the optimal solution.

7.1.3 Discussion of the vacuum tool

The setup of the vacuum tool is described in Chapter 4.3, with its results being presented in Chapter 6.1.3. The setup reliably provided the Venturi vacuum generator with compressed air at the correct pressure, and the solenoid valve never failed to turn the vacuum gripper on/off on demand. In this section, the performance of the gripper with suction cups is evaluated in regards to the experiments of the thesis and the full assembly process at KDA.

For the thesis: The vacuum gripper worked perfectly during testing of the assembly tool. This was rather expected as the gripper was built from a professional components set. The only point of improvement for the vacuum gripper would be to change the suction cups from bellow to flat. Bellow suction cups work better with soft packaging, whereas flat suction cups have better grip on flat, hard surfaces. The only reason this was not done, was because the two sets used in this thesis did not contain flat suction cups that fit with the vacuum gripper.

The vacuum tool was a bit unnecessary for the experiments of the thesis. No advanced pick and place tasks were to be performed. Considering how much time the development and testing of the sealant tool required, implementing the vacuum tool might have been dropped if the research was started over. It did have great educational benefits however, to learn how to implement a full pneumatics setup with all its components. The vacuum tests also had value as a feasibility study for implementation in the full assembly process with the real parts.

For the full assembly process: The vacuum gripper was tested on small-scale, simplified parts. Nevertheless, using suction cups to pick up the original parts mentioned in Chapter 2, could be a good idea. The original parts do not have obvious gripping possibilities for a mechanical gripper – especially the cover, which is almost completely flat. Furthermore, adding extra suction cups spread farther apart would give a wider, more stable grip, which would counter the problem of having to pick up the parts approximately above their center of gravity. A vacuum gripper could be a quick and effective solution for the pick and place subprocess.

7.1.4 Discussion of the finished assembly tool

With the full assembly of the aluminium parts weighing in at 8.542 kilograms, it exceeds the maximum payload of the UR10 by 34 grams. Technically, the UR10 would therefore be able to pick and place the parts individually, lift any combination of a sub-assembly of two of the parts, but not be able to lift the full assembly with the finished assembly tool developed in this thesis. It should be possible to make the tool 34 grams lighter, for instance by exchanging one of the aluminium actuator supports with polymer, or do the suggested redesign of the sealant tool with one actuator.

The UR10 is in fact capable of lifting heavier payloads than 10 kilograms at very slow accelerations, so an excess 34 grams may not cause any problems. Nonetheless, the safest option would be to invest in a bigger robot. Especially because the CAD models only are approximations of the actual aluminium parts, and in worst case may be heavier in reality.

7.2 Discussion of using offline programming with Visual Components to control the UR10

Having a connection directly to the UR10 from VC was an elegant way of controlling and monitoring the robot remotely. Chapter 5.1 mentions how this helped bypass the restricted access times of the workshop. Establishing the OLP connection meant that robot programs could be created in the office after closing hours, for later to be implemented to the physical robot when the workshop was open again.

More importantly the connection simplifies the design of robot programs. The intuitive graphical user interface (GUI) of VC may give an advantage when it comes to robot programming. This is because the GUI does not require any prior programming skills, but rather depends on jogging the robot around in the simulation. Controlling the robot and designing trajectories in VC is thus easier and less time consuming than coding robot programs from scratch – at least if one is not already proficient with URScript, Python urx, or other languages that are compatible with the UR10. Additionally, note that this would be the case for all industrial robots, not only the UR10 from Universal Robots.

A downside of using VC for offline programming of the UR10 is that the simulation requires CAD models of the entire work environment of the assembly process. The solution is, as mentioned, to take advantage of VC's CAD model catalogue, which already contains 2000+ models from hundreds of manufacturers, or other open source CAD sharing platforms. For distinctively unique parts the solution is to personally model the part in 3D CAD software (e.g. SolidWorks, NX) and import it into VC, which was done in this thesis and shown in Figure 5.1 and repeated below in Figure 7.4.

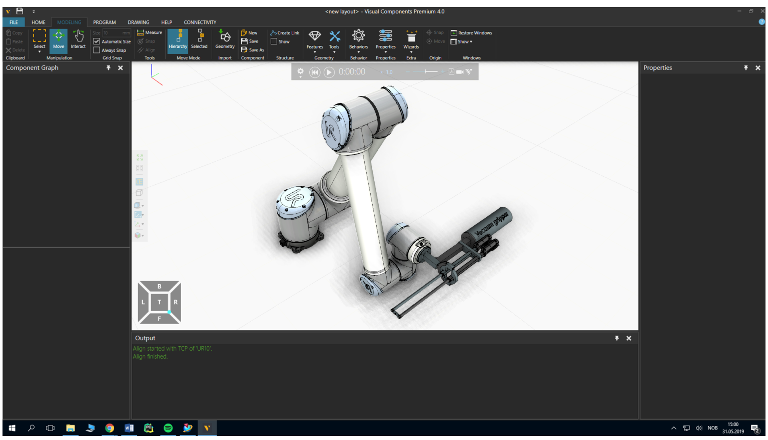


Figure 7.4: The CAD model of the assembly tool imported into the UR10S virtual environment in VC. Note: For some reason the components of the caulking gun would not import properly.

7.2.1 Discussion of establishing the workaround connection

Enabling the connection between VC and the UR10 involved setting up a remote control via the RTDE interface. The RTDE interface provided a way to synchronize VC with the UR10's robot controller over a TCP/IP socket connection. The goal was to make a real-time client for controlling and monitoring the UR10 remotely with VC.

Chapter 5.2.2 explains how to activate the RTDE server connection. Chapter 5.4 describes how the workaround for acquiring direct one-way communication from VC to the UR10 was found and implemented. Finally, Chapter 6.2 shows the results from the workaround implementation.

The results are clear: Successfully connecting VC to the UR10 has several advantages. Some of which are mentioned in Chapters 3.4 and 5.1. The biggest advantage by far, is the possibility of creating accurate robot programs without needing any programming skills. There are only two exceptions: the RTDE Synchronization Loop script must be programmed in the robot controller and the variable pairs of the UR10 and the UR10S must be linked in VC, to setup the general purpose input registers. However, this only has to be done once.

Establishing the connection between VC and UR10 to enable the remote control is easy and intuitive:

- Activating the server is five steps.
- Setting up the registers is five steps.
- The RTDE Synchronization Loop script consists of ten lines of code.
- The final setup of the PolyScope controller is six steps
- The final setup of VC is seven steps.

Once the server is activated, the registers are set up, and the RTDE script is written, these tasks do not have to be done again. If the robot program in the PolyScope controller is saved after changing the while loop to ”True”, the final setup of the PolyScope controller is reduced to only four steps. Same with the final setup of VC, which also can be reduced to four steps by saving after changing the parameters of the server. Finally, this amounts to a total of eight steps to enable the OLP connection and be able to remote control the UR10 with VC. The eight steps are listed below:

Steps in the PolyScope controller:

1. Start and initialize the UR10.

2. Load the program ”TA RTDE Synchronization Loop.urp”.
3. Run the program.
4. Auto move into position ⟶ press OK ⟶ Press play.

Steps in VC:

5. Open the saved layout containing all the variables linked to registers.
6. Go to the Connectivity tab. Double check that the server is connected and that the group ”Simulation to server” is enabled.
7. Run the simulation.
8. Jog the robot in VC and see if the UR10 is moving.

After the connection is enabled, jogging the simulated robot around in VC is intuitive enough for inexperienced users to immediately begin creating robot programs. When the program is completed, simply running the simulation would cause the real robot to perform the robot program.

For the thesis: The OLP connection between VC and the UR10 was found too late in the semester – just a week before the thesis deadline. Consequently, there was no time to utilize the connection to create robot programs and test them with the UR10 in the robot lab. A robot work cell was designed during the project thesis of last semester along with the full assembly process of picking and placing, applying sealant, and mounting and bolting the parts together. However, that robot program was designed for a KUKA KR 120. Although it was possible to exchange robots in VC and keep the robot program, the reach of a UR10 is far shorter than that of a KUKA KR 120, so the already-made work cell and robot program did not work. Figure 7.5 shows how the attempt to exchange the robots went.

For the bigger picture: Offline programming is nothing new. Using an OLP connection with VC to remote control a UR10 seems more rare. In Chapter 5.5, a literature review is presented of which Chapter 6.2.2 presents the results. To quickly sum up, the review on the subject shows that VC solutions are simple to grasp and intuitive. With VC’s easy to grasp GUI, the solutions can be implemented by non-experts without prior experience with the software. No other more relevant, scientific literature was found on the subject of ”offline programming with Visual Components”. Using an OLP connection with VC to remote control a UR10 seems like a special solution, but more extensive research must be performed before the rarity of the solution can be assessed.

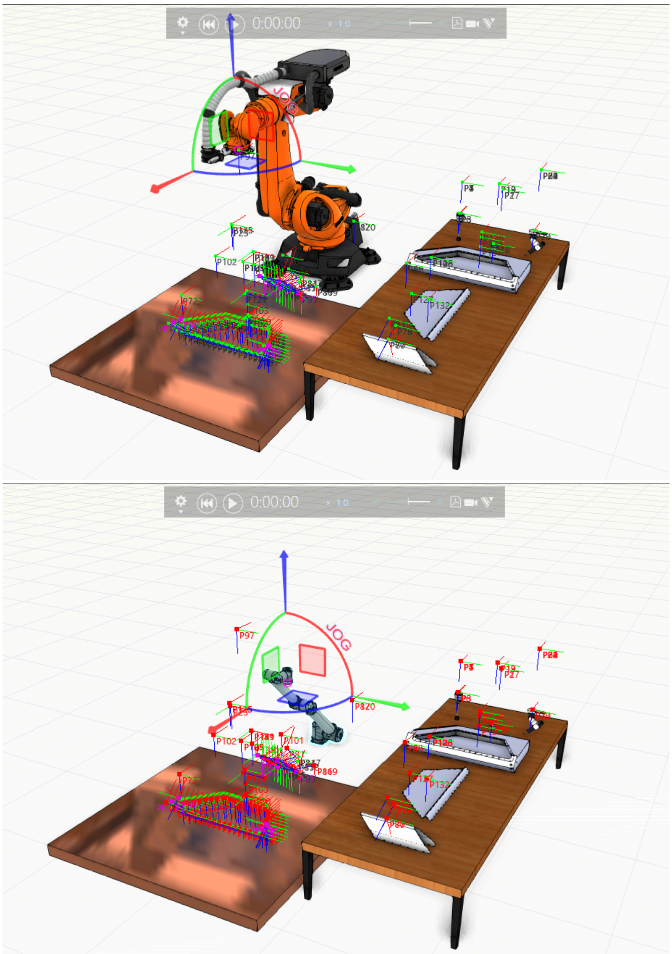


Figure 7.5: Exchanging the KUKA KR 120 with the UR10, but keeping the robot program. Notice: all robot position frames turned red; only two of the original positions were within reach of the UR10.

Having a way of controlling and monitoring industrial robots without the requirement of programming skills can be very useful. The solution found in this thesis should also be able to work with any robot with RTDE socket connection possibilities, but so far, only the case of UR10 has been tested.

For the full assembly process: For the full assembly process at KDA, the general advantages of OLP mentioned in Chapter 3.4 applies. The most important one is probably that the automatic assembly process with an industrial robot can be designed and created in VC, while the production works as normal. When the design is complete, the downtime in production is minimized to just the time it takes to change into the new, automatic assembly process performed by the industrial robot.

Chapter 8

Conclusion

The overall goal of the thesis is to make progress towards a fully automatic solution to the assembly process presented in Chapter 2. The thesis has two main parts: Developing an assembly tool used for testing automatic sealant application, and offline programming for remote control of a UR10 with Visual Components.

8.1 Conclusion of developing and testing the assembly tool

In Chapter 4 the development of the assembly tools are described. In Chapters 6.1 and 7.1 the development and test results are presented and discussed respectively. The development of the assembly tool consists of three main parts: the dual tool mount, the sealant tool, and the vacuum gripper. The overall performance of the tool is also evaluated, especially the sealant tool, which was tested in automatic sealant application tests.

8.1.1 The dual tool mount

The dual tool mount is an alternative solution to an automatic tool changer. The possibility to perform both subprocesses of the assembly with one robot was needed to make the solution automatic. The dual tool mount was therefore a mounting mechanism made for wielding both the required assembly tools to the same UR10 robot.

Parts for a dual tool mount were modeled, machined, and assembled. The resulting design worked well during the testing of the assembly tool. The welds hindered unwanted rotations, and the extender fixed the problem of the two mounted tools colliding with each others work space. Both the sealant tool and vacuum tool were able to perform their tests while mounted to the dual tool mount, without any problems caused by the mounting mechanism. The final design is shown in Figures 4.1, which is copied in Figure 8.1.

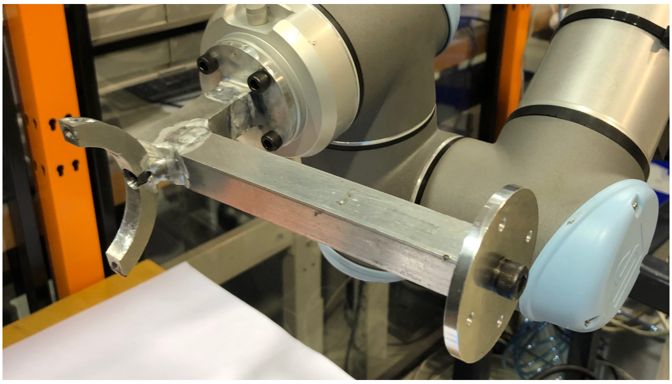


Figure 8.1: The finished dual tool mount.

The usage of the dual tool mount was discussed in a broader context in Chapter 7.1.1, where the dual tool mount was compared to other solutions such as; an automatic tool changer, multi tool mounts, and using several robots each with their own tool. It was argued that it depends on the different use case scenarios.

In the end, the dual tool mount worked just fine for the implementations in this thesis, and could potentially be used in an automatic solution of the full assembly process.

8.1.2 The sealant tool

The sealant tool was used for the sealant application of the assembly introduced in Chapter 2.1.4. The development of the tool is described in Chapter 4.2, with its results being presented in Chapter 6.1.2. The performance of the mechanical and mechatronical setups are discussed in Chapter 7.1.2, as well as the results from the sealant application tests. The goal was to make a reliable sealant tool for automatic sealant application.

Conclusion of the mechanical and mechatronical setups

The mechanical setup of the sealant tool was complete after the necessary actuator supports were created and the caulking gun was modified. Figure 8.2 shows the final result. Note that the plunger head support is not shown.

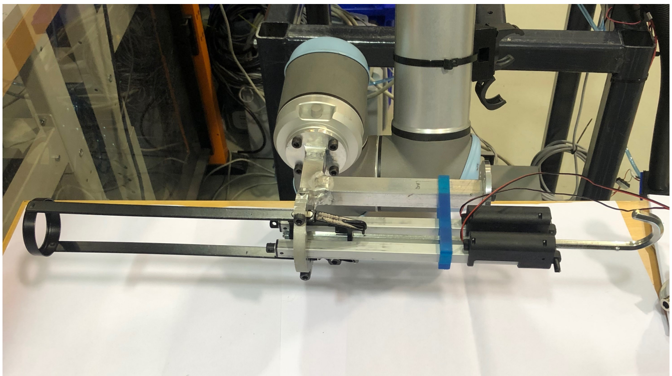


Figure 8.2: Final mechanical setup mounted on the dual tool mount.

Testing showed that the final design had two issues, both connected to third party hardware. Firstly, the sealant cartridges did not fit with the caulking gun, and secondly, the plastic tips of the linear actuators broke causing the plunger head support to fall off. The former was easily fixed, but the latter had a significant impact on the sealant application tests. The plunger head, which was attached to the plastic tips, was found essential for pushing out sealant correctly. As a result of the plunger head support falling off, one of two sealant cartridges got destroyed during the application tests. This stopped the chances of further testing, which was needed to tune the sealant tool properly.

Chapter 6.1.2 shows that the final design of the mechatronical setup was too complex. It did not allow for a fully automatic solution. In Chapter 7.1.2, it is discussed that the DPDT switch should be changed into a signal-controlled relay solution, as presented in Figure 7.2, which can also be seen in Figure 8.3. Based on the results from the leakage test, this automatic solution could have improved the overall results of the sealant application tests.

Based on the performance of the setups during testing, a redesign of the sealant tool was suggested with the correct version of the DPDT switch, to achieve an automatic sealant tool for the full assembly process. The redesign can be seen in Figures 7.1 and 8.4.

In the end, the final mechanical and mechatronical setup was not good enough to be used in an automatic solution. The sealant tool is still at an early prototyping stage.

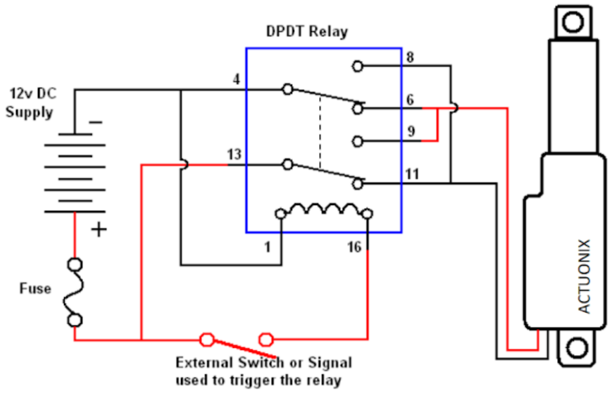


Figure 8.3: An automatic DPDT solution.

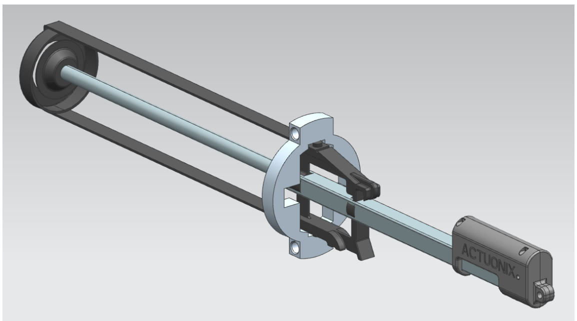


Figure 8.4: Potential redesign of the sealant tool.

Conclusion of the sealant application test results

The goals of the sealant application tests were to figure out how to tune the sealant tool to apply the correct amount of sealant and to check the reliability of the tool. Different types of sealant tests were performed, but the quantity of total tests was too low. Too many complications were encountered during the few tests to give a stable testing environment. The crucial factor was the plunger head falling off, which broke the last cartridge and put a stop to further testing. Additionally, during the tests that were carried out, the sealant did not apply as expected and conclusive results were not found; the sealant tool was deemed unreliable.

In the end, the sealant application tests were unsuccessful and proper tuning of the tool failed. More testing must be conducted to give the sealant tool more reliability.

8.1.3 The vacuum tool

The vacuum tool was used for picking and placing the aluminium parts of the experiment. The setup of the vacuum tool is described in Chapter 4.3, with its results being presented in Chapter 6.1.3. In Chapter 7.1.3, the performance of the gripper with suction cups was evaluated. During testing the assembly tool worked well. Because the original aluminium parts do not have obvious gripping possibilities for a mechanical gripper, a vacuum gripper could be a viable option for the full assembly process.

In the end, the vacuum gripper is a quick and effective solution for the pick and place subprocess and could potentially be used in an automatic solution of the full assembly process.

8.1.4 Future work on the assembly tool

To achieve an automatic solution for the assembly process, the assembly tool must be redesigned. Potential future work could be to implement the proposed redesigns of using one linear actuator and a DPDT switch that can be triggered by signals.

Although, the sealant tool was deemed unreliable, it should be possible to find an automatic solution that applies the exact amount of sealant needed to create a 360 degree squeeze-out simply by fine-tuning a finished automatic sealant tool. More sealant testing should be performed to tune the sealant tool properly.

8.2 Conclusion of the offline programming for remote control of the UR10 with Visual Components

In Chapter 3.4 it is presented that working with the OLP software VC already started during the project thesis of last semester (Autumn 2018). A natural continuation for this thesis was to establish an OLP connection between VC and the UR10 robot.

The final, successful connection was an RTDE socket connection, which enabled remote control of the UR10 via the TCP/IP socket. The connection paired simulation joint variables to general purpose input registers to send position updates from the UR10S in VC to the UR10. RTDE, TCP/IP sockets, and general purpose input registers are explained in Chapter 5.

The step-by-step procedures on how the connection can be set up are also explained, especially in Chapter 5.2.2 and 5.4. These procedures comprise of activating the RTDE server, setting up the registers, creating the RTDE Synchronization Loop, and initiating the programs in both the PolyScope controller and VC. After the total setup has been performed once and configurations are saved, the procedures to enable the connection reduces to only eight steps, which are listed in Chapter 7.2.1.

In Chapters 6.2 and 7.2, the results of successfully establishing the connection are presented and discussed, respectively. The connection resulted in a way to remote control and monitor the UR10. The most important aspect of that result is that the UR10 could be controlled almost without having to write a single line of code. Having a way of controlling and monitoring industrial robots without the requirement of programming skills can be very useful. Instead, the robot programs would be created through jogging the UR10S or building robot programs in VC. The GUI in VC makes this a more intuitive option than traditional programming, and because no programming skills are required, inexperienced users can almost immediately begin creating robot programs for the UR10.

The results of remote controlling the UR10 with VC is presented in Chapter 6.2.1 and shown in Figure 6.15, which has been added below in Figure 8.5. Videos of the remote control results are appended in the Digital appendix. The solution found in this thesis should also work with any robot that has RTDE socket connection possibilities, but so far, only the case of UR10 has been studied.

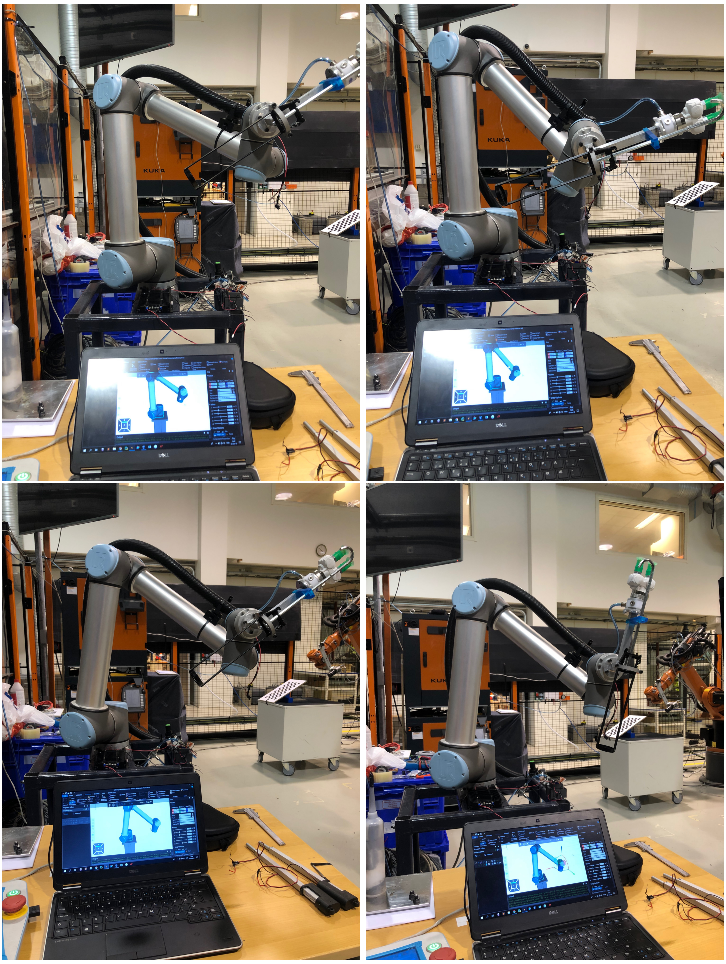


Figure 8.5: Remote controlling the UR10 with VC.

In the end, Offline programming of the UR10 with Visual Components works and allows for remote controlling and monitoring of the robot. In an intuitive way, robot programs can be created without requiring programming skills.

8.2.1 Future work on the remote control connection

The connection between Visual Components and the UR10 should be tested further. During the thesis, only jogging was tried, so the first step would be to create a full robot program of the assembly process in VC and implement it with the UR10. Testing with other industrial robots with RTDE socket connection possibilities should also be tried.

Bibliography

[1] Universal-Robots-A/S. Universal Robots UR10/CB3; 2015. User manual for robot UR10/CB3. Available from: https://s3-eu-west-1.amazonaws.com/ur-support-site/50380/UR10_User_Manual_en_Global.pdf.

[2] Vacuum Suction Cups;. Note on vacuum suction cups. Available from: https://www.schmalz.com/en/vacuum-knowledge/the-vacuum-system-and-its-components/vacuum-suction-cups/.

[3] Korane K. Vacuum cups: Key uses; 2016. Note on how vacuum cups are used. Available from: https://www.pneumatictips.com/how-are-vacuum-cups-used/.

[4] Operating principles of vacuum generation;. Note on how vacuum cups are used. Available from: https://www.schmalz.com/en/vacuum-knowledge/basic-knowledge/operating-principles-of-vacuum-generation/.

[5] Zimmer-Group. Tool Changers Series WWR;. Note on how vacuum cups are used. Available from: https://www.zimmer-group.de/en/structure/24mg5-15518564/wwr+series.

[6] Universal-Robots. SMARTSHIFT AUTOMATED TOOL CHANGER;. Note on how vacuum cups are used. Available from: https://www.universal-robots.com/plus/accessories/smartshift-automated-tool-changer/.

[7] Autonix. P16-S Linear Actuator with Limit Switches;. Note on how vacuum cups are used. Available from: https://www.actuonix.com/P16-S-Linear-Actuator-p/p16-s.htm.

[8] formlabs. Form 2;. Note on how vacuum cups are used. Available from: https://formlabs.com/eu/3d-printers/form-2/.

[9] Formlabs. An Introduction to Post-Curing SLA 3D Prints; 2017. Note on how vacuum cups are used. Available from: https://formlabs.com/blog/introduction-post-curing-sla-3d-prints/.

[10] design energy. Guide to Robot Grippers (Part 1: Vacuum Grippers);. Guide for robot gripper. Available from: https://www.designenergy.co.nz/NEWS/General/Guide+to+Robot+Grippers+Part+1+Vacuum+Grippers.html.

[11] Measuring and Control Components;. Note on how vacuum cups are used. Available from: https://www.schmalz.com/en/vacuum-knowledge/the-vacuum-system-and-its-components/switches-and-system-monitoring/measuring-and-control-components/.

[12] Solenoid Valves;. Note on how vacuum cups are used. Available from: https://www.schmalz.com/en/vacuum-knowledge/the-vacuum-system-and-its-components/valve-technology/solenoid-valves/.

[13] Rouse M. TCP (Transmission Control Protocol); 2014. Python library to control a robot from 'Universal Robots'. Available from: https://searchnetworking.techtarget.com/definition/TCP.

[14] Remote Control Via TCP/IP - 16496;. Python library to control a robot from 'Universal Robots'. Available from: https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/remote-control-via-tcpip-16496/.

[15] Real-Time Data Exchange (RTDE) Guide - 22229;. Python library to control a robot from 'Universal Robots'. Available from: https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/real-time-data-exchange-rtde-guide-22229/.

[16] SocketTest v3.0.0;. Python library to control a robot from 'Universal Robots'. Available from: http://sockettest.sourceforge.net/.

[17] Dashboard Server CB-Series, port 29999 - 15690;. Python library to control a robot from 'Universal Robots'. Available from: https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/dashboard-server-cb-series-port-29999-15690/.

[18] Testing TCP connection Host to UR;. Python library to control a robot from 'Universal Robots'. Available from: http://www.zacobria.com/universal-robots-knowledge-base-tech-support-forum-hints-tips/knowledge-base/testing-tcp-connection-host-to-ur/.

[19] UR Download;. Python library to control a robot from 'Universal Robots'. Available from: https://www.universal-robots.com/download/?option=51823#section16597.

[20] Vollmann K. A new approach to robot simulation tools with parametric components. In: 2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02.. vol. 2; 2002. p. 881–885 vol.2.

[21] Sett A, Vollmann K. Computer based robot training in a virtual environment. In: 2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02.. vol. 2; 2002. p. 1185–1189 vol.2.

[22] Piesk S, Kaarela J, Mkel J. Simulation and programming experiences of collaborative robots for small-scale manufacturing. In: 2018 2nd International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS); 2018. p. 1–4.

Thomas Fredrik Bech Aschehoug

Automatic assembly of aircraft parts with automatic sealant inspection and OLP remote control of industrial robot

NTNU

Kunnskap for en bedre verden