



Norwegian University of
Science and Technology

Robotic Multiple-Pass Welding of V-Groove Butt Joints

Bjørn Emil Evensen

Mechanical Engineering

Submission date: June 2016

Supervisor: Olav Egeland, IPK

Norwegian University of Science and Technology
Department of Production and Quality Engineering

MASTEROPPGAVE 2016

Bjørn Emil Evensen

Tittel: Robotisert flerlags sveising av V-fuge buttseiser

Tittel (engelsk): Robotic Multiple-Pass Welding of V-groove Butt Joints

Oppgavens tekst:

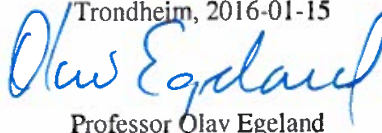
I offshorekonstruksjoner brukes i stor grad flerlags sveising for å fylle store sveisefuger. Ved robotisert sveising med flere lag er det behov for å planlegge strukturen for antall sveiselag og pass basert på geometrien til fugen. I denne oppgaven skal offline programmering av flerlags sveising studeres og en prosedyre for automatisk generering av sveiseprogram for V-fuger utvikles. Systemet skal prøves ut i instituttets robotlaboratorium.

1. Gi en kort beskrivelse av flerlags sveising for store offshorekonstruksjoner.
2. Beskriv metoder for automatisert flerlags sveiseplanlegging.
3. Lag en parameterstyrt prosedyre som genererer et robotprogram for flerlags sveising av V-fuge buttseiser med ulik geometri. Dette inkluderer robotprogrammering og simulering i Visual Components og nedlasting av program til robotsystemet.
4. Prøv ut systemet i eksperimenter.

Oppgave utlevert: 2016-01-15

Innlevering: 2016-06-10

Utført ved Institutt for produksjons- og kvalitetsteknikk

Trondheim, 2016-01-15

Professor Olav Egeland
Faglærer

Preface

This work is the concluding Master's thesis at the study programme in Mechanical Engineering at NTNU. The work was carried out between January and June 2016.

Since I started specializing my education towards manufacturing and automation, I have been fascinated by the world of robotics. During the work on my project thesis last semester, I visited the Kvaerner construction site in Verdal where I got an introduction to jacket fabrication. During the visit, we discussed welding of the large joints in these structures and the problems related to automating these processes. The discussion inspired me to look more closely at this issue and resulted in the topic of this thesis.

I have no background in welding and had little prior knowledge about the robot-welder interface. The knowledge I have gained by trying and failing in experiments during the course of this thesis have been enormous. In the end, it has led to a deeper understanding of automated welding and a successful laboratory implementation.

There are several people who have been valuable for the completion of this thesis. First of all, I would like to thank my supervisor Olav Egeland for giving me the opportunity to work on such an interesting and applicable topic. Working on both a theoretical and practical thesis, with the implementation of an experiment in the robotics laboratory at IPK has been motivating and educational. I extend my gratitude to Eirik Njåstad for his useful comments, remarks and for our giving discussions throughout this thesis. His assistance in the laboratory implementation has also been appreciated. Furthermore, the workshop employees under the leadership of Arild Sæther have been invaluable assets as they have manufactured all test parts, created a fixture and helped me process the welded joints. Also, Ketil Børmark from Tromek AS have contributed to this thesis by evaluating some of my final test pieces with their welding expertise. Lastly, I thank Arve Sørgerd who has been my contact at Kvaerner.

Trondheim, June 10, 2016

A handwritten signature in black ink, reading "Bjørn Emil Evensen". The signature is written in a cursive style with a long horizontal stroke at the end.

Bjørn Emil Evensen

Summary

Intelligent and automated production have become an inevitable trend for manufacturing companies in developed countries. Robotic welding is a key component in order to accomplish a competitive and effective production. Successful implementation of robotic welding is especially seen throughout the automobile industry. However, other industries are still on the verge of utilizing robotic welding. One of them is the offshore sector. Offshore structures are large constructions welded together by minor components. An example of this is ships, assembled by thick plates welded together end-to-end. To ensure sufficient strength in the welded joints, full penetration welds are needed and a multiple-pass weld sequence is required. Multiple-pass welding is generally done manually as the process is complex and difficult to automate.

This thesis presents an automated solution for multiple-pass welding of thick V-groove butt joints with variable geometry. This welding configuration is common in shipbuilding and in fabrication of high pressure vessels. Previous research on automated multiple-pass planning have been reviewed and used as a basis for the solution presented in this thesis. A procedure that determines a welding sequence containing the number of layers and passes based on geometrical properties of a butt joint, is created using different programming and simulation software. Weld beads are approximated as sets of parallelograms and trapeziums. Algorithms in this procedure calculates the position and pose of the welding torch, amplitude of weaving and the number of passes for each layer based on a set of parameters and criteria. These are welding speed, filler wire feed speed, maximum amplitude of weaving and maximum allowed bead height. A robot program is then created for execution of the welding sequence.

The procedure have been implemented in the robotics laboratory at the Institute of Production and Quality Engineering, NTNU. This implementation showed the effectiveness of the automated procedure by successfully welding different V-groove butt joints. A set of preliminary tests were conducted to determine viable combinations of welding parameters. Continuous testing was then carried out to adjust the deposition efficiency coefficient and the weaving factor. In the end, conformity between the procedure and the experimental results was obtained. The results from all welding experiments are presented in the thesis.

Sammendrag

Intelligent og automatisert produksjon har blitt en unngåelig trend for bedrifter i industrilandene. I den forbindelse har robotisert sveising markert seg som en viktig komponent for å oppnå en konkurransedyktig og effektiv produksjon. En av de mest vellykkede implementeringer av robotsveising er i bilindustrien hvor store deler av produksjonslinjene er automatisert. Andre sektorer har ikke utnyttet potensialet i robotisert sveising i samme grad. Dette er spesielt gjeldende for offshoreindustrien. Offshore-konstruksjoner er store byggverk sveiset sammen av mindre komponenter. Et eksempel på dette er fabrikasjon av skip hvor skroget er sammensatt av plater sveiset sammen ende mot ende. Full sveisegjennomføring er viktig for å sikre tilstrekkelig styrke i sveiseskjøtene, og dette er ofte kun gjennomførbart med flerlags sveis. Flerlags sveising er vanligvis gjennomført manuelt ettersom prosessen er kompleks og vanskelig å automatisere.

Denne avhandlingen presenterer en automatisert løsning for flerlags sveising av tykke V-fuge buttssveiser med variabel geometri. Denne sveisekonfigurasjonen er svært vanlig innen skipsbygging og fabrikasjon av trykkbeholdere. Tidligere forskning på automatisert flerlags sveiseplanlegging er gjennomgått og brukt som grunnlag for løsning i denne avhandlingen. En automatisert prosedyre som generer en sveisesekvens inneholdende antall nødvendige lag og sveisepass basert på de geometriske egenskapene til V-fugen er utviklet. Dette er gjort ved hjelp av ulike databaserte programmerings- og simuleringstøytøy. Tverrsnittet av sveisstrengene er approksimert i form av parallellogrammer og trapeser. Algoritmer i denne prosedyren beregner posisjon og orientering av sveispistolen, samt pendlingsamplitude og antall nødvendige pass for hvert sveiselag basert på et sett med parametere og kriterier. Disse er sveisehastighet, trådmatingshastighet, maksimal amplitude for pendling og maksimalt tillatt høyde for sveisestrengen. Et robotprogram blir så generert for utførelse av sveisesekvensen.

Proseduren har blitt implementert i robotlaboratoriet ved Institutt for produksjons- og kvalitetsteknikk, NTNU. Eksperimenter med sveising av ulike geometrier for V-fuge buttssveiser viste effektiviteten av den automatiserte prosedyren. Et sett av innledende tester ble utført for å bestemme fungerende kombinasjoner av sveiseparametere. Videre ble testing gjennomført for å regulere avsetningskoeffisienten og pendlefaktoren. Samsvar mellom de modellerte buttssveisene og eksperimentelle resultater ble påvist, og en velfungerende automatisert prosedyre oppnådd. Resultatene fra alle forsøkene er presentert i denne avhandlingen.

Contents

Preface	v
Summary	vii
Sammendrag	ix
Table of Contents	xii
List of Tables	xiii
List of Figures	xvi
Abbreviations	xvii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives	2
1.3 Approach	2
1.4 Structure of the Report	4
2 Background Theory	5
2.1 Introduction	5
2.2 Welding Terminology	6
2.3 Arc Welding Technology	12
2.4 Offshore Construction Welding	17
2.5 Automated Multiple-Pass Planning	21
3 Solution	31
3.1 Introduction	31
3.2 Equations	32
3.3 Multiple-Pass Sequencing Script	36
3.4 Visualization and Simulation	39

4	Laboratory Implementation and Testing	45
4.1	Introduction	45
4.2	Robot Cell	46
4.3	Fronius Welding System	46
4.4	Software	48
4.5	Overall Information Flow	51
4.6	Experimental Setup	52
4.7	Test Plan	54
5	Results & Analysis	55
5.1	Preliminary Tests	56
5.2	Butt Joint Type I	61
5.3	Butt Joint Type II	64
5.4	Butt Joint Type III	68
5.5	Butt Joint Type IV	73
5.6	Final Tests	77
5.7	Third Party Visual Inspection	82
5.8	Remarks	83
6	Conclusion and Future Work	85
	Bibliography	87
	Appendices	91
A	MATLAB Script: <i>sequencer.m</i>	93
B	Python Script: <i>jointmodel.py</i>	98
C	Python Script: <i>sequencer.py</i>	103
D	Python Script: <i>robotmotion.py</i>	108
E	Digital Appendix	114

List of Tables

2.1	Advantages/disadvantages of welding processes using consumable electrodes, Bowditch (2005).	14
3.1	Welding equation notation.	32
3.2	Parameter table output from <i>sequencer.m</i> example.	36
3.3	MATLAB: Script, functions and corresponding equations	37
4.1	Butt joint types geometrical data	53
5.1	Butt joint type I: 3 layer 4 pass test parameters for $\delta = 2.0$	62
5.2	Butt joint type I: 3 layer 4 pass test parameters for $\delta = 1.5$	62
5.3	Butt joint type I: Adjusted 3 layer 4 passes test parameters for $\delta = 2.0$. . .	63
5.4	Butt joint type II: 2-1 test parameters	65
5.5	Butt joint type II: 2-2 test parameters	66
5.6	Butt joint type II: 2-3 test parameters	66
5.7	Butt joint type III: 3-1 test parameters	68
5.8	Butt joint type III: 3-2 test parameters	69
5.9	Butt joint type IV: 4-1 test parameters	74
5.10	Butt joint type IV: 4-2 test parameters	74
5.11	Butt joint type II: Final test parameters	77
5.12	Butt joint type III: Final test parameters	78
5.13	Butt joint type IV: Final test parameters	79

List of Figures

2.1	V-groove butt joint terminology	6
2.2	Weaving width, weaving amplitude, ISO (1998)	7
2.3	Weld bead terminology	8
2.4	Multiple-pass layer terminology	8
2.5	Torch angle positions	9
2.6	Weld defects	10
2.7	Schematic representation of a conventional GMAW process, Pires et al. (2006)	13
2.8	Metal transfer modes. From left to right: Short circuiting - globular - spray	15
2.9	Characteristics between droplet diameter and current for a 0.035in (0.89mm) diameter electrode, Soderstrom and Mendez (2008)	15
2.10	Jacket <i>Gudrun</i> load-out, Kvaerner (2015)	17
2.11	Jacket fabrication	19
2.12	Fish farming cage, Ocean Farming (2015)	20
2.13	Actual bead shape versus simplified bead shape, Yang et al. (2014)	21
2.14	Build-up method and welding sequence for horizontal fillet welding, Moriyasu et al. (1993)	21
2.15	Welding sequence of single-sided V-groove, Zhang et al. (2009)	22
2.16	Multiple-pass fill layout of V-groove, Zhang et al. (2009)	23
2.17	Reference path selection, Zhang et al. (2011)	24
2.18	Multiple-pass welding experiment, Zhang et al. (2011)	25
2.19	Multiple-pass welding experiment, Yang et al. (2014)	26
2.20	Schematic diagram of the groove, Huang et al. (2015)	27
2.21	Multiple-pass path planning, Huang et al. (2015)	28
2.22	Multiple-pass welding experiment, Huang et al. (2015).	29
3.1	Section shape of multiple pass welding seam, Zhang et al. (2011)	33
3.2	Weld bead geometry, Zhang et al. (2011)	34
3.3	Weaving width of torch, Zhang et al. (2011)	35
3.4	Visual output from <i>sequencer.m</i> example	36

3.5	<i>sequencer:m</i> information and work flow	38
3.6	Parametric joint model and GUI in 3DAutomate	39
3.7	Base shift and rotation	41
3.8	3DAutomate information and work flow	43
4.1	Fronius welding system components, Fronius International GmbH (2010)	46
4.2	KUKA.Sim and KUKA.OfficeLite	48
4.3	Weave patterns in ArcTech	49
4.4	Weave frequency diagram, KUKA Roboter GmbH (2012)	50
4.5	Overall information flow	51
4.6	Laboratory test setup	52
4.7	Butt joints test specimen overview	53
5.1	Preliminary plate tests: Ignition wait time	56
5.2	Preliminary plate tests: End crater time	57
5.3	Preliminary plate tests: Weave pattern	58
5.4	Preliminary plate tests: Various frequencies and amplitude	59
5.5	Preliminary plate tests: Various amplitude at 1 Hz triangle weave	59
5.6	Preliminary plate tests: Feed speed	60
5.7	Preliminary plate tests: Cover pass	60
5.8	Butt joint type I: Schematic layout	61
5.9	Butt joint type I: Test results	62
5.10	Butt joint type I: Miscellaneous passes	63
5.11	Butt joint type II: Schematic layout	64
5.12	Butt joint type II: 2-1 root pass	65
5.13	Butt joint type II: Test results	67
5.14	Butt joint type III: Schematic layout	68
5.15	Butt joint type III: 3-1 weaving configuration	69
5.16	Butt joint type III: Test results	70
5.17	Butt joint type III: 3-1 and 3-2 weld face	71
5.18	Butt joint type III: 3-1 weld pass progression	72
5.19	Butt joint type IV: Schematic layout	73
5.20	Butt joint type IV: Test results	75
5.21	Butt joint type IV: 4-1 and 4-2 weld face	76
5.22	Final test weld faces	80
5.23	Final test results	81
5.24	Backing plate	82

Abbreviations

ASME	=	American Society of Mechanical Engineers
ATRAW	=	Asymmetrical Trapezoid Weaving
AWS	=	American Welding Society
CAD	=	Computer-Aided Design
DCEN	=	Direct Current Electrode Negative
DCEP	=	Direct Current Electrode Positive
FCAW	=	Flux Cored Arc Welding
FCAW-S	=	Flux Cored Arc Welding, Self-Shielding
FPSO	=	Floating Production Storage and Offloading
GMAW	=	Gas Metal Arc Welding
GTAW	=	Gas Tungsten Arc Welding
GUI	=	Graphical User Interface
KCP	=	KUKA Control Panel
KRL	=	KUKA Robot Language
KSS	=	KUKA System Software
MAG	=	Metal Active Gas
MIG	=	Metal Inert Gas
NDT	=	Non-Destructive Testing
PLC	=	Programmable Logic Solver
RSL	=	Robot Scripting Language
SAW	=	Submerged Arc Welding
SMAW	=	Shielded Metal Arc Welding
SPIW	=	Spiral Weaving
SW	=	Spiral Weaving
TCP	=	Tool Center Point
TRAW	=	Trapezoid Weaving
TRIW	=	Triangle Weaving
VRC	=	Virtual Robot Controller
WPS	=	Welding Procedure Specification

Introduction

1.1 Background and Motivation

Industry in developed countries face two challenges today: better quality at lower prices and the need to improve productivity in order to meet the competition from the low salary regions of the world. Manufacturing companies need to respond quickly to market requirements to keep their products competitive.

The industrial robot have become a vital component for the realization of automated manufacturing. Robot manipulators offer capabilities that make them ideal for automation. They can perform repetitive task at low cost and satisfactory quality. Robotic welding is one of the most successful applications of the industrial robot. Even though robotic welding systems are far from able to replace a skilled welder, the technology have seen remarkable developments over the last couple of decades. It has been widely implemented in production processes, especially within the automobile industry. However, it is still scratching the surface in sectors such as the offshore industry. Offshore industry welding is in general done manually to this date.

The motivation behind this thesis came from offshore jacket manufacturing. These large structures contain numerous joints that are manually welded. Most of the joints are complex configurations created by multiple intersecting pipes. Welding these structures in accordance to offshore manufacturing standards require skilled welders. Because of the shear size of the joints, a multiple-pass welding sequence is needed to fill the grooves. Welding the joints become time consuming and tiresome as some of the grooves are so large they require several hundred weld passes. Automating certain parts of this process would most definitely improve productivity and competitiveness.

One condition for automating multiple-pass welding without the use of real time feedback is a mathematical algorithm that determines how many weld passes that are needed and the position for each weld pass. This have to be based on the geometry of the joint and various welding parameters. In this thesis, the problem is simplified to robotized multiple-pass welding of thick plates. More specifically butt joints with a V-groove edge preparation. This welding configuration is especially found in shipbuilding where large

plates are welded together end-to-end. Although jacket structures features more complex weld geometries, it is important to first establish a working model for a basic joint. To achieve this, an automatic path layout algorithm have to be developed, translated into motion statements and implemented to a robotic welding system.

1.2 Objectives

The main objective of this thesis is to establish an automated system that creates and executes a multiple-pass robotic welding program based on variable geometries for V-groove butt joints. This includes:

1. Providing a brief explanation of multiple-pass welding of large offshore structures.
2. Reviewing previous work on automated multiple-pass planning.
3. Creating a parameter-controlled procedure that generates a robot program for multiple-pass welding of V-groove butt joints with variable geometry. This includes robot programming and simulation in *Visual Components* and downloading the program to the robot system.
4. Implementation of the procedure in the robot laboratory at *Department of Production and Quality Engineering (IPK), NTNU*.

1.3 Approach

The objectives in this thesis have been both of theoretical and practical nature. Literature and articles have been reviewed in order to answer the theoretical aspects, and also to prepare for the experimental implementation. The approach to each specific objective is given here.

Objective 1

To provide a brief and understandable explanation of the multiple-pass welding of large offshore structures, and also for items given in objective 2, it is essential to establish some basic theory and terminology for arc welding. This is given in section 2.2 and 2.3. A few examples of different offshore structures and their welding processes is given in section 2.4.

Objective 2

Relevant research and experiments conducted related to multiple-pass planning is given in section 2.5. The different approaches are discussed and compared. Findings in this section will constitute the groundwork for a new solution.

Objective 3

The procedure is created using different software. *MATLAB* was first used to generate a multiple-pass sequencing script and a visual model. This is outlined in section 3.3. Equations used for the sequencing is given in 3.2. The script is implemented into *Visual Components* for simulation and robot motion statement generation purposes. All work related to Visual Components is given in 3.4. Translation of the robot motion statement program to robot language is done with *KUKA.Sim* and *KUKA.OfficeLite*, outlined in section 4.4. Overall information flow and how the program is downloaded to the robot system is given in section 4.5.

Objective 4

This thesis has culminated into a welding experiment. The procedure created under objective 3 is deployed and tested in the robot laboratory to determine the effectiveness of it. How this implementation was carried out is stated in chapter 4. The laboratory setup is given in section 4.6, with used equipment in section 4.3. The results and adjustments of the procedure according to experimental results is given in chapter 5.

Literature

Most of the literature used in this thesis is related to automated multiple-pass planning for thick plates. During the literature review of multiple-pass planning, a variety of different and unique approaches was found. Even though principles taken from various studies has been implemented, there were a couple of articles that constitutes the basis for the solution presented in this thesis. These are:

- *Robot Path Planning in Multi-Pass Weaving Welding for Thick Plates* by Zhang et al.
- *Multi-pass Path Planning for Thick Plate by DSAW Based on Vision Sensor* by Yang et al.

There are an enormous amount of knowledge related to welding technology available. It has been important to provide an introduction to arc welding technology and establish relevant parameters in order to analyze the results from conducted experiments. The challenge has been to select the relevant information from credible sources. The main background on welding technology is therefore based on work published by trusted welding institutions or acknowledged welding technology providers. This includes:

- *Welding Technology Fundamentals* by Bowditch
- *Welding Handbook: Welding Science & Technology* by AWS
- *Basic Welding Filler Metal Technology: Lesson II - Common Electric Arc Welding Processes* by ESAB

Additionally, documentation from KUKA for the KR C4 controller, ArcTech, DeviceNet and other KUKA specific manuals as well as Fronius documentation for the welding system has been used in the experimental implementation.

1.4 Structure of the Report

The report is structured the following way:

- Chapter 2 provides an overview over important welding terminology and arc welding technologies. Special attention is paid to gas metal arc welding as this is the welding technology used in the laboratory implementation. Offshore industry specific practices and challenges in regard to multiple-pass welding is briefly presented and explained. The relevant research and development within automated multiple-pass planning and welding is also presented.
- Chapter 3 outline the solution for the parameter-controlled procedure. The chapter explains the logic and functionality of the process steps as well as code for path planning, visualization of sequencing and simulation of the welding process.
- Chapter 4 introduce the laboratory implementation of this thesis. Information on the used equipment, that is software and hardware, in conjunction with the experiments is given. This also includes translation of the solution from chapter 3 into robot language, information flow and experimental setup. Lastly, Chapter 4 outline a plan for the welding.
- Chapter 5 contain all the results from the laboratory implementation. Results are analyzed and discussed in a sequenced manner. Adjustments are made and explained in progression with the obtained experimental results.
- Chapter 6 summarize the work carried out and important results of the developed system. Recommendations for future work are also given.
- Important source code for different software is included in the appendix. Additionally, the contents of the digital appendix is provided.

Background Theory

2.1 Introduction

This chapter outlines the theoretical background material for the solution presented later in this thesis. The chapter introduces both well establish material and some of the latest research on the respective areas.

It is important to have an basic understanding of welding methods and terminology in order to completely comprehend this thesis. Section 2.2 and 2.3 will outline the basic terminology and give a introduction to the world of arc welding. The objective of is, as previously stated, multiple-pass welding of butt joint with V-groove edge preparation. Hence, special attention is paid to welding terminology concerning butt joints. Furthermore, only arc welding technology is explained with focus on gas metal arc welding.

Section 2.4 reviews offshore construction welding. As the market progressively gets more competitive, the need for automated solutions rises. Practices and challenges concerning automated welding in the offshore industry are discussed in this section.

The last section are dedicated to reviewing relevant research and experiments on multiple-pass planning. This includes sequencing, robotic implementation and execution.

2.2 Welding Terminology

There are a lot of technical terms related to welding. From a welding technical point of view, this thesis will only scratch the surface of the complex process. Nonetheless, a basic set of terms will be introduced in this section and used throughout the thesis. The terminology is taken from *Welding Technology Fundamentals* by Bowditch (2005) which is a standard terminology approved by The American Welding Society (AWS) and the International Organization for Standardization (ISO).

Joint Geometry

The joint geometry is defined by the AWS as "the shape and dimension of a joint, in cross section, prior to welding." This geometry is usually determined by a welding engineer or a designer based on the requirements of the part. There are five basic types of joints: butt-, lap-, corner-, T- and edge joint. As this thesis focus on welding of thick butt joints, only the geometry parameters for this type is given. Butt joints are used when parts are joined edge-to-edge and is common within pipeline construction and shipbuilding. Here, pipes or plates are welding together end-to-end to form long pipelines are strong ship hulls. Generally, a welded joint should be at least as strong as the *base metal*. A deep penetration is needed to accomplish this. In order to achieve a *full penetration weld* on thick pieces, the base metal are often subjected to some sort of edge preparation. A full penetration weld occurs when the weld penetrates through the entire thickness of the base metal. Edge preparation entails machining or cutting the surface enabling the welder to reach near the bottom of the joint and to provide sufficient space for the torch. Common edge preparations include J-, U- and V-grooves. The focus will specifically be on welding the V-groove. A drawing of a V-groove with associated characteristics is given in figure 2.1. The drawing show the *bevel angle*, *groove angle*, *groove face*, *root face*, *root gap* and the *depth* of the bevel.

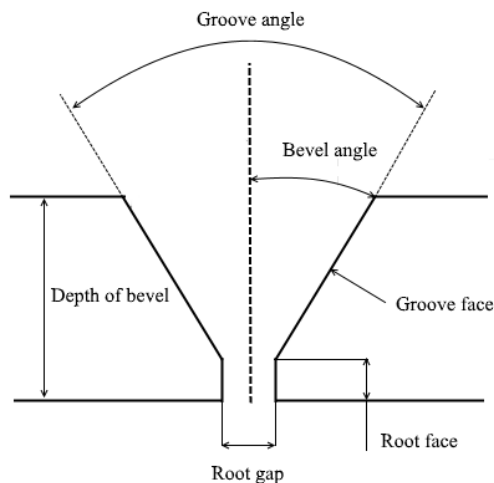


Figure 2.1: V-groove butt joint terminology

Weld Beads

A *Weld bead* occur in the form of stringer- and weave beads. The bead is the filler metal that is added to a joint after each time a *weld pass* is made. Weave beads are used to create a wider weld pool by oscillating the torch in a side-to-side motion along the weld joint. Stringer beads are made without any side-to-side motion and are used when a standard bead width is acceptable. Various movement patterns are utilized to make a weave bead. Examples of patterns are given in section 4.4. The resulting height of a bead is correlated with the amplitude of the weaving. A larger amplitude creates a lower bead and vice versa. Figure 2.2 show the relationship between the weaving width and weaving amplitude in a weave bead. The weld bead should in general not be thicker than 6.4 millimeters due to enhanced risk for weld defects, Bowditch (2005). Additionally, bead width is seldom greater than 19-25 mm. Restrictions on weaving width is often set by standards in different industries, or by the companies applicable welding procedure specification (WPS).

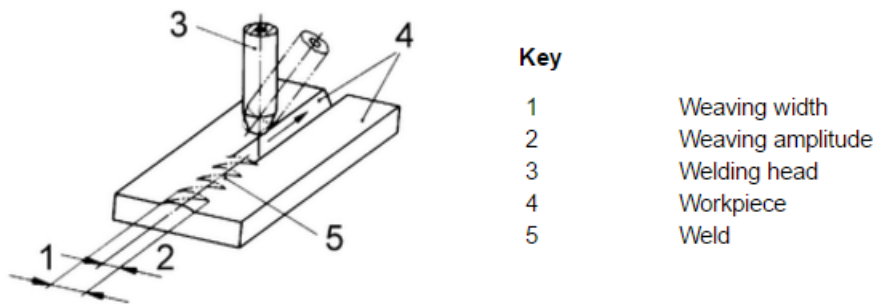


Figure 2.2: Weaving width, weaving amplitude, ISO (1998)

A groove weld have a set of important characteristics that are used to evaluate the bead. As shown in figure 2.3, the *weld face* is the outer surface of the weld bead which stretches from *weld toe* to *weld toe*. A convex weld face is desired for butt joints. The distance from the top of the weld face to the surface of the base metal is called the *face reinforcement*. *Weld size* refers to the depth that a weld extends into the joint from the surface. The *root reinforcement* signifies the penetration of the weld at the root side of the joint. *Backing* can be used to control the penetration, thus the root reinforcement. Backings are metal strips attached to the base metal. While most backing strips are removed after welding, some become part of the completed joint. Removable backings are made in a material that is not melted by the respective welding method. Backings that allow a small root reinforcement from the weld beads are most commonly used as this provides the strongest joint.

Multiple-pass Terminology

In the world of multiple-pass welding there are a few terms that should be known. Multiple-pass welding is a standard method when welding a beveled groove or other large joints. One weld pass may not be sufficient to achieve full penetration and ensure a strong joint,

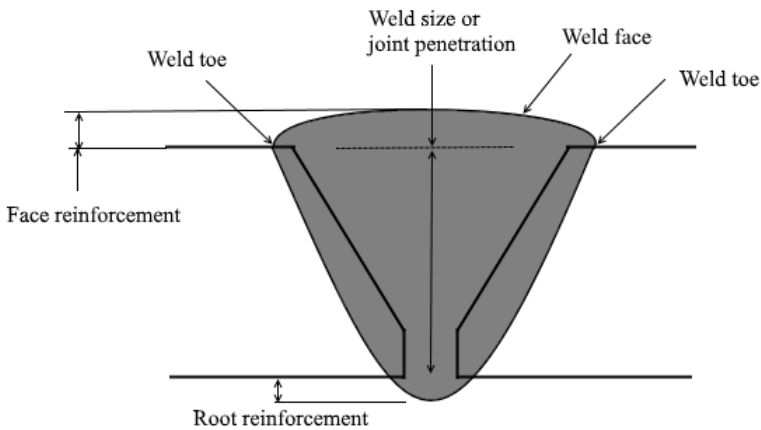


Figure 2.3: Weld bead terminology

therefore several weld beads are laid in a structure. This structure usually contain three types of passes. The *root pass* is the first and most important pass. A full penetration weld can only occur on this pass. Intermediate weld passes are called *filler passes* and can range from one to several hundred passes depending on the joint size. The final pass is the *cover pass*. This pass must be made with a weaving motion which tie in all filler passes to form a single, wide convex weld face. Figure 2.4 show the different passes. In some cases, a *hot pass* is laid after the root pass. This is especially known within pipeline welding. It is done as a method to eliminate any undercutting that may have occurred in the root pass. Undercut is a depression at the weld toe area indicating that the weld piece has been melted but not filled with filler metal. This weld defect is explained later in this section. The hot pass uses a higher amperage and faster travel rate to reshape the weld bead and burn out trapped slag, Jeffus (1999). A multiple-pass weld can be done with both stringer beads and weave beads.

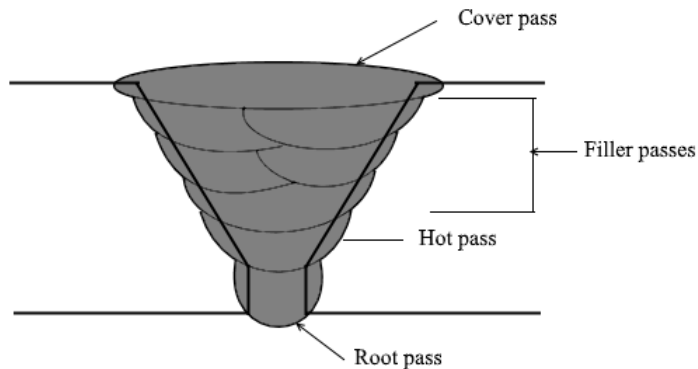


Figure 2.4: Multiple-pass layer terminology

Torch- and Welding Positions

There are two angles of importance concerning the torch position in welding. This is the *work angle* and *travel angle*. The work angle is the angle between the perpendicular line of the base metal surface and a plane determined by the weld axis and torch axis. A torch that is perpendicular to the surface of a butt weld has a zero degree work angle. The travel angle is the angle between the line perpendicular to the weld axis and the axis of the electrode. Three different welding methods are given depending on the direction and size of the travel angle. Different angles will have significant effects on the weld bead shape, penetration and efficiency. The *backhand welding* method pulls or drags the weld pool along the weld axis, producing a narrow weld bead with a deep penetration. When a backhand welding method is used, the travel angle is referred to as the *drag angle*. A zero degree travel angle corresponds to the perpendicular position, producing a medium-width, medium penetration weld bead. The last method is the *forehand welding* position. In this position, the weld pool is pushed in front the welding torch, producing a wide but shallow weld bead. The travel angle is here referred to as the *push angle*. Both the push- and drag angle normally vary between 20-25 degrees.

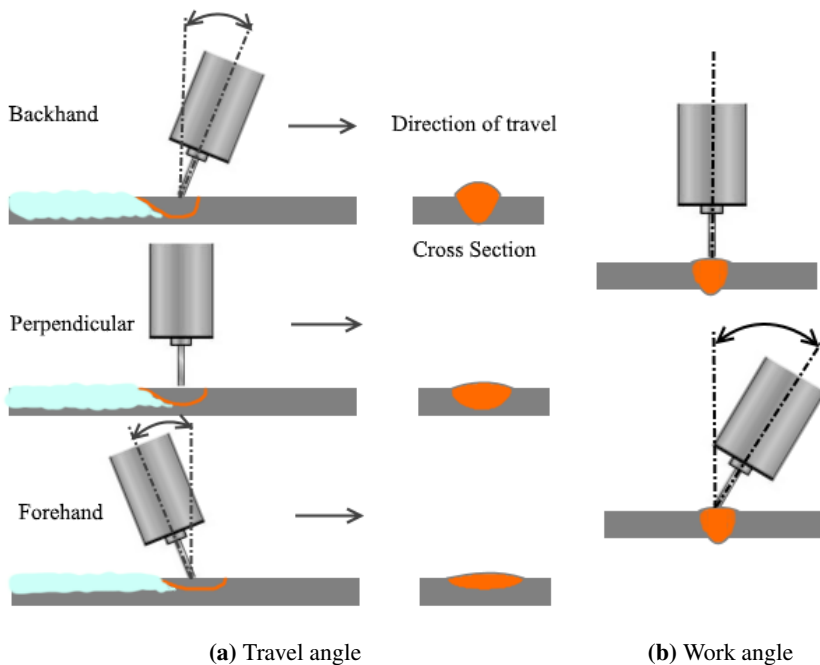


Figure 2.5: Torch angle positions

The groove weld orientation in space is described by positional codes according to different standards. The four main positions is the flat, horizontal, vertical and the overhead position. According to the standard developed by the American Society of Mechanical

Engineers (ASME), the positions are denoted 1G, 2G, 3G and 4G, ASME (2010). The corresponding ISO standard denotes the same positions as PA, PC, PG/PF and PE, ISO (2011). ISO distinguish between downhill (PG) and uphill (PF) welding in the corresponding 3G-position. Other orientation codes include fillet welds and pipeline welds in different orientations. As most of the positions are irrelevant for this thesis, they will not be explained further here.

Weld Defects

Defects cause the welded joint to be weaker than its design requirements. Common weld defects in arc welding include *incomplete penetration*, *lack of fusion*, *slag inclusions*, *undercut*, *overlap* and *porosity*. Defects usually occurs due to an improper welding procedure and are easily corrected by the operator once the causes are detected. Figure 2.6 display some of the different defects listed.

Incomplete penetration appear when the weld bead does not penetrate the entire thickness of the base metal, or two opposing weld beads do not interpenetrate. Welding current has the greatest effect on penetration. The defect are usually caused by a too low welding current and can be easily avoided by using a higher amperage. Other causes may be a too low travel speed or an incorrect torch angle. Both can create a protective cushion of the molten weld metal which roll in front of the arc, preventing penetration.

Lack of fusion is a defect where there is no fusion between the base metal and the filler metal. Similar to incomplete penetration, this can be caused by a protective cushion of weld metal due to a too low travel speed. Another common cause is found in welding of wide joints. If the arc is directed downward to the center of the joint, the molten metal may only flow to the sidewalls without actually melting the base metal and cause fusion. Large welds bridging wide gaps should therefore be avoided. A split bead technique is recommended in multiple-pass welding whenever possible to prevent this defect. Lack of fusion is also referred to as *cold lapping*.

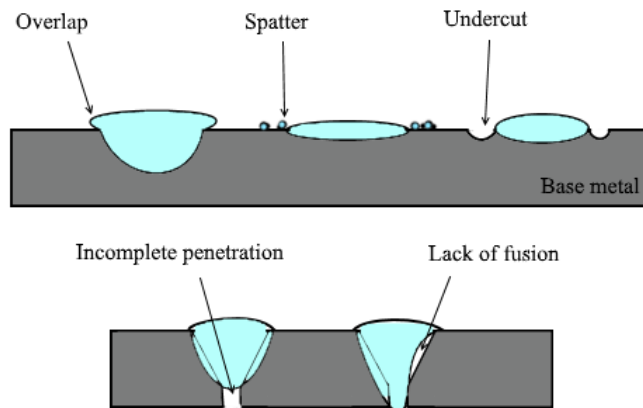


Figure 2.6: Weld defects

Slag inclusions are a weld defect where slag is entrapped in the weld metal before it

can float to the surface. Slag is the brittle mass that forms over the weld bead. Slag do not contribute to the strength of the weld or protection of the metal after the material is welded. It has to removed before other potential passes are made, or the weld is to be inspected or coated with a surface finish. Only certain types of welding processes produce slag.

Undercutting is a depression at the weld toe indicating that the weld metal is below the level of the base metal. When the travel speed is too high, the rapid solidification of the weld metal produces surface tension. These forces draws the molten filler metal along the edges of the weld bead inwards, causing it to pile up along the center. The same effect is found on the base metal. Ultimately, a depression or a undercut groove is left because the melted base material has been drawn into the weld and not allowed to wet back properly. Lowering travel speed will reduce undercutting.

Overlap is a condition in which the weld pool flows onto the base metal surface and is not fused into the base metal.

Porosity are essentially gas pores found in the solidified weld bead. They are most often caused by atmospheric contamination or the presence of foreign matter.

Though it is not directly a weld defect, *spatter* is also mentioned here. Spatter is the scattering of molten metal droplets over the surface near an arc weld. The production of spatter is a good indicator of right electrode extension, or wire stick out, which is one of the primary causes for it, alongside an incorrect work angle. Lengthening the electrode extension will reduce spatter.

Lastly, *distortion* is also a defect which can alter the position of the base metal pieces in relation to each other. Any process that uses a localized heat source, such as the arc, will likely have distortion associated with it. It can be minimized by preheating the base metal, lowering heat input during the weld pass, or avoid excessive weld bead sizes. The distortion can also be accounted for by presetting the components in a position to offset the distortion. Using fixtures and jigs can prevent distortion but will most likely result in tension in the base metal.

Deposition Efficiency

All arc welding processes experience loss of material during operation in form of spatter or other weld defects. The loss is described in terms of nominal electrode efficiency, effective electrode efficiency, deposition efficiency, overall metal recovery and deposition coefficient, ISO (1972). Collectively, this thesis summarize the loss under the term *deposition efficiency*. The deposition efficiency can be calculated by the following formula, ESAB (2000):

$$\text{Deposition efficiency} = \frac{\text{Weight of Weld Metal}}{\text{Weight of Electrode Used}} \quad (2.1)$$

The deposition efficiency tells how much weld metal that can be expected from a given weight of filler wire. It can be precisely determined by a timed test.

2.3 Arc Welding Technology

The term *arc welding* applies to a large and diversified group of welding methods. It covers all methods where metals are melted by the heat of an electric arc, with or without the use of filler metals. Arc welding is the most widely used welding method in trade and industry. The most common arc welding processes available are listed.

- Shielded metal arc welding (SMAW)
- Gas tungsten arc welding (GTAW)
- Gas metal arc welding (GMAW)
- Flux cored arc welding (FCAW)
- Submerged arc welding (SAW)

From hereon, only short notations of welding methods will be used. The different welding methods described in this section are based on teachings by ESAB (2000). The robotics laboratory used for the experimental implementation provides a type GMAW technology that will be used. Thus, special attention is paid to the physics of this welding method.

All welding methods except GTAW uses consumable electrodes. That is electrodes that melt and become part of the weld. GTAW, formerly known as tungsten inert gas or TIG, is the process where the base metal and the filler metal are melted by the intense heat of an arc that is maintained between the work piece and a non-melting tungsten electrode. The torch holding the non-consumable electrode transfers current through the electrode and provide an inert shielding gas to prevent atmospheric contamination of the weld metal. Heat produced by the arc melts the base metal creating a weld puddle. If filler wire is necessary, it is added to the leading edge of the molten pool. GTAW produces no slag and is used for welding almost all types of metals.

All four other methods have, in contrast to GTAW, an consumable electrode that is fed continuously through the torch in the form of filler wire. The arc is established between the electrode and the base metal. SMAW, GMAW, FCAW and SAW vary in terms of how the weld puddle is shielded from atmospheric contamination. SMAW is the most widely used method of all arc welding processes. It has a flux coated electrode that decomposes and produces a shielding gas for the weld puddle. The electrode metal core is transferred across the arc to the weld puddle. Molten slag floats to the top of the puddle where it protects the weld metal from contamination during solidification. The simplicity of this method makes it very popular as few components are needed.

GMAW has a bare metal wire electrode that is transferred across the arc into the molten weld puddle. The puddle is protected by a shielding gas that is fed by a gas regulator through the torch system. Figure 2.7 show a schematic representation of a conventional GMAW process. An inert gas was initially used for shielding, hence the term metal inert gas welding or MIG. Today, carbon dioxide and argon or other mixes of active and inert gases is used in the process. This is commonly referred to as metal active gas welding or MAG.

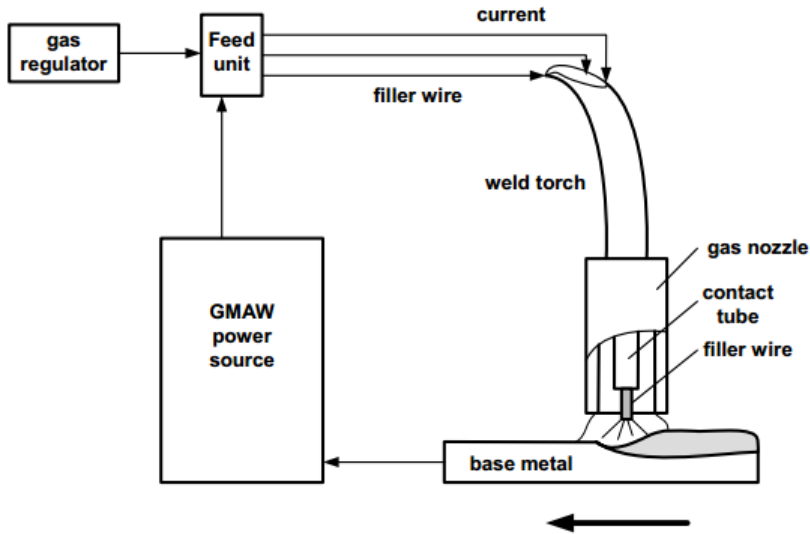


Figure 2.7: Schematic representation of a conventional GMAW process, Pires et al. (2006)

FCAW, like SMAW, uses a consumable electrode with flux. However, the electrode has a flux core instead of flux coating. The flux core decomposes, generating both a shielding gas and slag cover for the weld puddle. Based on the chemical composition of the core, both FCAW with and without external shielding gas are used. External shielding gas is added dependent on the base metal material. If no external gas is needed, the process is referred to as self-shielding FCAW, or FCAW-S. This configuration has a simple setup as no external gas equipment is needed. Although the flux core provides good protection against atmospheric contamination, slag have to removed in between passes. FCAW is, like GMAW and SMAW, effective in all positions.

SAW is different from the previous arc welding methods in the sense that the arc is not visible. SAW has an external feed that submerges the weld puddle in loose granular flux. The flux helps in forming the weld puddle, slowing the cooling rate and acting as a protective shield. As with FCAW, a slag cover is created. SAW is not suitable for welding in positions other than flat and horizontal because of the loose granular flux.

Advantages and disadvantages of the consumable electrode arc welding processes have been summarized in table 2.1.

Metal Transfer Modes

In order to obtain the most effective welding process, the filler metal should be transferred to the base metal with minimum loss to spatter. Arc instability caused by erratic transfer, especially in GMAW processes, can generate pressure fluctuations that draw air into the vicinity of the arc. There are three types of metal transfer modes: Globular transfer, spray transfer and short-circuiting transfer. Descriptions of metal transfer modes is given by

Process	Advantages	Disadvantages
SMAW	<ul style="list-style-type: none"> - Low initial cost - Flexibility - Usable in all positions - Portability - Numerous filler metals available 	<ul style="list-style-type: none"> - Slag
GMAW	<ul style="list-style-type: none"> - Higher deposition rates than SMAW - Flexibility - Adaptable to robotic or automatic welding methods - No slag 	<ul style="list-style-type: none"> - Needs special power source to be usable in all positions - Needs external gas supply and wire feeder
FCAW	<ul style="list-style-type: none"> - Higher deposition rates than SMAW and GMAW - Adaptable to robotic or automatic welding methods - Flexibility - Usable in all positions 	<ul style="list-style-type: none"> - Slag - Needs wire feeder - Needs external gas supply for most electrodes
SAW	<ul style="list-style-type: none"> - High deposition rates - High-quality, low-cost when mechanized 	<ul style="list-style-type: none"> - Only usable in flat and horizontal position - Large capital investment - Slag

Table 2.1: Advantages/disadvantages of welding processes using consumable electrodes, Bowditch (2005).

AWS (2001).

Globular and spray transfer modes are characterized by drops of filler metal transferred from the electrode to the base metal. A globular transfer contain large drops of filler metal while spray transfer contain a large number of small drops. In fact, the electrode never contact the base metal, but droplets of filler metal detaches from the electrode before they attach to the base metal. These modes are rarely found alone during a process but in some sort of combination of the two. A shift from globular to spray transfer mode occur at a critical current referred to as the transition current. A higher level of spatter is associated with globular transfer between these two. Short-circuiting transfer is characterized by the electrode periodically contacting the weld pool. Metal deposited this way is less fluid and less penetrating compared to the other transfer modes. A higher level of spatter is associated with this mode. It is minimized with the use of electrical inductance or feedback to control the rate of current rise when the electrode wire and pool are in contact. The amperage and voltage settings used are therefore low with this process. Short circuiting occurs at a rate of hundreds of shorts per second. Figure 2.8 show a schematic representation of the different transfer modes.

Soderstrom and Mendez (2008) studied the droplets size and transition current in GMAW welding with thin electrodes shielded by a mixture argon and carbon dioxide.

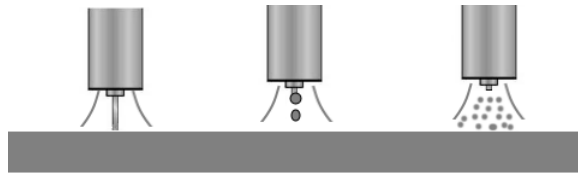


Figure 2.8: Metal transfer modes. From left to right: Short circuiting - globular - spray

One of the findings that this thesis would like to highlight is given by the graph in figure 2.9. The figure shows the observed transition current for a 0.035 inch (0.89 mm) electrode in a 90 percent argon 10 percent carbon dioxide environment. Here, the transition from globular transfer to spray transfer occurs at currents above 160-175 amperes. For larger diameter electrodes the transition currents are even higher. At carbon dioxide levels lower than 30 percent, they concluded that the transition current is not much affected by the amount of carbon dioxide from small electrode diameters (<1mm). This is an important finding for determining metal transfer in the laboratory implementation for this thesis.

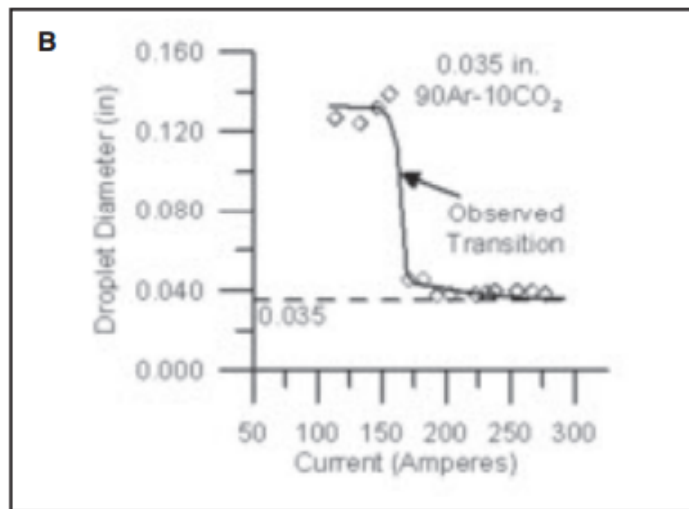


Figure 2.9: Characteristics between droplet diameter and current for a 0.035in (0.89mm) diameter electrode, Soderstrom and Mendez (2008)

The polarity of the base metal and electrode have a major influence on the metal transfer. Practically all GMAW is done using direct current electrode positive (DCEP). DCEP mode is achieved by connecting the direct current in such a way that the base metal is the negative pole and the electrode is the positive pole in the welding process. This polarity provides deep penetration, a stable arc and low spatter levels. If the connection is set the other way around, direct current electrode negative (DCEN) mode is achieved. Few GMAW processes are done using DCEN. The negative electrode creates an unstable arc and excessive spatter. The drop size is large, and the arc forces created by induced magnetic

fields actually propel the drops away from the base metal.

Process Parameters

The stability of the welding process is highly dependent on different process parameters. This is especially current, welding speed, voltage, wire stick-out and shielding gas, Pires et al. (2003). These parameters and their general influence on GMAW welding is briefly stated in this section. Keep in mind that all process parameters are to some extent dependent on each other.

Current and voltage are two codependent variables. Combinations of these together with the type of shielding gas determines the kind of metal transfer mode that is obtained. Low voltage and current, combined with an gas mixture containing an active gas give short circuiting transfer. For higher currents and voltages, the globular metal transfer is obtained. Higher current is needed for globular transfer if the shielding gas is purely inert. For very high voltages and currents, the metal transfer mode shifts to spray transfer.

Current in combination with welding speed control the depth of penetration and the level of fusion. High currents leads to increased energy input to the base metal, causing higher melting rates and deeper penetration. High welding currents together with a too low welding speed can also cause lack of fusion and cold lapping due to the cushion effect discussed earlier. To low currents combined with too high welding speeds can also cause cold lapping due the unmelted base metal.

Voltage controls the heat input to the base metal together with current by the electrical power equation ($\text{watts} = \text{current} \times \text{voltage}$). Increasing voltage widens and flattens the weld bead. However, excessively high voltages can cause arc instability, spatter, porosity and undercut. Low voltages increase the weld reinforcement.

Wire stick-out represents the length of filler wire sticking out of the contact tube of the torch. Increasing wire stick-out length increases electrode melting rate and affects the metal transfer mode. Lower stick-out lengths are used for short circuiting transfer mode while higher values are used for other modes. Too short stick-out can lead to slag inclusion in the weld bead and a uneven weld face due to low voltage. Too long stick-out length leads to excessive spatter and flat wide beads due to higher voltage.

The choice of shielding gas affects the arc stability, metal transfer mode as earlier stated, weld bead shape and melting rate. Pure carbon dioxide provide deep penetration but high levels of spatter. Pure argon generates less spatter but also less penetration. Adding carbon dioxide to argon stabilizes the welding arc and changes the bead shape to medium depth penetration and medium width bead.

2.4 Offshore Construction Welding

The fabrication of offshore constructions is usually a process with large amounts of welding involved. Offshore constructions covers offshore wind power, jackets, platforms, subsea facilities, fish farming cages, floating production storage and offloading facilities (FPSO) and pipelines. These immense structures are composed of subsections welded together to form a super structure. As the welded sections constitutes the main load carrying path of the structures, high quality welds are required. All structural steel fabrication for Norwegian offshore installations follow the NORSOK M-101 fabrication standard, NTS (2000). NORSOK M-101 states that all welding processes shall have WPS established in accordance with EN 288 part 2. A WPS contain valuable information and therefore often held secret by manufacturers. Hence, the following section are a general description of welding procedures and will vary with actual practices.

Jackets

Jacket manufacturing is an example of fabrication where welding constitute the main production process. A jacket have multiple nodes composed of braces intersecting with a chord. The main legs of a jacket have numerous intersections like these, as shown in figure 2.10. A single node can be seen in figure 2.11a. Welding them are a time-consuming process due to the shear size of the grooves.



Figure 2.10: Jacket *Gudrun* load-out, Kvaerner (2015)

Requirements for full penetration welds demand edge prepared joints. At the immense

size of the node components, edge preparation are generally done by manual grinding. All stubs in a node are prepared by either single bevel groove or double bevel groove depending on the angle between the stub and the chord. For angles less than 50° , the stub is welded on both the inside and the outside. There are different WPS for different sections in a jacket structure. A typical section is the intersection between a chord and a stub, creating a double-sided tubular joint. The tubular joints are positioned and tack welded together with a root gap between 3 and 6 mm. A popular method for welding tubular joints is FCAW. First, the root pass is laid with a ceramic backing. The operator then makes enough passes and beads to fill the groove, which can amount to several hundred at these dimensions. Figure 2.11d show a typical groove. Slag created from the process is removed in between each pass. The current and voltage are generally higher for filler passes than the root- and cover pass, while welding speed is lower for the root pass. Figure 2.11b show a completed weld at the intersection of two stubs and a chord. In narrow corners, such as the crown (the narrow point between chord and stub), a fillet weld is laid.

Automating the process of welding jacket structures has proven to be challenging. The main challenge related to welding nodes is the accuracy of the prefabricated parts. Parts at these dimensions has geometrical errors that need to be compensated for when automating. Manual grinding in the edge preparation contributes to a varying volume along the groove. Apart from geometrical errors, the number of passes needed to fill the groove is highly fluctuating. In fact, if the stub is attached at an angle, the weld volume will change around to edge of the stub. This can be seen in figure 2.11c. The figure also show both inside and outside edge preparations. Creating an automated system that is able to evaluate and decide where all weld passes are to be laid based on the highly variable geometry is one of the main challenges to robotic welding. Determining the number of passes needed and where to start and end them is for now done by a skilled welder. Other jacket components, such as conductor guides are better suited for to automation. The conductor guides can be seen at the bottom of the structure in figure 2.10. They are the 16 hollow tubes at the bottom arranged in two sets of 8 tubes, which is repeated to the top of the structure. Robotic welding firms are currently developing solutions for welding these smaller components. Whether automated welding of these have been implemented in the jacket fabrication process yet is not known to the author.

The structural nodes in the next generation offshore fish farming cages bear resemblance to jacket structures. These 100m diameter wide cages are rigid steel structures fabricated by welding together smaller components. Nodes are created by hollow tubular pipes intersecting at different angles. These nodes will face the same conditions as a jacket structural node. They are the load carrying component in the structure and require strong full penetration welds. Figure 2.12 show a offshore fish farming cage.

Thick Plate Welding

Thick plate welding is especially common in shipbuilding, heavy duty machinery, in fabrication of high pressure vessels and even nuclear components, Welding Institute (1975). Fabrication of large ships, such as FPSOs, is generally done by block assembly. These blocks are composed of several steel plates and steel sections with predetermined shapes. The construction process of a hull block comprises approximately 50 percent of the total shipbuilding process, Nakayama (1994). Hull blocks is manufactured by welding thick



(a) Hub/Node



(b) Multiple-pass weld



(c) Two-sided edge preparation



(d) Brace-chord groove

Figure 2.11: Jacket fabrication

metal plates together, both in terms of butt welds and fillet welds. As a FPSO consists of several blocks, welding geometries are repetitive. To this date, there are few automated solutions for welding ship hulls. Manually programming a robotic arc welding system for the manufacturing of a large vehicle hull takes more than eight months, while the cycle time of the welding process itself is only 16 hours, Pan et al. (2012). Due to this overhead programming time it is not beneficial to implement robotic welding. Robotic welding of these hulls is currently realized with online programming. That is walk through and lead-through of the end effector to teach robot the desired positions. The draw back of this method is that guiding the robot through the desired motion accurately, while never allowing a collision with an object in the workspace is difficult and time-consuming. The

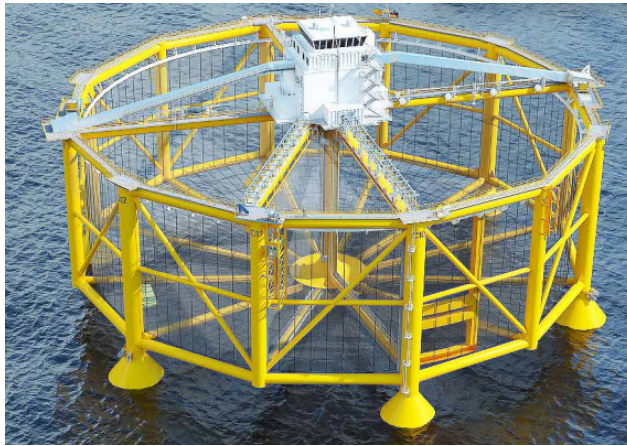


Figure 2.12: Fish farming cage, Ocean Farming (2015)

method lacks flexibility and re usability as the programming have to be repeated for a slightly different workpiece. Also, the robot cannot be used while online programming occurs.

2.5 Automated Multiple-Pass Planning

Mathematically describing welding procedures and joint geometries are necessary to automate multiple-pass welding. Describing the cross section of weld beads as sets of parallelograms and trapeziums is a practice that have been widely adopted by researchers. In real life, the section shape of a weld bead has an abnormal and variable geometric figure. Isao et al. (1979) presented the simplification and approximation for the weld beads. Figure 2.13 show how all weld beads are approximated. Based on the different section shape it can be seen that there are two distinct types of weld beads. The weld beads in the first and second layer as well as the right side bead in layer three and layer four can be simplified as trapezium, while the left side bead in layer three and four are simplified as parallelograms.

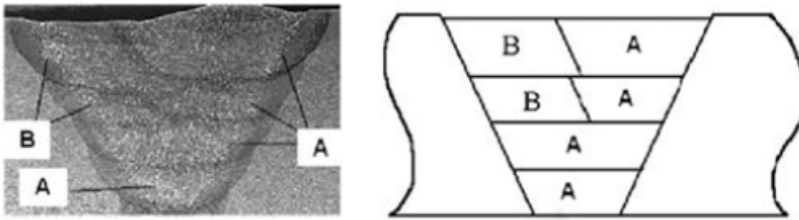


Figure 2.13: Actual bead shape versus simplified bead shape, Yang et al. (2014)

Offline programming is at present the used method for layout the multiple-pass planning. A few different variations of this have been implemented by researchers. Moriyasu et al. (1993) developed a program for automatically setting welding conditions, eliminating teaching of welding parameters for individual layers and passes in multiple-pass welding. The author stresses the importance of simplifying the teaching operation as a way to robotizing the welding of non-mass produced heavy electrical sections with various shapes of joints.

The developed system automatically sets the optimal parameters and torch positions based on the input welding position, groove shape and its size after teaching the weld line of the first pass. Moriyasu et al. adopted an equal bead height method. Figure 2.14 show a build-up sequence by the equal height method. A fixed leg length increase of Δl_2 will follow for all subsequent layers, after the first layer is laid. With a leg length increase of Δl_2 , the cross sectional area will increase by $\Delta(l_2)^2$ for each layer. After the welding position, groove shape and size were inputted, Moriyasu et al. implemented an algorithm to determine welding parameters. The algorithm selects wire feed speed from a database based on upper limit of

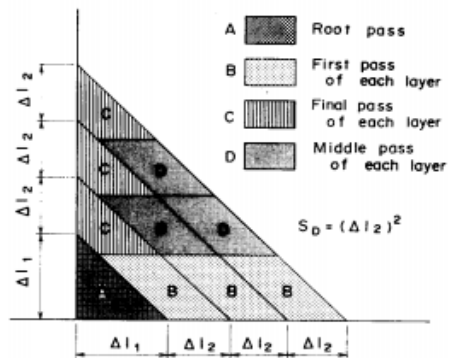


Figure 2.14: Build-up method and welding sequence for horizontal fillet welding, Moriyasu et al. (1993)

welding current and welding speed corresponding to the particular geometry. An appropriate voltage is set as soon as the mentioned parameters have been decided. This is repeated until the final pass is inputted. Li et al. (2001) later adopted the equal height method in a similar study. The two mentioned studies both investigated two strategies of multiple-pass welding of V-groove joints. One strategy was multiple layers with several passes in each layer. The other strategy was a single bead in each layer, meaning weaving was implemented. None of them reported any specifics on how weaving parameters were determined.

A weld bead shape is closely related with the welding parameters. There are databases that are developed to automatically calculate the current, voltage, welding speed and number of beads and layers based on the type of weld, the welding position, wire diameter and plate thickness, such as the one at the Welding Institute in the United Kingdom, Welding Institute (1975). Pires et al. (2003) proposed a software system using this database to extract welding parameters. The system implemented a computer-aided design (CAD) model of a V-butt joint and generated the robot motion program based on the welding parameters given by the database. The process was performed step by step to allow adjustments in the welding sequence that was programmed based on real time tracking from a 3D laser camera. This system was primarily a demonstration of how to assist and simplify industrial welding procedures as it reduced a set of existing limitation due to the nature of robot controllers. Robot controllers are not generally compatible with robot guidance sensors, nor are they powerful enough to handle tasks requiring complex control techniques. The proposed solution overcame these limitations by using a distributed software architecture enabling remote control through Ethernet. The guidance sensor would communicate with a personal computer that control the robot and not communicate with the robot controller itself. This way, it is possible to advantage of the huge amount of analysis and programming tools provided by a computer.

Zhang et al. (2009) developed another multiple-pass layout strategy for single sided V-groove and double sided K-grooves. The strategy was that from geometric features of the groove and some set welding parameters, complex calculations could predict weld scheduling. Here, the weld beads are approximated as trapeziums and parallelograms. Two types of welding sequence layouts were introduced: A side to side sequence, seen on the left in figure 2.15 and a side to center sequence, seen on the right in the same figure.

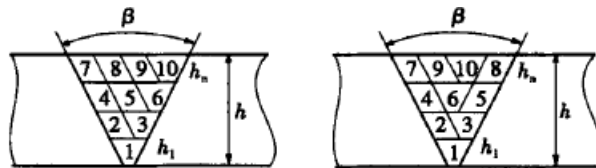


Figure 2.15: Welding sequence of single-sided V-groove, Zhang et al. (2009)

Zhang et al. introduced equations to calculate robot positions from given welding parameters. Equation 2.2 is the empirical formula describing the cross sectional area of a weld bead by DCEP MAG welding.

$$S = \frac{100E (-0.407d^2 + 1.384 \times 10^{-2}Id^2 + 1.9626 \times 10^{-6}I^2L)}{\rho vd^2} \quad (2.2)$$

Where E is the deposition efficiency coefficient, d is the filler wire diameter, ρ is the density of filler wire, I the current and v is the welding speed. The following three equations (2.3, 2.4 and 2.5) states the horizontal offset (y-direction), vertical offset (z-direction) and the work angle of the torch, see figure 2.16. In each layer i the total number of passes is m_i , where j corresponds to a specific pass. H represents the bevel depth, ΔZ_{ij} the vertical offset from bottom of the bevel to the top of layer i , ΔY_{ij} the horizontal offset from the middle of the groove to the torch position, β the groove angle, and $\Delta\theta$ the torch work angle.

$$\Delta Y_{ij} = \frac{m_i - 1 - 2(j - 1)}{m_i} \sqrt{\frac{\sum_{i=1}^n m_i S_i}{\sin\left(\frac{\beta}{2}\right)}} \quad (2.3)$$

$$\Delta Z_{ij} = \sqrt{\frac{\sum_{i=1}^n m_i S_i}{\sin\left(\frac{\beta}{2}\right)}} \quad (2.4)$$

$$\Delta\theta = \frac{1}{2} \left\{ \arctan\left(\frac{H - \Delta Z_{ij}}{2H \sin\left(\frac{\beta}{2}\right) - \Delta Y_{ij}}\right) - \arctan\left(\frac{H - \Delta Z_{ij}}{2H \sin\left(\frac{\beta}{2}\right) + \Delta Y_{ij}}\right) \right\} \quad (2.5)$$

These equations are valid for V-groove joints with zero root gap. Due to equal increments for ΔY_{ij} and the even equation 2.5, the work angle $\Delta\theta$ is mirrored about the vertical middle axis, thus the welding configuration will be of side to center as shown in the right side joint in figure 2.15. Zhang et al. reported no approach to multiple-pass welding using weaving, nor was any experiments conducted.

In 2011, Zhang et al. studied multiple-pass planning with the implementation of weaving. A method of setting welding parameters of every layer was adopted by the respective researchers. This provides more flexibility as welding conditions may not be suitable for welding using the equal height method. Also, a root gap g is introduced in the model. The solution algorithm follow this sequence:

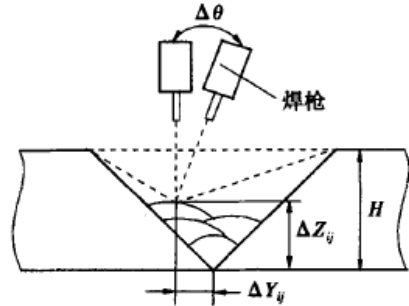


Figure 2.16: Multiple-pass fill layout of V-groove, Zhang et al. (2009)

3D modelling of joint, feature modelling, multiple-pass path layout, generation of path parameters (position, pose and oscillation displacement or weaving amplitude), selection of reference path, generation of multiple-pass path and finally generation of robot program. During the path layout step, users can set individual welding parameters for every layer, number of layers and number of passes in each layer. Selection of a reference path is done in a visual simulation program. Usually, this will be the middle line at the bottom of the groove. That is the angle bisector of both sides, see figure 2.17. Multiple-pass paths are then calculated.

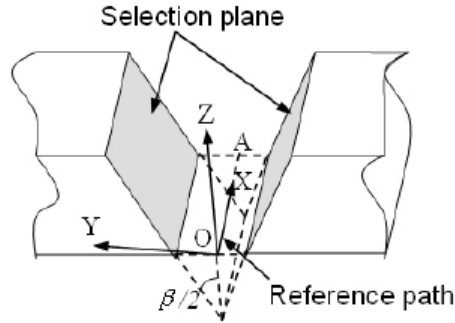


Figure 2.17: Reference path selection, Zhang et al. (2011)

Several of the equations given in this article are stated in chapter 3 as they are used for the solution of this thesis objectives. The equations are based on the geometric characteristics of a trapezium and a parallelogram. So now the beads are simplified to these two shapes, and the welding parameters such as the weaving width and torch position is based on trapezium and parallelogram geometry. Instead of an empirical formula for the cross-sectional area for weld beads, Zhang et al. introduced the following equation for calculation of the cross-sectional area, S .

$$S = \frac{\pi D^2 \phi(I)}{2v_2} \quad (2.6)$$

Here, D is the filler wire diameter, v_2 is the welding speed and $\phi(I)$ which states the wire feed speed as a function of the current I . Equations for torch positions and offsets, weaving amplitudes and torch work angles are deduced and stated in chapter 3. This study conducted laboratory experiments with the model. A layout model with torch poses and positions for each pass is given in figure 2.18a. For this five layer eight pass layout, increasing values for current is chosen in combination with decreasing welding speed, producing a larger cross-sectional area for each layer. The corresponding welded joint metallographic section is given in figure 2.18b. The figure shows the estimated sections shapes of each pass and reveal good fusion with the sidewalls. This experiment was done using MAG welding, similar to what is used in the laboratory implementation for this thesis.

Zhang et al. introduced a working model including weaving. However, the researchers did not report on experiments containing root- and cover pass.

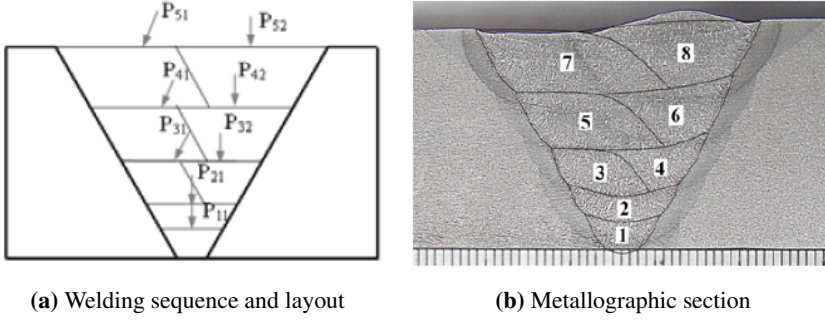


Figure 2.18: Multiple-pass welding experiment, Zhang et al. (2011)

Yang et al. published in 2014 a article proposing a solution to the issue of variable groove size and root gap. Multiple-pass planning based on a passive vision sensor was proposed. While Zhang et al. had a working model for multiple-pass planning, Yang et al. argued that the method was mainly aimed at path planning for an invariant geometry. In a manufacturing process, the geometry size of the grooves may be changed for each plate due to variable machining, or position deviation may occur due to inaccurate joint fit-up. Therefore, a vision sensor was introduced to extract real-time geometry sizes for multiple-pass planning. The vision sensor itself will not be discussed in more detail. For the multiple-pass path planning the authors implemented a side to center welding configuration, like Zhang et al. (2009). The following equation for the cross-sectional bead area are given, introducing the deposition efficiency coefficient a_H and wire feed speed v_1 .

$$S = \frac{a_H \pi v_1 D^2}{4v_2} \quad (2.7)$$

Other variables use the same notation is in equation 2.6. The article implemented an equal layer height/pass area method by holding a constant welding speed and wire feed speed, except for the root pass. Because of a narrower space to fill in the root pass, the deviation in height will be offset by lowering the wire feed rate. This permits the calculation of total number of layers, L , and beads, N by equation 2.8 and 2.9.

$$L = \frac{4v_2 h w}{a_H \pi v_1 D^2} \quad (2.8)$$

$$N = \frac{4v_2 h \left(h \tan \left(\frac{\beta}{2} \right) + g \right)}{a_H \pi v_1 D^2} \quad (2.9)$$

Here, h is the bevel depth and w is the weaving amplitude. Other parameters follow previous notation. Compared to previous studies, an alternative approach were also taken concerning the work angle of the torch. Instead of calculating variable work angles, a

simple rule was followed. If the root pass had a work angle α , the angle of the left and right bead adjacent to the sidewall would be $\alpha + 15^\circ$ and $\alpha - 15^\circ$. The work angle of the last pass in each layer would be α in similarity to the root pass. Other passes would have an work angle of $\alpha + 10^\circ$ and $\alpha - 10^\circ$ on the left and right side areas, respectively. Furthermore, a residence time of 0.5 seconds was implemented each time the torch moves to the sidewalls to ensure good fusion. Equations for calculating the vertical (z-direction) and horizontal (y-direction) offset is given by equation 2.10 and 2.11. Both offsets are calculated from the origin centered in the bottom of the groove.

$$\Delta Z_i = \frac{a_H \pi v_1 D^2 (i - 1)}{4v_2 w} \quad (2.10)$$

$$\Delta Y_{ij} = - \left\{ \left(\frac{(i - 1) a_H \pi v_1 D^2 \tan \left(\frac{\beta}{2} \right)}{4v_2 w} + g \right) - \delta \sum_1^{j-1} 2w + w \right\} \quad (2.11)$$

The interesting momentum in these equations is the introduction of the parameter δ . This parameter represents the length of the next weld bead covering the former one. This was discussed by Zou et al. (2000) who stated that the advantage of this method is that the next weld bead produces an effective heat treatment on the former one. δ is in this experiment set to half the former weld bead width. Figure 2.19 show both a welding sequence layout and the metallographic section of the corresponding welded joint. The researchers reported good agreement between the path planning and the experimental results, as well as good fusion between passes and penetration to sidewalls.

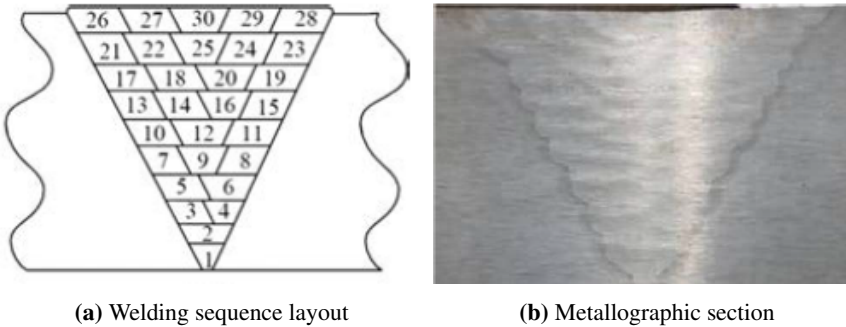


Figure 2.19: Multiple-pass welding experiment, Yang et al. (2014)

Yet another approach to automated multiple-pass planning for thick plates was presented by Huang et al. in 2015. A strategy for multiple-pass path planning was established on the equal area assumption for each bead, and the adoption of the parallelogram/trapezium approximation to plan layers, number of beads in each layer, and the position of each bead. In addition to a new planning approach for the welding sequence, Huang et al. also introduced a specially designed cover pass. The authors argue that the filler wire cannot fill

all the vacancies of the groove under conventional path planning. This will lead to notches causing stress concentrations. The cover pass need to fill the vacancies and connect the surfaces of the two sides. Figure 2.20 show the schematic diagram of the groove. Notice the notation change from previously shown schematics.

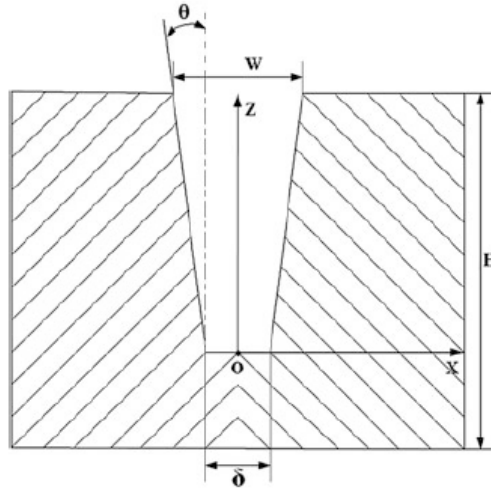


Figure 2.20: Schematic diagram of the groove, Huang et al. (2015)

A few new parameters are introduced related to the area of the bead. S remains the cross-sectional area of the bead, while l is introduced as the bottom width of the bead and l_i as the bottom width of layer i . W is the upper width of the groove. The number of welding beads in each layer, m_i can be calculated by equation 2.12. As m_i generally will be a decimal number, it is rounded to the nearest integer.

$$m_i = \left\lceil \frac{l_i}{l} \right\rceil \quad (2.12)$$

The vertical offset of each layer (z -direction in this study) is calculated by equation 2.13, and the horizontal offset (x -direction) by equation 2.14.

$$\Delta Z_i = \frac{-l_i + \sqrt{l_i^2 + 4 \tan(\theta) \sum_{i=1}^n m_i S}}{2 \tan(\theta)} \quad (2.13)$$

$$\Delta X_{ij} = \frac{m_i - 1 - 2(j-1)}{2m_i} (l_i + 2\Delta Z \tan(\theta)) \quad (2.14)$$

A constant K is introduced as the assumed total number of layers. With these three equations and respective parameters, a path planning algorithm is followed. The algorithm

is shown by the diagram in figure 2.21a. The system starts with measuring the layer width l_i then calculating the numbers of passes m_i . The vertical offset ΔZ can now be found. All passes are then looped through, from $j = 1$ to $j = m_i$ calculating the offset ΔX and respective welding coordinates for each pass. The whole sequence is repeated until layer $i = K$. From there, a special cover pass sequence is initiated.

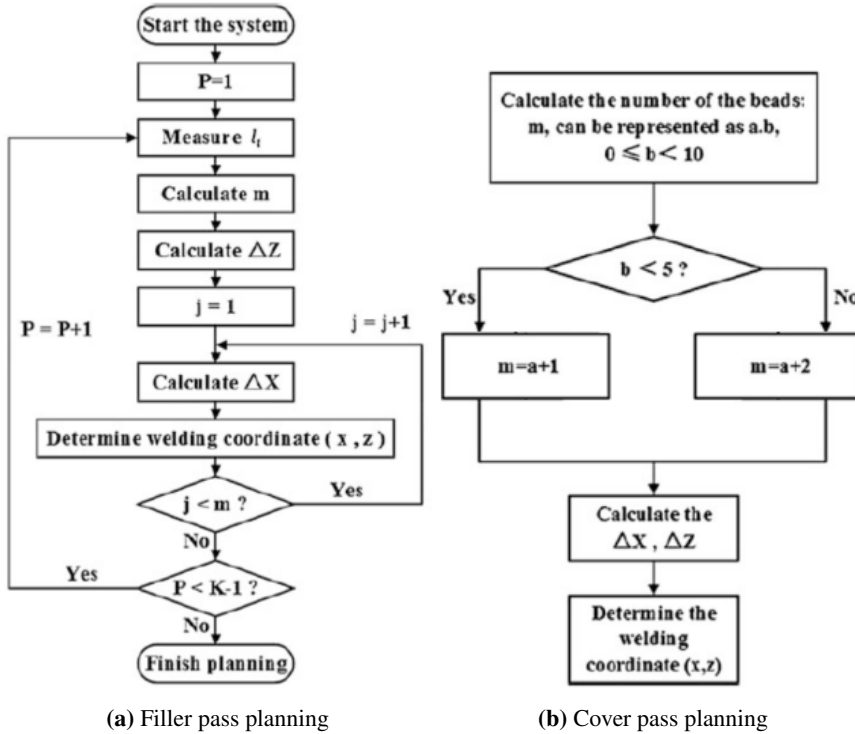


Figure 2.21: Multiple-pass path planning, Huang et al. (2015)

The core principle of the cover pass planning is to add one more bead to the last layer. This is done to feed more wire, thus increasing the total area of the filling layer. The cover pass sequence algorithm is shown in figure 2.21b. It starts with calculating m_i . Instead of simply rounding the calculated bead number to the nearest integer, it is represented on the format $a + b/10$. The number b is taken as a decision criterion for how many beads are required to fill the vacancies. If $0 \leq b < 5$, one more bead is added to the last layer. If $5 \leq b < 10$, two beads are added to the last layer.

Figure 2.22 show a comparison between the welding sequence layout and the welded joint. The experiment was done using laser welding with continuous filler wire. An interesting observation in this experiment is the vacancies in the first few layers. This is lack of fusion between layers and the base metal. The researchers attribute this to the instability of the feeding of filler wire and shielding gas, and deviation of laser spot and wire. Nonetheless, the researchers were content with the achieved results and the cover pass process result.

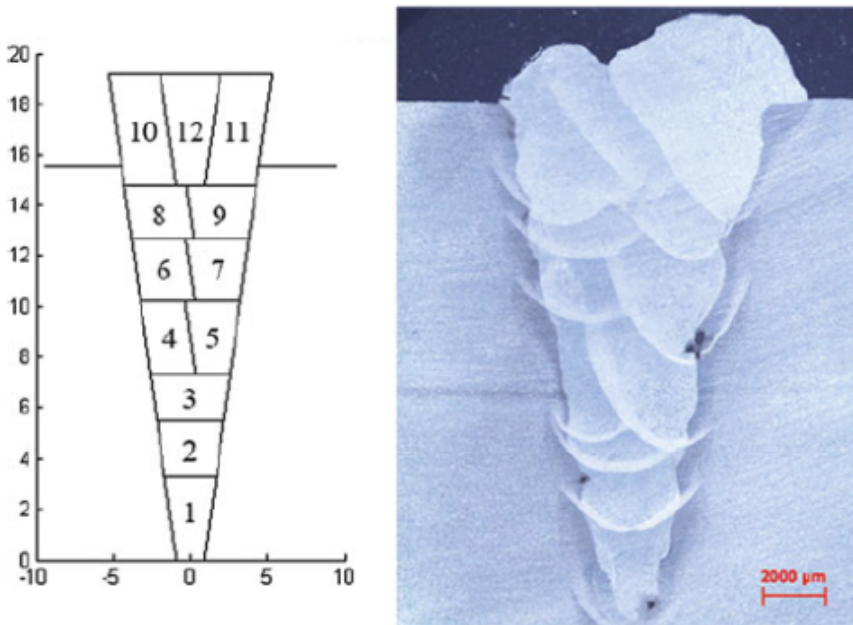


Figure 2.22: Multiple-pass welding experiment, Huang et al. (2015). Left: welding sequence layout. Right: metallographic section

Discussion

All the discussed research in this section has introduced different methods of multiple-pass path planning and approach the issue of thick plate welding. Based on the various results of this research, some principles are retained in the presented solution in this thesis while some principles are discarded or further developed.

To maintain flexibility of the automated solution, the ability to choose different welding parameters for each pass will be maintained. Experiments by Huang et al., Zhang et al. and Yang et al. show that holding these parameters constant simplifies the sequence calculations and path planning. However, a solution better suited for facing highly variable groove geometries will benefit from the option of changing parameters in between layers. Reducing or increasing the number of beads may be desirable. Zhang et al. implemented this in their new study in 2011. The same principle will be maintained in this thesis. A side to side welding sequence is taken.

None of the conducted experiments had root faces in their joints, which eliminated the root pass. The grooves had a zero root face and was welded with backing. Thick plates are usually edge prepared with a root face, raising the need for a root pass. A solution containing root pass will be presented in this thesis. Conventionally, the first pass in a welding sequence is always called the root pass. As a way to easier distinguish joints with and without root faces, the first pass in a weld without root face will be referred to as a filler pass in this thesis.

The study by Huang et al. had a special sequence plan for the cover pass where they

added an extra layer in order to fill the groove. The cover pass generated by the experiment was, however, an uneven surface. A desired butt joint weld surface is a slightly convex surface with an uniform weld face. Another approach is therefore presented in the solution. Here, the cover pass will respectively be laid by one pass with an high welding amplitude.

Solution

3.1 Introduction

The solution for the third objective of this thesis is mainly solved using *MATLAB* by MathWorks and *3DAutomate* by Visual Components. *MATLAB* provides a user friendly interface and a powerful tool in terms of easy programming and graphics. *3DAutomate* is a 3D factory simulation solution capable of creating large scale production-line simulations. Its capability for creating simulations with robot motion statement programs with available models of various industrial robots makes it ideal for this solution.

Section 3.2 provides the equations used for calculating the geometry of the weld passes and the positioning of the robot end effector. These equations are what constitute the basis for the robot motion statement program generation.

Section 3.3 outlines the developed *MATLAB* script and its functions. The visualization and output from the script is given as well as the algorithm that determines the multiple-pass planning. This algorithm, with associated functions, is transferred into *3DAutomate* for simulation purposes and robot motion statement program generation.

Section 3.4 looks more closely at the implementation in *3DAutomate* and the components and scripts developed in this software environment. This includes a parametric model of the joint, translation of the *MATLAB* script into understandable Python language and also Python script for generating robot motion statements. Additionally, a method for generating root pass and cover pass motion statements is outlined. The limitations due to the built-in Python version in *3DAutomate* are also discussed in this section.

3.2 Equations

The derivation of the following equations are mainly based on previously presented article by Zhang et al. (2011) with some adjustments by the author. Some equations are also solely derived by the author. These are related to the weaving width, transverse and longitudinal deviation related to the parallelogram. A notation for the different parameters is introduced in table 3.1. Note that some parameters may differ from previously introduced notation.

θ	Torch angle
β	Groove angle
δ	Weaving factor
Δh_i	Vertical offset from top of root face to layer i
Δy	Horizontal offset from center to torch position
Δz	Vertical offset from top of root face to torch
a_H	Deposition efficiency coefficient
D	Diameter of filler wire
g	Root gap
h_i	Thickness of layer i
i	Layer number
j	Pass number
m_i	Total number of passes j in layer i
S_i	Area of layer i
$S_{i,j}$	Area of layer i pass j
v_1	Feed speed
v_2	Welding speed
W	Weaving amplitude

Table 3.1: Welding equation notation.

The cross-sectional area of a single weld string can be described in terms of the parameters welding speed, feed speed and filler wire diameter, see equation 3.1.

$$S_{i,j} = \frac{\pi v_1 D^2}{2v_2} \quad (3.1)$$

Feed speed is essentially a function of the welding current. Zhang et al. (2011) used the current I as a adjustment parameter, replacing v_1 in equation 3.1 by a function $\phi(I)$. Li et al. (2015) and Yang et al. (2014) included a correction for the cross-sectional bead area formula by a factor of $\frac{1}{2}$. As this thesis will mainly adjust the feed speed as a parameter, the v_1 parameter is kept. Additionally, a deposition efficiency coefficient a_H is introduced to account for material loss, similar to Yang et al. (2014) equation for the cross-sectional area. The final equation then becomes

$$S_{i,j} = \frac{a_H \pi v_1 D^2}{4v_2} \quad (3.2)$$

As the total shape of one layer will be a trapezium regardless of the number of passes, the area could be described in terms of the geometry, given in equation 3.3.

$$Area = \frac{1}{2} (bottom + top) h_i = S_i \quad (3.3)$$

The total area of i layers is in equation 3.4 written in terms of the introduced notation. Rearranging this equation with respect to Δh_i , utilizing the second degree formula, gives the offset from the top of the root face to the top of layer i , equation 3.5.

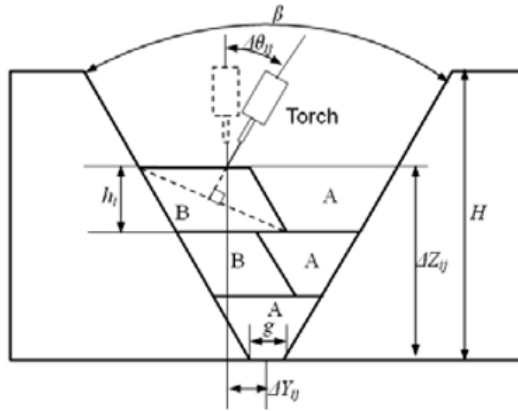


Figure 3.1: Section shape of multiple pass welding seam, Zhang et al. (2011)

$$\sum_k^i \sum_l^{m_i} S_{k,l} = \Delta h_i^2 \tan\left(\frac{\beta}{2}\right) + g \Delta h_i \quad (3.4)$$

$$\Delta h_i = \frac{\sqrt{g^2 + 4 \tan\left(\frac{\beta}{2}\right) \sum_k^i \sum_l^{m_i} S_{k,l}} - g}{2 \tan\left(\frac{\beta}{2}\right)} \quad (3.5)$$

Layer specific thickness is then given by equation 3.6.

$$h_i = \Delta h_i - \sum_{k=0}^{i-1} h_k \quad (3.6)$$

When $m_i = 1$ or $j = m_1$, the shape of the weld bead belongs to a trapezium A, or a triangle in the case of $g = 0$, see figure 3.1. The pose and positional deviation are calculated by equation 3.7, 3.8 and 3.9.

$$\Delta y_{i,m_i} = \frac{(m_i - 1) S_i}{2m_i h_i} \quad (3.7)$$

$$\Delta z = \Delta h_i \quad (3.8)$$

$$\Delta \theta = 0 \quad (3.9)$$

When $m_i > 1$ and $j < m_1$, the shape of the weld bead belongs to the parallelogram B, ref. figure 3.1. Figure 3.2a show the different line segments in the parallelogram which can be calculated by geometrical relationships. Note that the equations related to the parallelogram is not consistent with what is deduced by Zhang et al. (2011). This applies to equations 3.10-3.13, 3.15 and 3.17. These equations are deduced in this thesis.

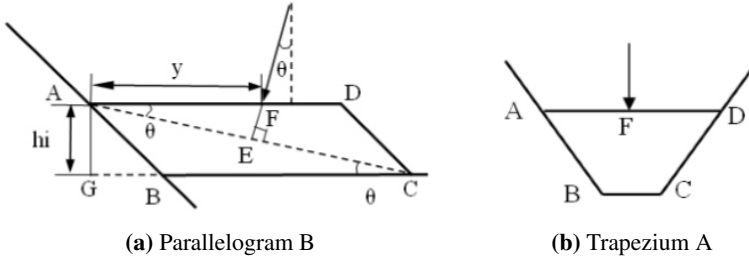


Figure 3.2: Weld bead geometry, Zhang et al. (2011)

By realizing the similar triangles, i.e. $\triangle ACG \sim \triangle AFE$, the ratio between the line segments can be found. The ratio of interest is given in equation 3.10.

$$\frac{y}{AC} = \frac{AE}{GC} \quad (3.10)$$

Further, the intersecting line FE, an extension of the torch direction, is the perpendicular bisector of line segment AC. Inserting that $AE = \frac{1}{2}AC$ yields

$$y = \frac{\frac{1}{2}AC^2}{GC} \quad (3.11)$$

The transverse deviation y is then as follows.

$$y = \frac{\frac{1}{2} \left(\left(h_i \tan \left(\frac{\beta}{2} \right) + \frac{S_i}{m_i h_i} \right)^2 + h_i^2 \right)}{h_i \tan \left(\frac{\beta}{2} \right) + \frac{S_i}{m_i h_i}} \quad (3.12)$$

The offset deviation in y - and z -direction relative to midpoint at the top of the root face, in addition to the pose of the torch is given by equation 3.13, 3.14 and 3.15.

$$\Delta y_{i,j} = y - \frac{g}{2} - \Delta h_i \tan\left(\frac{\beta}{2}\right) + \frac{\sum_{k=1}^{j-1} S_{i,k}}{h_i} \quad (3.13)$$

$$\Delta z = \Delta h_i \quad (3.14)$$

$$\Delta\theta = \arctan\left(\frac{h_i}{h_i \tan\left(\frac{\beta}{2}\right) + \frac{S_i}{m_i h_i}}\right) \quad (3.15)$$

Equation 3.16 and 3.17 are used to determine the weaving amplitude of the trapezium and the parallelogram. Here, δ is the weaving factor shown in figure 3.3. The value of δ is empirically found.

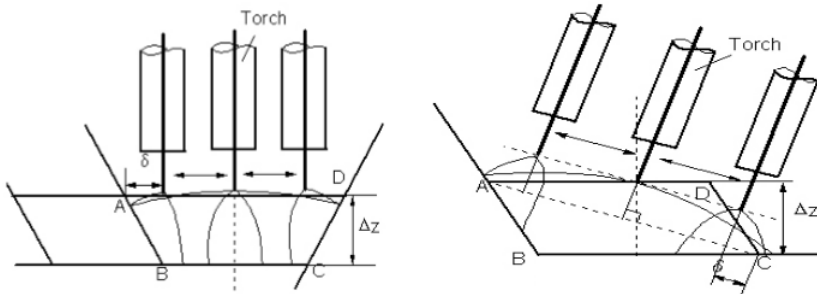


Figure 3.3: Weaving width of torch, Zhang et al. (2011)

$$W_{trapezoid} = \frac{1}{2} h_i \tan\left(\frac{\beta}{2}\right) + \frac{S_i}{2m_i h_i} - \delta \quad (3.16)$$

$$W_{parallelogram} = \frac{1}{2} \sqrt{\left(h_i \tan\left(\frac{\beta}{2}\right) + \frac{S_i}{m_i h_i}\right)^2 + h_i^2} - \delta \quad (3.17)$$

3.3 Multiple-Pass Sequencing Script

A vital part of the solution is the multiple-pass sequencing script. *sequencer.m* is a script written in MATLAB containing an algorithm for multiple-pass sequencing. The script is based on equations given in the previous section. A table containing its sub functions and corresponding equations is given in table 3.3. The sequencing algorithm is designed for welding butt joints with a zero root face and with backing. Subsequently, the script is designed for calculating all filler passes of the joint. Parameters for root pass and cover pass will be studied in later sections. An example of the visual output of the script can be seen in figure 3.4. The figure show a 60 degrees, 10 mm bevel with three layers. Layer one and two each consists of one pass denoted 1,1 and 2,1. Layer three has two passes denoted 3,1 and 3,2. The red arrow represents the torch position and orientation for each pass. Axis are set with aspect ratio equal to one.

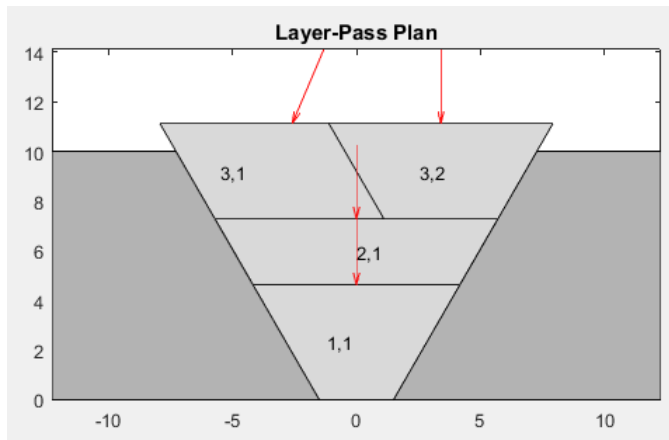


Figure 3.4: Visual output from *sequencer.m* example

Layer	Pass	v_1 [mm/s]	v_2 [mm/s]	Δy [mm]	Δz [mm]	$\Delta\theta$ [$^\circ$]	W [mm]
1	1	100	3	0	4.62	0	2.2
2	1	100	3	0	7.27	0	3.7
3	1	100	3	-2.59	11.1	23.1	2.9
3	2	100	3	3.40	11.1	0	2.5

Table 3.2: Parameter table output from *sequencer.m* example.

The script offer a great deal of flexibility in terms of its inputted parameters. A user initiates the script by setting the geometrical parameter values for the geometry, i.e. groove angle, root gap and depth of bevel. Further, the user set parameter values for diameter of filler wire, wire feed speed and welding speed. The first layer is automatically set to contain one pass by default. However, in the following layers an algorithm for proposing a minimum number of passes in the respective layer is ran. This algorithm, located in

function *passlayerprop.m* have two determinative factors: Maximum weaving width and maximum bead height. Maximum values of these two factors can be arbitrarily set by changing the logic test in *passlayerprop.m*. Still, careful attention should be paid to these values as a too large bead height may cause a too large electrode wire extension thus weld defects, while too large weaving width may cause too fast oscillation of the torch if the frequency of weaving is not accounted for. Default maximum value for bead height is determined by recommendations in reviewed literature and is set to 6.4 mm, Bowditch (2005). Default maximum value for weaving amplitude is set to four times the electrode diameter. Even though they are called maximum values, the user is able to overwrite and arbitrarily choose the number of passes. Function *passlayerprop.m* only provides a recommendation, if possible. Welding speed and wire feed speed is the main input parameters for this algorithm. Based on the area for a bead due to these two values and the geometry of the groove, the algorithm may or may not be able to find a solution. Too large bead areas cause problems related to either the maximum bead height or the maximum weaving amplitude. Only certain combinations of these parameters and criteria are possible. If the algorithm cannot calculate a solution, a statement is printed suggesting a reconsideration of the chosen values for feed speed and welding speed.

Script	Function	Equation
<i>sequencer.m</i>	<i>crosssectionarea.m</i>	3.2
	<i>delta_h_i.m</i>	3.5
	<i>delta_theta_para.m</i>	3.15
	<i>delta_y_para.m</i>	3.13
	<i>delta_y_trapes.m</i>	3.7
	<i>h_i.m</i>	3.6
	<i>passlayerprop.m</i>	
	<i>weaving_para.m</i>	3.17
	<i>weaving_trap.m</i>	3.16

Table 3.3: MATLAB: Script, functions and corresponding equations

Welding speed and feed speed is by default set equal for all layers. However, different values for feed speed and welding speed can be set for each layer influencing the cross sectional weld area. As will be explained later in chapter 4, these two parameters will be held constant for all filler passes during the experiments. The script has two outputs. A visualization of the cross section containing all passes with torch position and work angle for each pass, and a parameter table. An example of the parameter table is given in table 3.2. The notation correspond to the introduced notation in table 3.1. In this example, $v_1 = 100\text{mm/s}$ and $v_2 = 3\text{mm/s}$ with weaving factor $\delta = 2.0$. The associated visualization is given by figure 3.4. A flow chart of the all logic in *sequencer.m* is given in figure 3.5.

The script *sequencer.m* with associated functions are available in the digital appendix. Also, *sequencer.m* is added in appendix A.

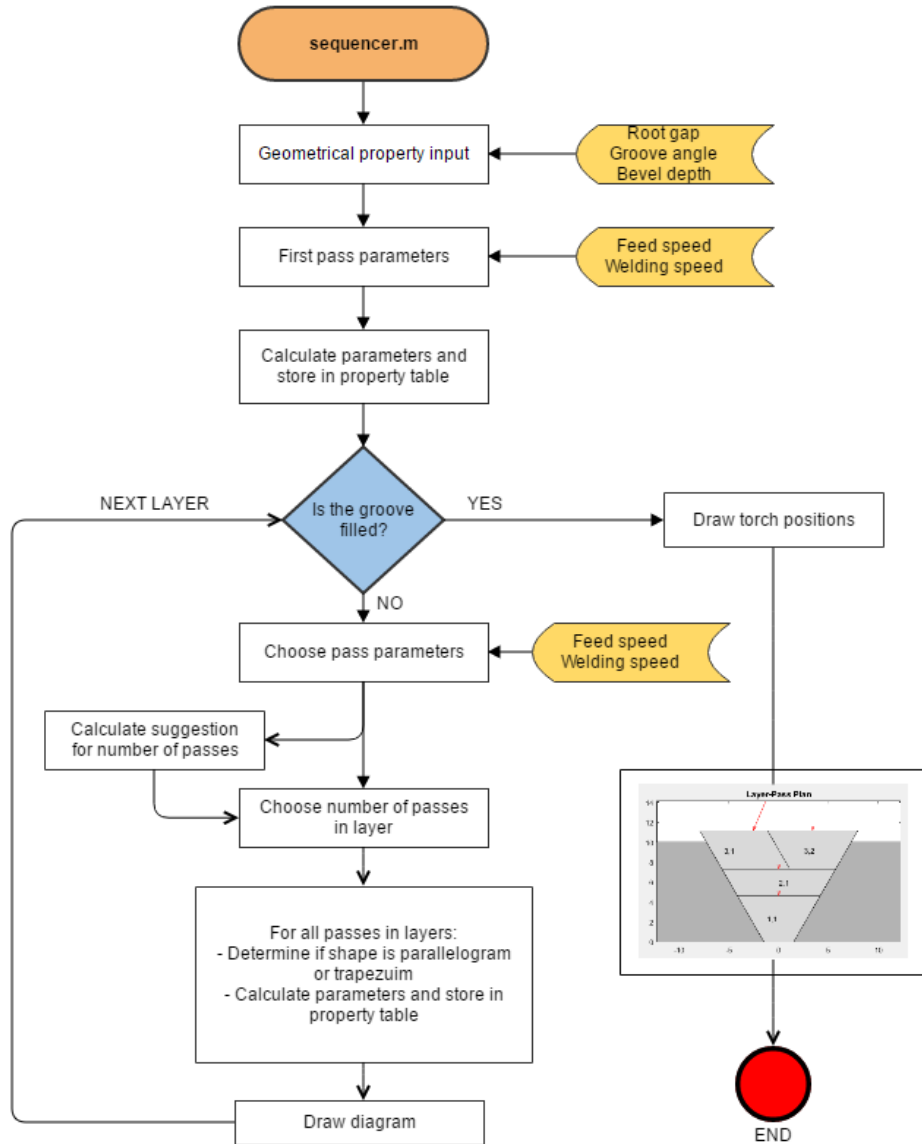


Figure 3.5: *sequencer.m* information and work flow

3.4 Visualization and Simulation

3DAutomate features a Python application programming interface (API) that enables the end user to program components, behaviours, motion patterns and so forth. The Python API has in conjunction to this thesis been used for making three different scripts: one script for creating a parametric model of the joint, one for implementing the same multiple-pass sequencing as in MATLAB, and one for creating a robot motion statement program. The former two scripts is linked to the joint component in 3DAutomate while the latter is linked to the robot component. All three scripts communicate information that is needed for creating the simulation.

Parametric Joint Model

One advantage with Python API is the ability to create simple 3D models. The script *jointmodel.py* creates a graphical user interface (GUI) in 3DAutomate which allow the user to choose geometrical properties for the butt joint. That includes the bevel angle, root face, root gap, plate thickness (depth of bevel) and width and length of the plates. Once the desired property values are set, the model is regenerated based on these values. The script makes use of the built in geometry feature *TRIANGLESET* which draws triangles from a set of three vectors. All together, the model is composed of 40 triangles. The base frame is set at the end of the bottom surface in between the plates, denoted 'Frame' in figure 3.7. Script *jointmodel.py* can be found in appendix B. Figure 3.6 show the GUI and the model side by side. The properties are stored in the GUI.

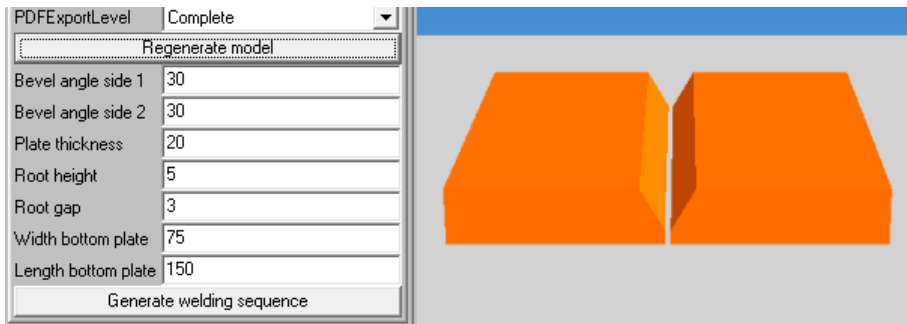


Figure 3.6: Parametric joint model and GUI in 3DAutomate

For information, 3DAutomate also features a Component Object Model API (COM API), providing support for importing and connecting to CAD-models built on other platforms. In essence, the model could have been made in *SolidWorks* and imported to 3DAutomate but possibly at the expense of the easily retrievable geometry properties.

Weld Sequencing Script in Python

Weld sequencing in 3DAutomate is nearly identical to the MATLAB script *sequencer.m*. The script is translated into *sequencer.py* with a few minor differences. First, geometrical

parameters for the joint is retrieved from the parametric joint model instead of inputted by the user at script execution as in *sequencer.m*. Second, the visual output from *sequencer.m* is omitted. One of the major drawbacks concerning the Python API in 3DAutomate is the I/O functionality, or the lack of it. 3DAutomate offer no support for user input during script execution like MATLAB. Instead of choosing the number of passes in each layer as the sequencing progresses, the numbers have to be predetermined. Also, the possibility to change feed speed and welding speed in between each layer is disabled. However, different values could still be predetermined.

There a few possible ways to attack this problem even though the solution will not be equal to what *sequencer.m* provide. One way is to create a new GUI in 3DAutomate with input fields for welding speed, feed speed and number of passes in each layer. The drawback of the method is that the GUI could potentially become large and confusing as the number of layers increases. Numerous input fields would acquire large parts of the 3DAutomate interface. Another way is to implement the algorithm *passlayerprop.m* in the Python script, and let it be the determinative force for pass number selection. The problem here is the instances where the algorithm cannot find a solution within the boundaries. In *sequencer.m*, the operator had the ability to overwrite this at the cost of either bead height or weaving width. This will not be possible in 3DAutomate. A workaround would be to always choose the pass number closest to the boundaries in these cases. Implementing the algorithm would provide a more automated solution as it would eliminate the need for user input. The chosen practice, though unfortunate due to I/O limitations, is to hard code the number of passes, feed speed and welding speed in the script before executing it. These numbers have to be in compliance with the numbers chosen in *sequencer.m* in order to compare the visual pass-layer plan from MATLAB with the experimental test result.

The values calculated by the sequencing script have to be stored for later use by the robot motion statement generation script. One I/O functionality supported by Python 2.5, that is not dependent on the 3DAutomate user interface, is writing or reading external files. For this reason, a external file system was used to communicate the weld sequence parameters. After all parameters are calculated by the sequencer and stored in arrays, it is written to a comma-separated value file (a .CSV-file). A CSV file stores tabular data as plain text separated by commas. It is very common method of exchanging data as it is a widely supported format.

The script *sequencing.py* can be found in appendix C

Robot Motion Statement Generation

The action script for robot motion statement program generation, *robotmotion.py*, has two main inputs: geometrical features from the joint model and the .CSV file from the sequencing script. The geometrical features include the positional matrix of the model and the geometrical properties set in the model GUI. Based on these inputs, a new robot base is defined at the top of the root face at the end of the model. Figure 3.7 illustrate the position of the new robot base, denoted 'BASE_DATA[2]'. Notice how both the joint model system 'Frame' and robot base 'BASE_DATA[2]' are rotated 90 degrees in negative direction around the z-axis in relation to the world coordinate system orientation. The world coordinate system has its origin at the bottom of the robot but the orientation is shown by the red, green and blue arrow, respectively the x-, y- and z-axis with the model base

'Frame' in figure 3.7. As a result, equations for the offset in y-direction given in section 3.2 corresponds to the offset in x-direction in *robotmotion.py*. This is also the case for all parameter and property tables given throughout the thesis.

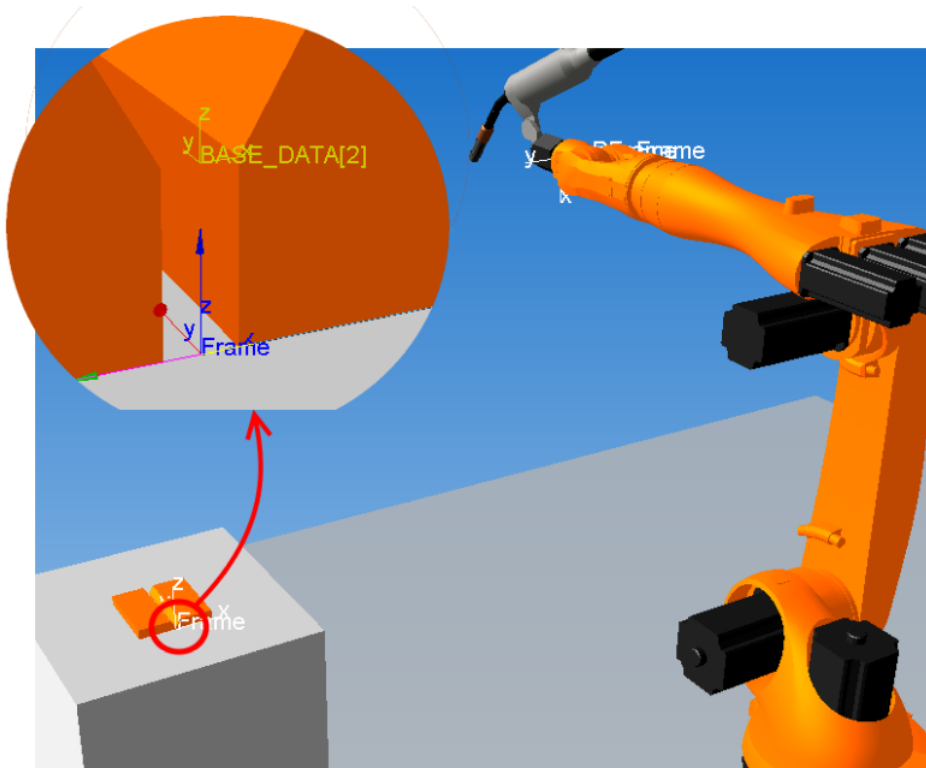


Figure 3.7: Base shift and rotation

The script retrieves the .CSV file once the new robot base is defined. The .CSV text is converted into arrays of floating point values. Values stored in the arrays constitute the basis for the motion generation. The script loops through all values, that is all layers and passes generated by the sequencer, and creates the robot motion statements. One loop is based on the following algorithm and then repeated for all passes:

1. Position tool at a distance above starting point, position (0,10,100)
2. Move to weld start position based on parameters given by the sequencer
3. Move to weld end position 10 mm from end of model
4. Retract tool to 100 mm above end position

robotmotion.py can also create motion statements for both the root pass and cover pass. The weld torch position is set to the top of the root face for the root pass and at the top of

the model surface for the cover pass. At current version, the script will always assume a root pass is needed. However, if the root face is set to zero the root pass statements will have to be commented out of the code in order to avoid an unnecessary weld pass.

Another functionality of the *robotmotion.py* is generation of a weaving pattern. A weaving functionality is coded into the script yet commented out due to a built in weaving function in *ArcTech Basic*. This will be explained further in chapter 4. However if needed, the user can set a desired weaving frequency in the robot motion script and a oscillating weaving motion is calculated for each pass. The only pattern supported at this point is triangle weaving, also known as zig-zag motion. The script *robotmotion.py* can be found in appendix D.

Weld Sequence Simulation

Simulating the weld sequence is a three step process. The processes have been explained in the previous three sections. Once all process steps are done and a robot motion statement program is generated, 3DAutomate stores the motion program as a RSL (Robot Scripting Language) file. To summarize the process:

1. Generate model
2. Generate welding sequence
3. Run simulation for motion statement program

The information flow is given in figure 3.8, showing all links within the 3DAutomate environment.

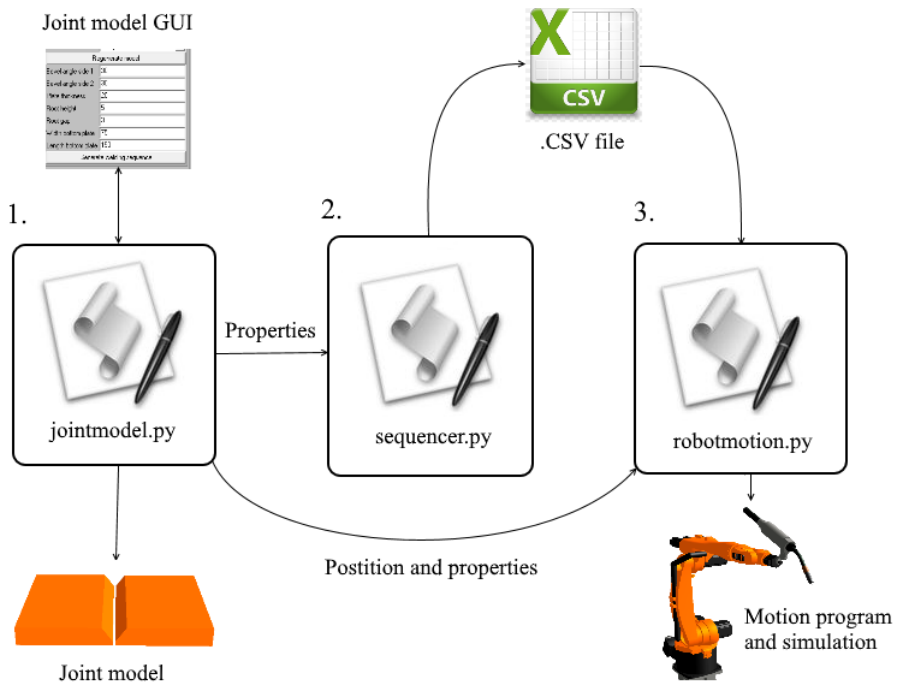


Figure 3.8: 3DAutomate information and work flow

Laboratory Implementation and Testing

4.1 Introduction

The laboratory implementation and testing of the solution model constitutes an important part of this thesis. A lot of time was dedicated to testing and achieving a satisfactory performance. Welding is a highly vibrant fabrication method dependent on equipment and multiple parameters. It is therefore of interests to thoroughly describe software, equipment and methods used to perform the laboratory experiments.

As a part of the goal of the thesis is to obtain compliance between the mathematical model and real life welding of the joints, different welding parameters have to be determined. Certain variables have to be empirically found in order to obtain an accurate model. All parameters that will be tested for will be explained in this chapter.

Section 4.2 will provide some basic information on the robot cell at Department of Production and Quality Engineering. Hardware and software used specifically for this thesis will be outlined in section 4.3. The experimental setup with the robot, the welding system, fixtures and test parts will be provided in section 4.6. Section 4.5 discusses the information flow from generation of weld sequence program to the execution by the robotic welding system. At the end, a plan for testing plan will be outlined in section 4.7. This includes what parameters will be tested for, why they are tested, and the progression of test welding.

4.2 Robot Cell

The robot laboratory cell at IPK consists of four robot from *KUKA Roboter GmbH*. Respectively two *KR 120 R2500 pro* robots from the *KUKA QUANTEC* series, one *KR 16-2* robot and one *KR 5 arc* robot. All of them are equipped with their own *KUKA KR C4* robot controller. A programmable logic controller (PLC) connected to all four robots ensures safety features like emergency stop and door switch, communicating via *ProfiNET*. The cell also includes a welding system from *Fronius International GmbH*.

4.3 Fronius Welding System

The *KR 5 arc* robot was used to conduct the welding operations. A welding system from Fronius configured for MAG welding is physically attached to the *KR 5 arc* robot. The system consists of a *TransSteel 5000 Rob* power source, *VR 5000 Rob* wire feed unit, a *DeviceNet* robot interface module, *FK 5000 MV* cooling unit, gas supply tank, welding wire drum and a *Robotca GTW5000* welding torch attached to a magnetic crash box. The system configuration can be seen in figure 4.1. The figure do not include the wire drum which is actually attached to the robot base. All information regarding the Fronius system is taken from various operating instructions and manuals by Fronius International GmbH (2010).

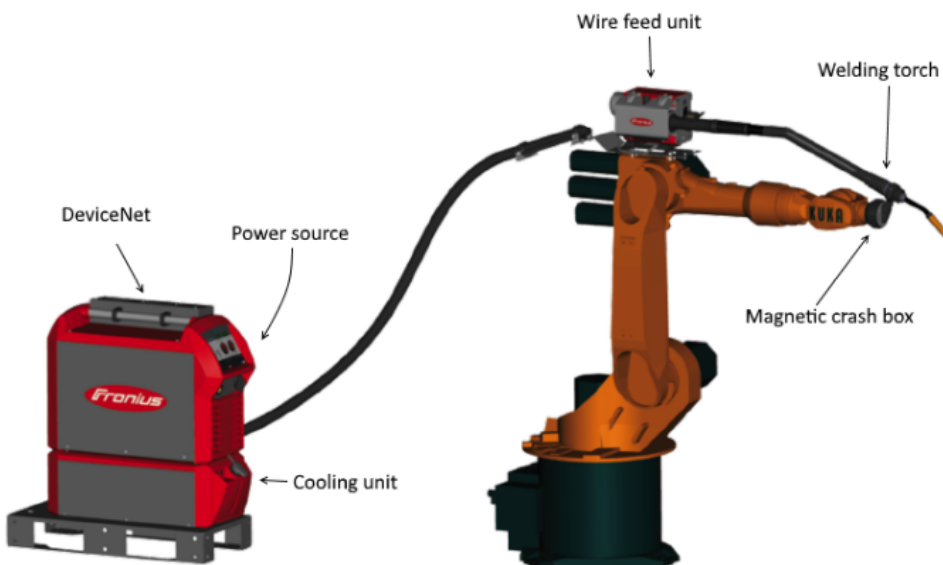


Figure 4.1: Fronius welding system components, Fronius International GmbH (2010)

Power Source

The power source contains a central control and regulation unit which is coupled with a digital signal processor, controlling the entire welding process. Process parameter readings are taken continuously during welding to ensure that the system responds immediately to changes. It supports currents ranging from 0 - 500 amps. The best welding parameters are calculated by the power source based on the Synergic control panel and certain general items of data such as sheet thickness, filler metal, wire diameter and protective gas shield. All the parameters may also be manually defined. Whether the parameters are calculated automatically or chosen manually depends on the selected process mode of MAG welding. The power source supports two process modes: Standard Synergic MAG welding and standard manual MAG welding. Synergic MAG welding is only available in combination with the Synergic control panel. Wire feed speed is, as discussed previously, the main parameter of interest considering the models equation input. Thus, the process mode Synergic MAG welding is used in combination with adjusting the wire feed speed on the Synergic control panel. This means that the Synergic function will adjust all other parameters automatically based on the chosen wire feed speed. The other parameters include sheet thickness, welding current and voltage.

The power source supports three different operating modes, each targeting a specific application. For tacking-work, shorts seams and automatic/robot welding, a "2-step mode" is recommended. A "4-step mode" is more suitable for longer weld seams when welding manually with a stick. In regards to robot welding, the "2-step mode" and "4-step mode" provide no practical difference. The arc is immediately set to the desired level and maintained for as long as needed. The last mode is a "Special 4-step mode", where the arc starts at a relatively low power (low current) for stabilization, then increased to a desired level. This mode is suitable for welding in high power ranges. The "2-step mode" is used for all tests for this thesis.

Crash Box, Wire Feed Unit and Torch

The main function of the magnetic crash box is to provide a fail safe reaction in the event of a collision. If a collision happens, the crash box sends a signal to the robot control which stops the robot immediately. It also provides a protection for the torch as the magnetically fastened box detaches to prevent damage.

The wire feed unit has an operating range of 0.5-25 m/min (8.3-416.7 mm/s) well within the required speed for the testing. It supports wire diameters ranging from 0.8-1.6 mm.

The torch has a gas nozzle diameter of 19 mm.

Robot Interface

The Fronius welding system is connected to the KUKA KR C4 control cabinet. Communication between the two interfaces occurs through the DeviceNet module. DeviceNET is an open CAN-based system. A variety of I/O signals is sent between the KR C4 controller and the power source. Input signals from the robot controller to the power source include welding start/stop, robot ready signal, gas test, wire feed value, operating modes (bit 0, 1

and 2), source error reset, value for power (voltage), torch blow out, arc length correction value, dynamic correction and burn-back value. In order to control the welding current from the power source and enable welding from within a robot program, the robot ready signal, the operating mode bit 1 and source error reset signal have to be set to set high. This is done on the digital I/O menu on the smartPAD. If the robots resets the source error reset signal during welding, the power source end the welding and the error ”-St oP-” appear on the power source display. When operating mode bit 1 is set high the operator will be allowed to use the power source control panel to adjust parameters. The source error reset signal is sent after a fault occurs and have been rectified. If this signal is set high, any error that occurs is reset immediately after it has been rectified.

4.4 Software

This section introduces a few acronyms and abbreviations related to KUKA specific technology. A list of them are given here for comprehension:

- KCP = KUKA Control Panel
- KRL = KUKA Robot Language
- KSS = KUKA System Software
- KUKA smartHMI = Name of the graphical user interface for the KR C4 controller
- KUKA smartPAD = Name of the KCP

KUKA.Sim and KUKA.OfficeLite

The robot motion statement program generated by 3DAutomate is not yet understandable for the KR C4 controller. It has to be translated into KRL modules. A module consist of two files: A source code file (SRC file) and a data list file (DAT file). The SRC file contain the program code while the DAT file contain permanent data and point coordinates. Dividing the module into two files keeps the statement file clean and easily readable. It is, however, possible to merge them into one file. Generation of the module occurs through interaction between *KUKA.Sim* and *KUKA.OfficeLite*, shown in figure 4.2.

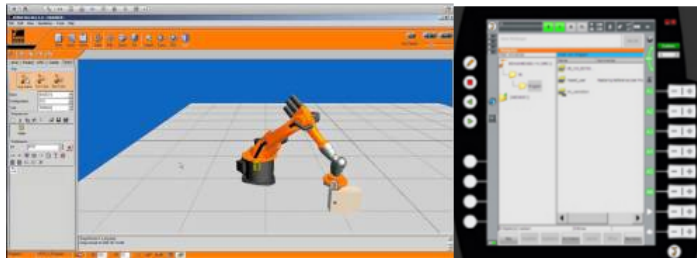


Figure 4.2: KUKA.Sim and KUKA.OfficeLite

KUKA.Sim is built on the simulation and visualization software suite from Visual Components. This means that the Python scripts and model created in 3DAutomate could have been created in or simply transferred to KUKA.Sim. Small changes may occur due to the fact that KUKA.Sim is not up to date with the newest version of 3DAutomate. The main difference between the two is the I/O functionality with KUKA.OfficeLite. KUKA.OfficeLite is a Virtual Robot Controller (VRC) nearly identical to the KUKA System Software for KR C4. It shows the KUKA smartPAD with the KUKA smartHMI. The RSL file in KUKA.Sim can be downloaded to the VRC when the connection between the two software is established. The RSL program file is then translated into an executable KRL language module. Running the program on the VRC will give a realistic simulation of the programmed operation as it corresponds exactly to the behavior of the robot in real life. The KRL module can now be transferred via a memory stick to the KR C4 controller for real life execution.

KUKA.ArcTech

ArcTech is an arc welding add-on technology package for the KSS. Programming simple tasks is easy through the ArcTech interface on the KUKA smartPAD. Some of the features ArcTech provide includes configuration of ignition and weld fault strategies, selection of defined data sets via inline forms, mechanical weaving for large gap welding and configuration of the weld power source.

The programming of welding have been done by utilizing the *ArcOn* and *ArcOff* commands in ArcTech. These two commands has several inputs. When programming the ArcOn command, the programmer can choose welding speed, ignition-, weld-, and weave parameters. The ArcOff command sets the endpoint of welding as well as end crater parameters. Some of the selectable options in the ArcOn/ArcOff commands are in form of parameter sets. These parameters sets are only programmed through WorkVisual and are basically a configuration of I/O communication with the power source. A function of the sets is indicating the viable ranges of wire feed speed, current, voltage, assigning parameters to a process and also how to handle information between the power source and KSS. Different I/O signals between the power source and KR C4 controller have been discussed in the previous section. It is not of importance to this thesis to describe this more in-depth. The key point is that the desired values for wire feed, welding speed and weaving parameters are available and selectable at the power source or in the ArcOn/ArcOff commands.

The mechanical weaving functionality in ArcTch is extensively used in the laboratory tests. A tab denoted "Weaving" appear during insertion of the ArcOn-command containing the following parameters: Pattern, length, deflection and angle. In addition to a no weaving selection, four predefined weave patterns are selectable. These are shown in figure 4.3.

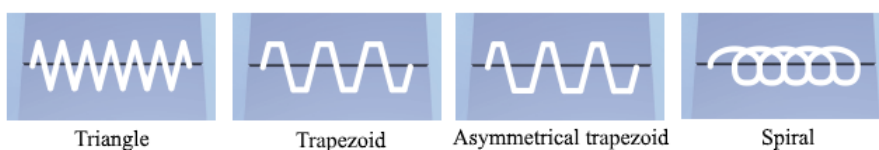


Figure 4.3: Weave patterns in ArcTech

The shape of the pattern is dependent on the weld velocity, weave length and amplitude. One weave length corresponds to one oscillation from start to end. The deflection corresponds to the weaving amplitude from chapter 3. Angle defines the weave plane angle relative to the tool orientation. Weaving in this thesis will always be in -90 or 90 degree configuration due to the tool coordinate system orientation. A -90 or 90 degree weave plane angle corresponds to weaving in the XY-plane at the base metal. Weaving frequency is calculated by the following formula:

$$\text{Weave frequency [Hz]} = \frac{\text{Weld speed [mm/s]}}{\text{Weave length [mm]}} = \frac{\text{Weld speed [m/min]} * 1000}{\text{Weave length [mm]} * 60} \quad (4.1)$$

While triangle, trapezoid and asymmetrical trapezoid weaving is signified by deflection of the welding torch in one direction, spiral weaving has deflection in two directions. When utilizing the spiral weave pattern, the weaving amplitude is given by half the length.

The parameters in ArcTech that will be adjusted in the laboratory implementation is the wait time after ignition, gas preflow time and all parameters related to weaving in the ArcOn command, and the end crater time in the ArcOff command. Figure 4.4 show a weaving frequency diagram provided by KUKA. It states the permitted frequencies given in terms welding speed on the x-axis and weave length on the y-axis. See equation 4.1 or how to calculate the frequency. All testing will be set within the frequency limits of the given diagram.

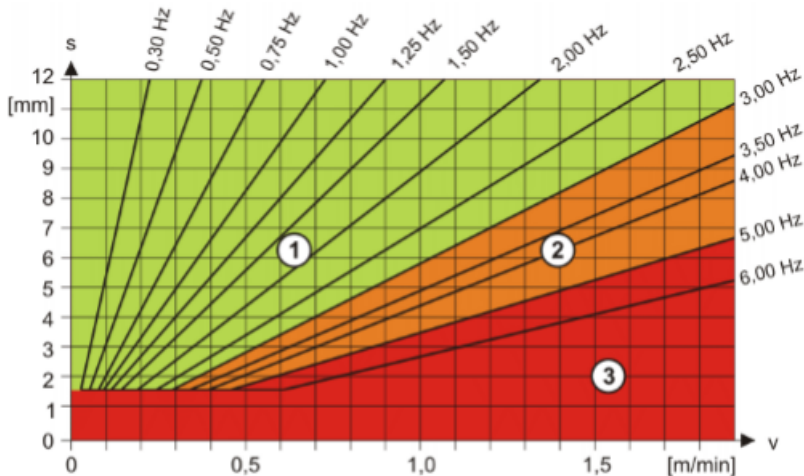


Figure 4.4: Weave frequency diagram, 1 - Permissible range, 2 - Critical range, 3 - Prohibited range KUKA Roboter GmbH (2012)

The ignition wait time is the duration of time from when the arc is struck until the robot translational movement is initiated. A too short ignition wait time will result in a unstable arc and most likely weld spatter. It is important to obtain a stable arc to achieve a uniform weld seam. The end crater time represents how long the arc is held at the end position before turned off. Gas preflow time is the elapsed time between gas supply is started to the

power source starts the actual welding. A too short gas preflow time can result in a poorly shielded weld puddle and possibly atmospheric contamination.

4.5 Overall Information Flow

Figure 4.5 show the overall information flow from start to finish. The different substeps in the visualization step have been discussed in more detail previously. For the multiple-pass visualization step, see section 3.3. The scripts and logic behind the offline programming is explained in section 3.4, while generation of the KRL program with KUKA.OfficeLite is discussed in section 4.4. Section 4.4 also explain the programming corrections done on the KR C4 controller and the I/O communication with the Fronius power source.

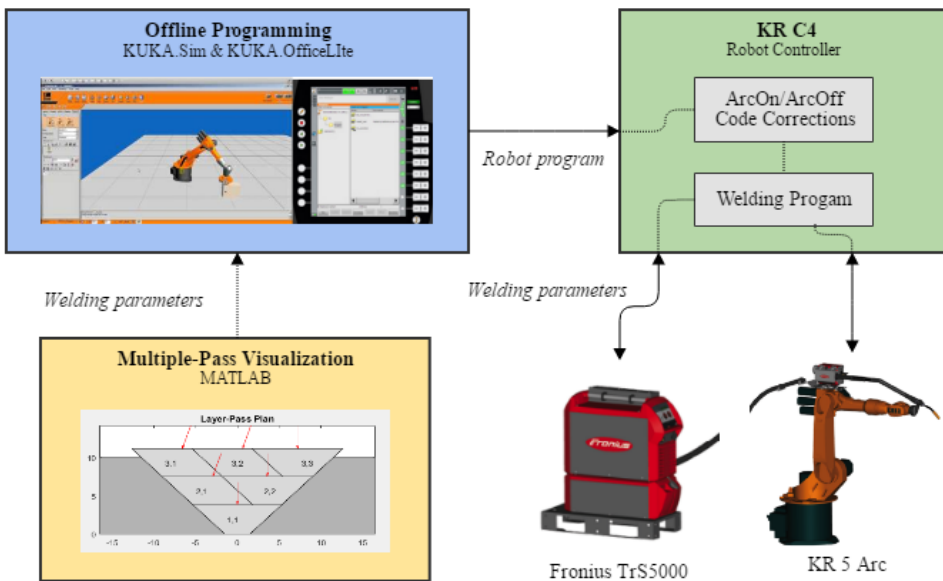


Figure 4.5: Overall information flow

An alternative and more ingenious information flow path is possible with the introduction of the software *WorkVisual* by KUKA. This software is the engineering environment for KR C4 controlled robotic cells. It is an offline development, online diagnosis and maintenance environment. Besides administrating all controllers in a robot cell, its main functions include deploying projects to the robot controller. KUKA.OfficeLite can be used in conjunction with WorkVisual, giving a possibility for program transmission without the memory stick. The prerequisites for this is a network connection to the real robot controller. Deploying and activating projects from WorkVisual to the KR C4 controller can also affect the safety configurations of the cell. Due to lack of experience in WorkVisual and time constraints, the use of the software was omitted in this thesis.

4.6 Experimental Setup

The workshop employees at IPK have fabricated all metal components needed for the testing. That includes a fixture for the base metal joints, backing plates, as well as the joints. The fixture consists of two fasteners bolted to a metal base plate. The plate is screwed onto a stand and electrically grounded by a cable and a clamp connected to the Fronius power source, see figure 4.6. The metal plate ensures good electrical conduction while the fasteners provide secure fastening of the base metal.



Figure 4.6: Laboratory test setup

A general structural steel with designation ST-37 according to the European standard NS-EN 10025-1 is used as the base metal. Four different types of joints with varying geometry have been fabricated. Two different bevel angles with or without a root face constitutes the different variants. Table 4.1 provides the geometrical data for the butt joints, designated with name type I, II, III and IV. Figure 4.7 also show all test specimens. There are all together four sets of type I, three sets of type II, four sets of type III and three sets of type IV.

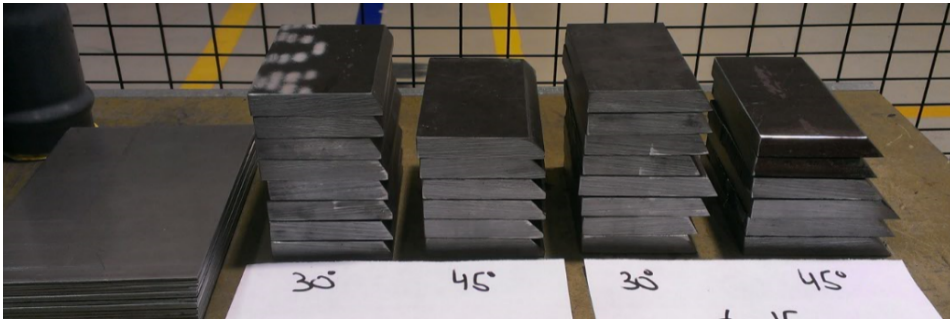


Figure 4.7: Butt joint test specimen overview. From left to right: backing plates, joint type II-IV-I-III

	Type I	Type II	Type III	Type IV
Root gap [mm]	3	3	3	3
Root face [mm]	0	5	0	5
Bevel angle [deg]	30	30	45	45
Thickness [mm]	15	15	15	15

Table 4.1: Butt joint types geometrical data

The power source is configured for DCEP MAG welding with a shielding gas mixture of argon and carbon dioxide (98%-2%). Filler wire is type OK Autrod 12.64 provided by ESAB. This is a copper-silicon Mn-G4Si1 wire electrode for gas metal arc welding of unalloyed steels. Manganese (Mn) and silicon (Si) are deoxidizing elements which reduce chances of porosity caused by entrapped oxygen. The diameter of the electrode filler wire is 1 mm. According to the data sheet of this filler wire provided by ESAB (2004), the metal recovery, or deposition efficiency, of the 1mm electrode is 0.96.

Relatively low currents will be used. Based on the research given in chapter 2.3 on metal transfer modes in argon carbon dioxide mixtures for small diameter electrodes, the metal transfer mode in the laboratory implementation will mainly be short circuiting. Globular transfer may occur, but the current is kept to low to experience spray transfer. Furthermore, the wire stick out length is configured to approximately 15mm.

4.7 Test Plan

Preliminary Tests

The testing is started off by experimenting with different parameters while welding on flat metal plates in order to obtain good quality weld beads later. Certain parameters such as welding speed are held constant, whereas other parameters (weave frequency, weave amplitude and weave pattern) will be isolated and tested. The theory from chapter 2 and 3 will be used as a basis for the preliminary tests. These results will become the basis for the multiple-pass welding of the variable V-butt joints.

The preliminary test includes different ignition wait times, variable end crater times, weaving at various frequencies combined with various amplitudes and also different weave patterns, and at varying feed speed. The goal with these tests is to form an idea of usable parameter sets that create stable welding and quality beads, which can be used to weld the butt joints.

Butt Joint Weld Sequence

After the preliminary tests, experiments on butt joint type I-IV will commence. As can be seen in the next chapter, a three layer four passes sequence is the initial test for butt joint type I. That is the 60 degree groove angle joint with a zero root face. Based on the results, adjustments are made to the model and further testing on the rest of the butt joint types continues.

To determine the accuracy of the mathematical model, the welded sections will be cut in two to inspect the cross-sectional area. The inspection of the cross section will reveal defects such as incomplete fusion, and help determine the welding deposition efficiency coefficient. This parameter will be adjusted as the testing progresses to compensate for the material loss. Note that to obtain the same bead area by equation 3.2, the deposition coefficient and feed speed is adjusted opposite to each other.

For joint type II and IV, different root pass parameters will be tested. As the two joint types have the same geometry in regards to root face and root gap, the results will be comparable.

Final Tests

One set of block type II, III and IV is kept for a final test. The final test is a fully automated test where the whole welding sequence is done in one session. That is no interference from the operator in between passes to inspect the weld beads. The tests includes root pass, filler passes and cover pass. Parameters will be chosen based on all testing up until this point. The cross sections of the final tests will be visually inspected by third party welding experts. Due to the nature of wire feed speed parameter selection, it will not be changed between root and filler passes or between filler and cover pass on the final tests. Wire feed speed is adjusted on the power source itself and will not be adjusted during welding execution.

Results & Analysis

This chapter outline the results of all weld tests carried out during the laboratory implementation of this thesis. All tests done will be given with all relevant parameters. As test parameters chosen for the different butt joints are based upon previous values, it is recommended to read this chapter in proper sequence from start to finish. Results will be discussed on-the-fly, with some conclusions and remarks in the last section.

Section 5.1 show and discusses the results of the preliminary tests. An important remark regarding the preliminary test was a later discovery of an error in the mathematical model. The model was based on equation 3.1 for the cross-sectional area of a weld bead. However, it was changed to equation 3.2 during the first joint type I tests. The error is more carefully explained in section 5.2. For now, the key point is that some of the preliminary tests presented in this chapter was executed with half the desired wire feed speed.

Section 5.2-5.6 follow the tests in sequence from welding of butt joint type I to final tests. Some of the welded joints are taken to a third party visual inspection. The results of this is given in section 5.7. Section 5.8 features some remarks about the obtained results and source of errors.

All welding is done with the perpendicular travel angle method discussed in chapter 2.2. Also, measured value for wire feed speed is given in m/min on the power source. Therefore, the corresponding values in mm/s may be uneven decimal numbers. Both units will be used. The same applies for the robot speed which is given in m/min in the KSS.

5.1 Preliminary Tests

A total of nine test plates have been used for the preliminary parameter testing. At the moment, the feed speed v_1 and welding speed v_2 is held constant at 50 mm/s and 3 mm/s. The first test is for variable ignition wait times. Figure 5.1 show the start point of five strings with variable ignition times. The far left string has a wait time of 1.0 seconds, decreasing by 0.2 seconds until til far right one with a 0.2 seconds ignition wait time. A 1.0 seconds wait time created a bulb at the starting point of the weld string, while second wait time was insufficient in the sense that the arc was not completely stable when the torch movement was initiated. 0.6 seconds wait time turned out to be a satisfactory mean value. The arc was stable at the start of the translational movement with no considerable bulb at the starting point. Thus, a 0.6 second ignition wait time is implemented for the rest of the test.

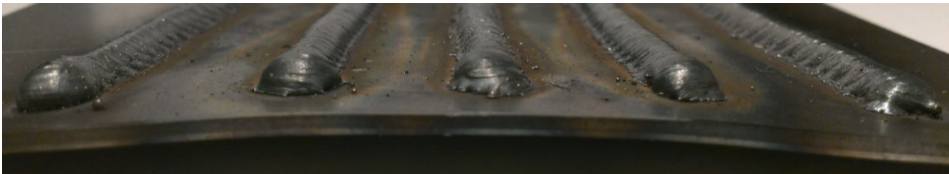


Figure 5.1: Preliminary test plates: Ignition wait time. From left to right: 0.2 - 0.4 - 0.6 - 0.8 - 1.0 [sec]

Three different end crater times is shown in figure 5.2. The three craters on the left side corresponds to the three nearest beads on the right side. The goal of the test was to determine if different end crater times would affect the crater at the termination of the arc welds. All three beads show a declining weld face at the end of the bead. Extending the end crater time did not significantly reduce the beads end point height. However, extended end crater times resulted in a more apparent crater and a deeper penetration. This could be seen on the back side of the plates. In order to fill the end craters, the travel direction should be reversed at the end of the pass. This will not, however, be incorporated to this experiment. End crater times will from hereon be held constant at 0.1 seconds for the rest of all tests.

Figure 5.3 shows the geometry for the different inbuilt weave patterns in ArcTech. A top view of the bead in addition a view of each end is given to illustrate both the width and height of each bead. On the left side a triangle weave bead is laid, then a trapezoid weave bead, asymmetrical weave and lastly a spiral weave bead on the right side. All of them was laid with a 1 Hz weave frequency and an amplitude of 3 mm. The welding bead show that triangle weaving produce a medium width medium height bead whereas both trapezoid and asymmetrical trapezoid produce a wider bead at the expense of bead height. Spiral weaving produce a more narrow and high bead. Notice also the of the rightward shift of the bead by the asymmetrical weaving pattern. The torch spends more time on the right side, causing this shift.

A few different weaving amplitude- and frequency combinations have been tested for the same feed speed and welding speed. In addition to the combinations already shown, a 2 mm and 4 mm amplitude is tested with different frequencies. Figure 5.4a show three

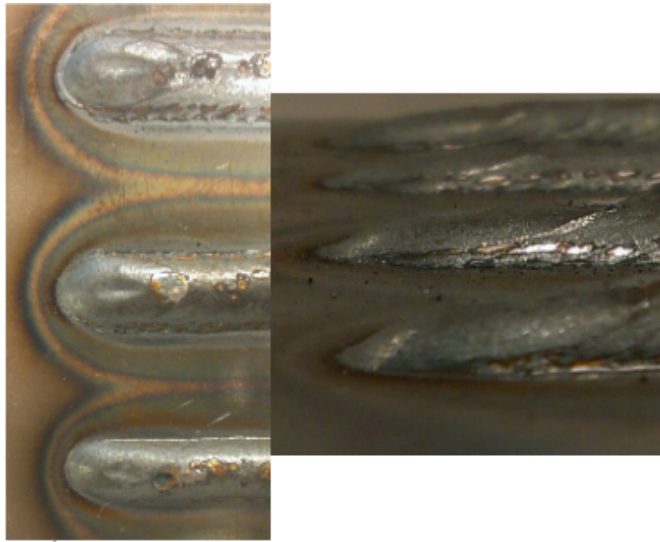


Figure 5.2: Preliminary plate tests: End crater time. From bottom to top: 0.1 - 0.3 - 0.5 [sec]

triangle weave beads with a amplitude 4mm and frequencies 1.25 Hz, 1.5 Hz and 1.75 Hz. A 1.75 Hz frequency at this amplitude resulted in a unstable bead path as the oscillation was too rapid. The unstable weld path seen to the right in figure 5.4a indicates a too fast travel speed for these parameters. 1.25 Hz and 1.5 Hz produced uniform beads. At a 2mm amplitude, frequencies ranging from 0.5 Hz to 2.5 Hz have been tested as shown in figure 5.4b. All frequencies except 0.5Hz created seemingly adequate results. Weaving at 0.5Hz produced a sinusoidal weld bead as the oscillation was to slow.

Furthermore, weaving amplitudes from 1.0 to 4.0 mm was tested at a frequency of 1 Hz, shown in figure 5.5. All four passes had stable arcs during the weld and produced adequate weld strings. As the amplitude increases, the weld bead shifts from a narrow high bead to a low wide bead. At 4mm, the sinusoidal shape is starting to appear. Wider weaving than this will not be applicable for these parameters.

As an mathematical error was discovered in the model, the need for testing higher wire feed speeds became necessary. Figure 5.6a show a set of four weld beads with variable feed speed. All passes are laid using triangle weaving at 1 Hz with an amplitude of 3 mm. The feed varies from 4.0 m/min to 7.0 m/min, respectively 66.7 mm/s to 116.7 mm/s, incremented by 1.0m/min for each pass. These wire feed speeds corresponds to a current of 75, 105, 137 and 154 amps and voltages between 16.5 and 19 volts. All passes produced uniform beads without spatter. The slightly sinusoidal pattern at the first and third bead was due to an unstable arc ignition at the beginning. Why this occurred during these two passes is difficult to determine. A possible cause may have been a two long wire extension at the start of the weld. The increase in wire feed speed/ampereage is easily seen by the increase in bead volume. Finding the right feed speed in regards to the cross-sectional area of the bead will help determine the deposition efficiency coefficient a_H .

Further testing of higher feed speeds included weaving amplitude variations and also

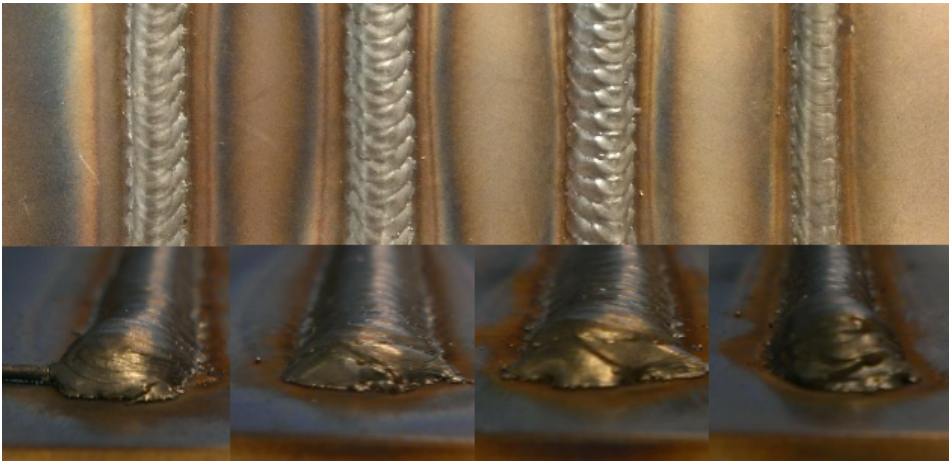


Figure 5.3: Preliminary test plates: Weave pattern. From left to right: Triangle - Trapezoid - Asymmetrical trapezoid - Spiral

one pass with spiral weaving. During the welding of beads given by figure 5.6b, feed speed was held constant at 6.0 m/min (100 mm/s) with a weaving frequency of 1 Hz. The weaving amplitude for the different passes, given from left to right, is 2 mm, 4 mm, 5 mm and 2 mm. None of the beads showed any sign of spatter and all beads are uniform. No signs of weaving patterns is seen on the weld face. It is probable that higher weaving amplitudes can be implemented at the respective feed speed.

A cover pass is generally wider than intermediate filler passes in order to tie inn all subsequent beads. Therefore, cover passes are tested at lower welding speeds to enable wider weaving. The distance between weld toes of the joint butt ranges from roughly 14.5 mm for butt joint type II to 23 mm for butt joint type III. Figure 5.7a show the first parameters test. These two beads is made with a 2.0 mm/s welding speed, 90 mm/s wire feed speed and triangle weaving at 1 Hz. The left side bead has an weaving amplitude of 7 mm, which produces an approximately 16 mm wide bead. A concave weld bead without spatter became the end results. Also, the arc was stable throughout the process, whereas an unstable arc was observed for the right side bead. At a weaving width of 10mm, the rapid oscillation led to pulsating arc ignition and a non uniform weld face. This can be slightly seen on the right side bead in figure 5.7a. A lower weaving frequency have to be implemented for this amplitude. Figure 5.7b show the second set of cover passes. Both passes is laid with a 90 mm/s feed speed and triangle weaving, but with varying welding speeds and frequencies. The left side pass is laid at a 1.5 mm/s welding speed, 0.5 Hz frequency and a 11 mm weaving amplitude. These process parameters appeared to produce an adequate end result. A stable arc was observed throughout the pass and a slightly convex bead was produced at a 25 mm toe-to-toe distance. The right side pass is done at a 0.75 Hz frequency and amplitude of 10.5 mm. The increase in frequency did not affect the stability of the arc. A convex uniform bead at a 24 mm toe-to-toe width was the result.



(a) Triangle weave, amplitude = 4mm. From left to right: 1.25 - 1.5 - 1.75 [Hz]

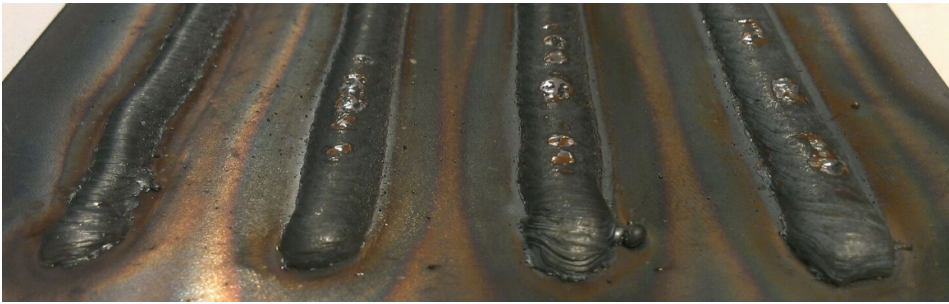


(b) Triangle weave, amplitude = 2mm. From left to right: 0.5 - 1.0 - 1.5 - 2.0 - 2.5 [Hz]

Figure 5.4: Preliminary plate tests: Various frequencies and amplitude



Figure 5.5: Preliminary plate tests: Various amplitude at 1 Hz triangle weave. From left to right: 1.0 - 2.0 - 3.0 - 4.0 [mm]

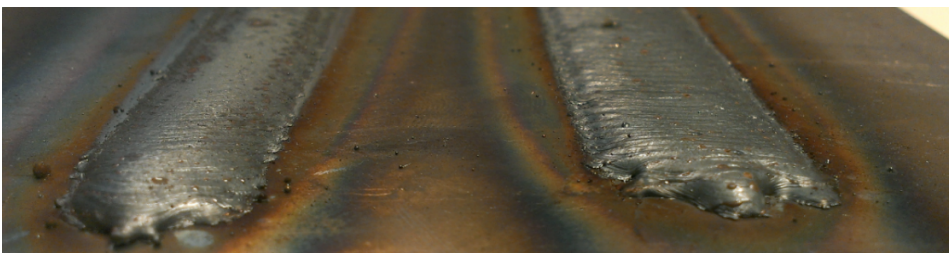


(a) Feed speed from left to right: 4.0 - 5.0 - 6.0 - 7.0 [m/min]

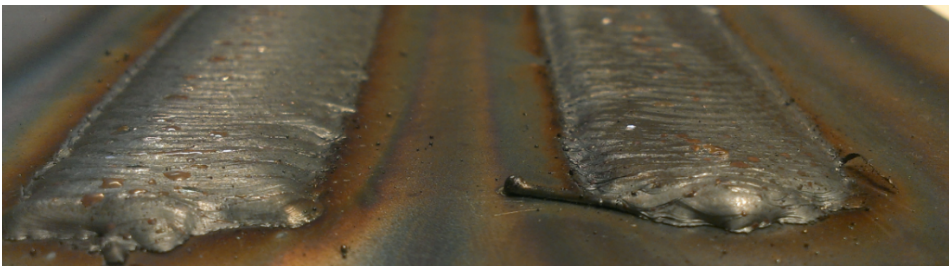


(b) Feed speed 6.0 m/min. Amplitude: 2 - 4 - 5 - 4 [mm]. Weave pattern: TW - TW - TW - SW

Figure 5.6: Preliminary plate tests: Feed speed



(a) Welding speed = 2.0mm/s. Amplitude = 7 and 10mm. Frequency = 1Hz



(b) Welding speed = 1.5mm/s. Amplitude = 11.0 and 10.5mm. Frequency = 0.5 and 0.75Hz

Figure 5.7: Preliminary plate tests: Cover pass

5.2 Butt Joint Type I

Welding plates versus V-groove butt joints is two different things. How shielding gas is dispersed around the weld puddle will deviate, the side walls will influence how the bead is laid in addition to other various process parameters. In the end, results obtained on weld plates may not be applicable to joint welding. Further adjustments may therefore be needed throughout the test sequence. For the sake of simplicity, a short notation for weaving type will be used in tables and figures.

- ATRAW = Asymmetrical trapezoid weaving
- SPIW = Spiral weaving
- TRAW = Trapezoid weaving
- TRIW = Triangle weaving

Initial alignment of the multiple-pass planning model started with a set of three layer four passes tests on butt joint type I. The original model produced the schematic representation given by figure 5.8, showing a nearly filled groove. Both parallelogram and trapezium beads are featured in this configuration, providing a good basis for assessing the model.

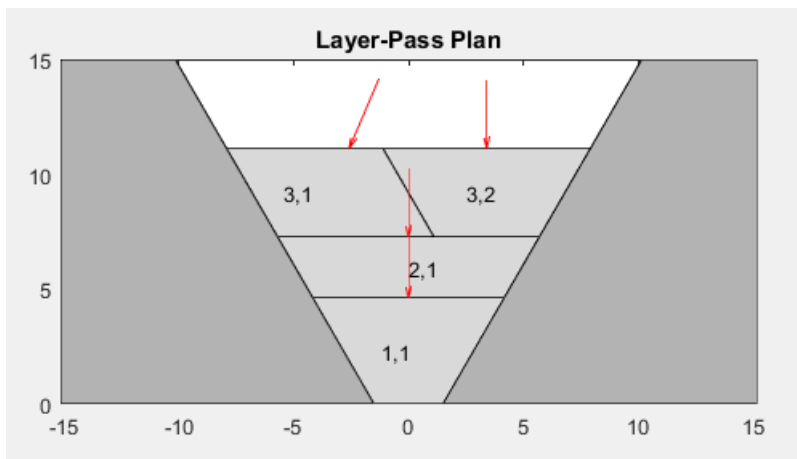


Figure 5.8: Butt joint type I: Schematic layout

The two first tests was set to triangle weaving at 1 Hz with different values for weaving factor δ . Parameters for these two test are given by table 5.1 and 5.2 The only difference in these two tables are the weaving amplitude. Decreasing values for δ generates wider weaving. The third test was done with spiral weaving at 1 Hz, also with $\delta = 2.0$, giving the same parameters as in table 5.1. Figure 5.9a-5.9c show the experimental result.

At first sight, it is easily seen that too little metal are deposited to the joint. About half of the desired weld volume is achieved, leaving a gap between the last two passes and

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W
1	1	50	3	0	4.62	0	2.2
2	1	50	3	0	7.27	0	3.7
3	1	50	3	-2.59	11.12	23.1	2.9
3	2	50	3	3.40	11.12	0	2.5

Table 5.1: Butt joint type I: 3 layer 4 pass test parameters for $\delta = 2.0$.

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W
1	1	50	3	0	4.62	0	2.7
2	1	50	3	0	7.27	0	4.2
3	1	50	3	-2.59	11.12	23.1	3.4
3	2	50	3	3.40	11.12	0	3.0

Table 5.2: Butt joint type I: 3 layer 4 pass test parameters for $\delta = 1.5$.

almost no fusion between the weld beads. Based on visual inspection, the bead has not penetrated sufficiently as the side wall lines are distinct. All the results points towards a too low current or feeding rate. Weaving with $\delta = 1.5$ produced a string with coarse edges. The arc was also unstable during the execution of the welding. This was not surprising as the preliminary test also showed the same results for weaving at 1 Hz with weaving amplitude 4 mm. Spiral weaving produced as expected more narrow beads. Still, too little filler wire is deposited.

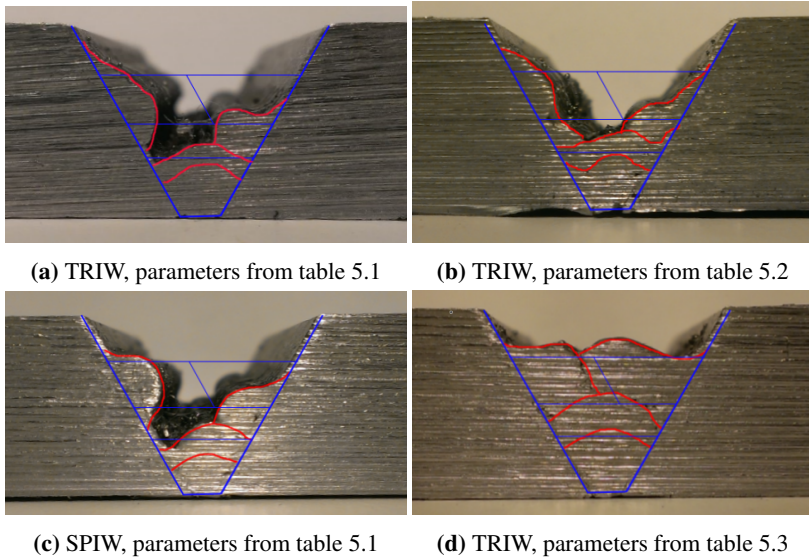


Figure 5.9: Butt joint type I: Test results

As the three first tests gave results not comparable to the model visualization, a revision of the equations given in section 3.2 was done. This showed an error in the calculation of cross-sectional bead area S , and a change from equation 3.1 to equation 3.2 was carried out. The introduced deposition efficiency coefficient was set to $a_H = 1$ and the feed speed doubled to 100 mm/s to generate the same schematic output as shown in figure 5.8. With new parameters given by table 5.3, a more promising results was obtained, shown by figure 5.9d. Now, the is groove filled with metal according to the visual model. In fact, filler metal appear to exceed the calculated area. Better fusion is also observed as the sidewall lines are no longer visible.

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W
1	1	100	3	0	4.62	0	2.2
2	1	100	3	0	7.27	0	3.7
3	1	100	3	-2.59	11.12	23.1	2.9
3	2	100	3	3.40	11.12	0	2.5

Table 5.3: Butt joint type I: Adjusted 3 layer 4 passes test parameters for $\delta = 2.0$.

From now on, the weaving factor δ is set to 2.0 as this produced adequate results for the last completed test. With the new equation for S implemented, testing of butt joint type II commences. This geometry includes root face, raising the necessity of a root pass. A variety of passes from welding of butt joint type I is shown in figure 5.10.

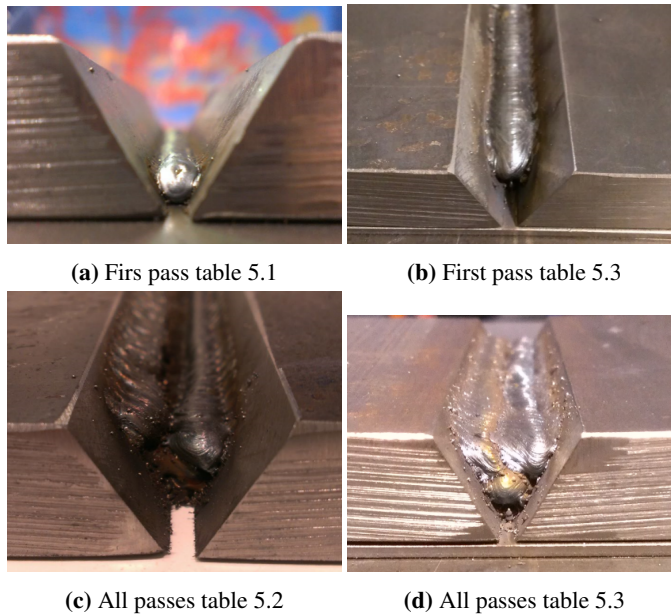


Figure 5.10: Butt joint type I: Miscellaneous passes

5.3 Butt Joint Type II

Three sets of tests have been conducted with butt joint type II in preparation for the final sequence. As the model is now roughly adjusted to fill the groove, testing will seek to tune the right value for a_H . The feed speed will be slightly reduced as the last test showed an exceeded area. This is compensated by increasing the deposition efficiency coefficient. A efficiency coefficient above one is a peculiar situation. It is reported that deposition efficiencies up to 200% is possible for high efficiency electrodes. These electrodes have metallic components such as iron powder added to the covering of the electrode. The iron powder can be added the the welded metal, increasing deposition efficiency, Davies (2003). As the electrode used in this experiment is not coated, the peculiar case of the high deposition efficiency coefficient is for now attributed to uncertainties related to the manufactured joints and positioning of the pieces. A discussion of this will follow in later sections. Figure 5.11 show the schematic representation of the desired bead geometry. Note that this represents only the filler passes. The root pass is not included.

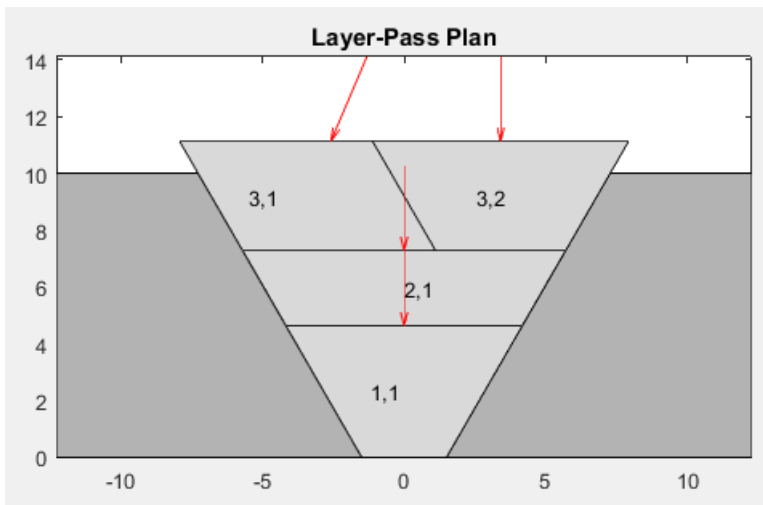


Figure 5.11: Butt joint type II: Schematic layout

The three tests conducted on this butt joint type is denoted 2-1, 2-2, and 2-3. As the different tests will have variable weaving types and weaving frequencies within the welding sequence, two extra columns are added to the test parameter tables. They are denoted *Type* and *Hz*, equivalent to the weaving pattern type and the frequency of weaving.

2-1

Test block 2-1 contains the first root pass trial in the experiments. The feed speed is lowered during this pass since the cross-sectional area is less than other bead areas. It is initially set to 4 m/min (66.7 mm/s) with an 1 mm weaving amplitude at 1 Hz. These parameters produced the root pass shown in figure 5.12. A bead slightly higher than the

root face wall can be seen. A lower feed speed will therefore be chosen for 2-2. The filler passes is set with a feed speed of 5.5 m/min (91.7 mm/s), which corresponds to a current of about 108 A in synergic mode. The lowered feed speed is offset by setting $a_H = 1.09$. All other parameters are given by table 5.4.

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
Root	-	66.7	3	0	0	0	1.0	TRIW	1.0
1	1	91.7	3	0	4.62	0	2.2	TRIW	1.0
2	1	91.7	3	0	7.27	0	3.7	TRIW	1.0
3	1	91.7	3	-2.59	11.12	23.1	2.9	TRIW	1.0
3	2	91.7	3	3.40	11.12	0	2.5	TRIW	1.0

Table 5.4: Butt joint type II: 2-1 test parameters

The resulting joint from test 2-1 is shown in figure 5.13a. Good penetration was obtained and sufficient fusion with the sidewalls was observed. However, a gap between the two passes in layer three is clearly visible. The first pass in layer three is shifted to the left, leaving almost no fusion between the passes in the respective layer.

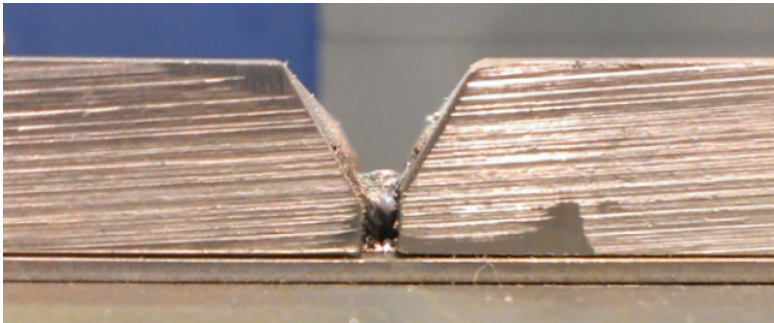


Figure 5.12: Butt joint type II: 2-1 root pass

2-2

The following changes are made from test 2-1: The feed speed is lowered to 3.5 m/min (58.3 mm/s) for the root pass in order to create a lower bead. The weaving factor δ is set to 1.5 instead of 2.0 to widen weaving. Since feed speed is almost doubled from the preliminary tests, wider weaving is most likely manageable. The purpose of widening the weaving is to get the two passes in layer three to connect and produce a more flat weld face. All other test parameters for test 2-2 are given in table 5.5.

The resulting welded joint is shown in figure 5.13b. A gap between pass one and two in layer three can still be seen, though slightly less than for 2-1. The lowered feed speed for the root pass gave a lower and more desirable bead height. Some lack of fusion can be seen on the left side of the root face in figure 5.13b. This is due to a poorly calibrated

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
Root	-	58.3	3	0	0	0	1.0	TRIW	1.0
1	1	91.7	3	0	4.62	0	2.7	TRIW	1.0
2	1	91.7	3	0	7.27	0	4.2	TRIW	1.0
3	1	91.7	3	-2.59	11.12	23.1	3.4	TRIW	1.0
3	2	91.7	3	3.40	11.12	0	3.0	TRIW	1.0

Table 5.5: Butt joint type II: 2-2 test parameters

work piece, leaving the robot the weld a little bit to the right. The widened weaving gave coarser bead edges at the weld face.

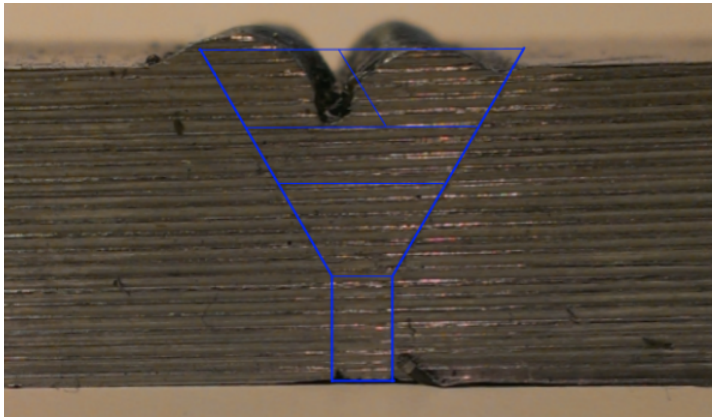
2-3

The following changes are made from test 2-2: Spiral weaving is tested for the root pass. Weaving factor δ is adjusted back to 2. To compensate for the gap between bead one and two in layer three, a asymmetrical trapezoid weaving pattern is implemented for layer three pass one. The goal of this is to shift the parallelogram bead a little to the right in order to flatten the weld face. All other test parameters is given by table 5.6.

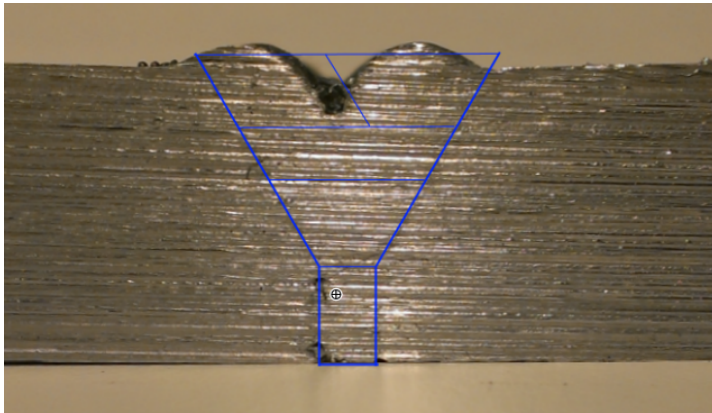
Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
Root	-	58.3	3	0	0	0	1.0	SPIW	1.0
1	1	91.7	3	0	4.62	0	2.2	TRIW	1.0
2	1	91.7	3	0	7.27	0	3.7	TRIW	1.0
3	1	91.7	3	-2.59	11.12	23.1	2.9	ATRAW	1.0
3	2	91.7	3	3.40	11.12	0	2.5	TRIW	1.0

Table 5.6: Butt joint type II: 2-3 test parameters

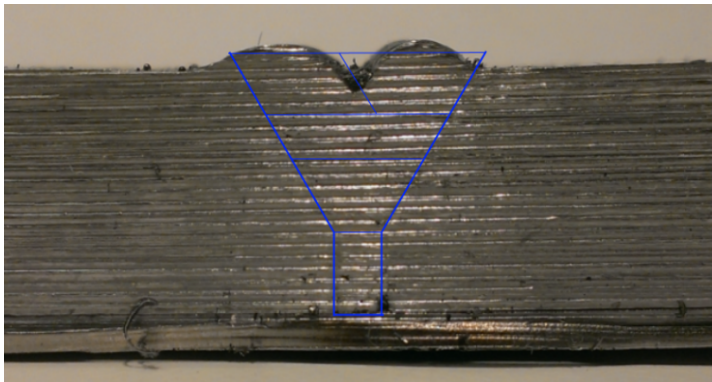
Figure 5.13c display the resulting joint. It clearly shows a reduced gap between the two beads in layer three. The feed speed seem to roughly produced the desired cross-sectional area for the weld beads. Spiral weaving on the root pass appeared to be unfit for this pass as the arc became highly unstable. For each circular motion the robot made the arc had to be reignited. This caused an arc that pulsed every second for the whole pass. Triangle weaving will therefore be used for the rest of all root passes.



(a) 2-1: Parameters from table 5.4



(b) 2-2: Parameters from table 5.5



(c) 2-3: Parameters from table 5.6

Figure 5.13: Butt joint type II: Test results

5.4 Butt Joint Type III

This joint is the largest joint in the sense that it requires the most passes among the four types. With its 90 degree 15 mm deep bevel, a four layer ten pass layout was chosen by the path planning algorithm. The schematic layout is shown in figure 5.14. Two sets of butt joint type III is tested in this section, denoted 3-1 and 3-2.

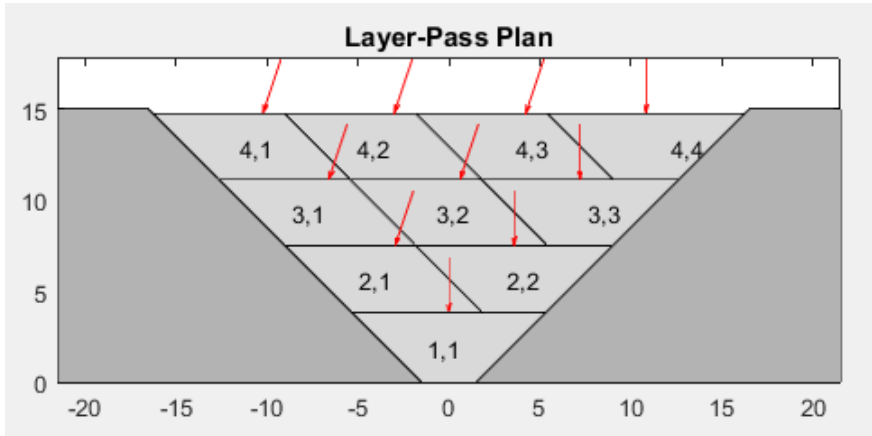


Figure 5.14: Butt joint type III: Schematic layout

3-1

The first block will have a simple setup. All weaving is set to the triangular pattern even though this created a gap between two passes in butt joint type II test 2-1 and 2-2. As this bevel has a larger bevel angle, it is tested under the same conditions to observe the effect of this. Table 5.7 show all test parameters.

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
1	1	91.7	3	0	3.83	0	3.3	TRIW	1.0
2	1	91.7	3	-2.96	7.49	18.7	3.7	TRIW	1.0
2	2	91.7	3	3.58	7.49	0	3.4	TRIW	1.0
3	1	91.7	3	-6.59	11.12	18.5	3.7	TRIW	1.0
3	2	91.7	3	0.61	11.12	18.5	3.7	TRIW	1.0
3	3	91.7	3	7.20	11.12	0	3.4	TRIW	1.0
4	1	91.7	3	-10.22	14.75	18.5	3.7	TRIW	1.0
4	2	91.7	3	-3.00	14.75	18.5	3.7	TRIW	1.0
4	3	91.7	3	4.21	14.75	18.5	3.7	TRIW	1.0
4	4	91.7	3	10.83	14.75	0	3.4	TRIW	1.0

Table 5.7: Butt joint type III: 3-1 test parameters

The welded cross section from test 3-1 is shown in figure 5.16a. No spatter was observed on the base metal and a stable arc was obtained throughout the welding. The cross-sectional area of filler metal appear to exceed the modelled area. Furthermore, there is a gap between the third parallelogram and last trapezium in the fourth layer. This is similar to what was observed during test 2-1 and 2-2. It appears like the filler wire for the parallelogram passes is deposited too far to the left. A step-by-step weld pass progression illustration can be seen in figure 5.18. Note that the pictures are taken from the opposite side resulting in a mirrored pass progression.

3-2

The following changes are made from test 3-1: Feed speed is decreased to 5.3 m/min (88.3 mm/s) to adjust the filler wire area. Hence, deposition coefficient a_H is increased to 1.13. A new weaving pattern configuration is set for all layers containing more than one pass. In these layers, asymmetrical trapezoid weaving is implemented. The weaving angle is flipped 180 degrees between the parallelograms and trapezium in such a way that the 'heavy' side of weaving is directed towards each other, as shown in figure 5.15. This done to shift the deposition area towards the middle, filling the gap observed in test 3-1. All other test parameters is given in table 5.8.

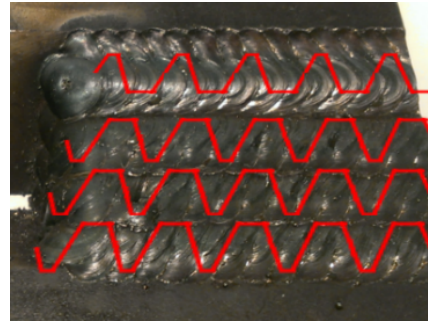


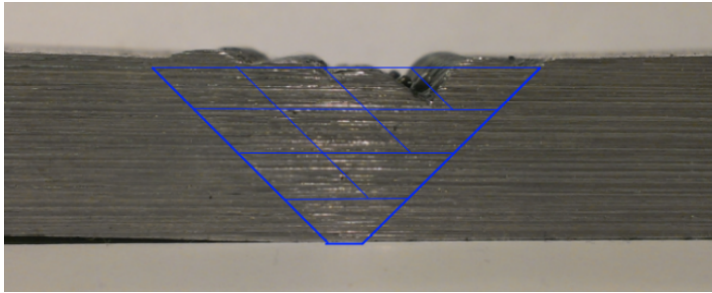
Figure 5.15: Butt joint type III: 3-1 weaving configuration

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta \theta$	W	Type	Hz
1	1	88.3	3	0	3.83	0	3.3	TRIW	1.0
2	1	88.3	3	-2.96	7.49	18.7	3.7	TRIW	1.0
2	2	88.3	3	3.58	7.49	0	3.4	ATRAW	1.0
3	1	88.3	3	-6.59	11.12	18.5	3.7	ATRAW	1.0
3	2	88.3	3	0.61	11.12	18.5	3.7	ATRAW	1.0
3	3	88.3	3	7.20	11.12	0	3.4	ATRAW	1.0
4	1	88.3	3	-10.22	14.75	18.5	3.7	ATRAW	1.0
4	2	88.3	3	-3.00	14.75	18.5	3.7	ATRAW	1.0
4	3	88.3	3	4.21	14.75	18.5	3.7	ATRAW	1.0
4	4	88.3	3	10.83	14.75	0	3.4	ATRAW	1.0

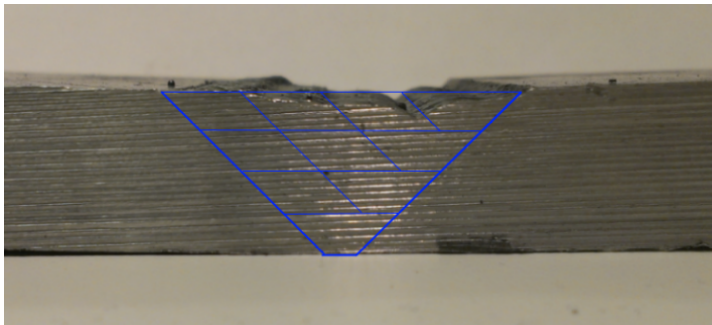
Table 5.8: Butt joint type III: 3-2 test parameters

Figure 5.16b show the final welded cross section. A more uniform weld face is clearly obtained. The gap between the last parallelogram and trapezium is reduced and the weld face is flattened. Small amounts of spatter was observed on the base metal after welding, even though the welding process was stable. Good fusion with the side walls was also

obtained. Reducing the feed gave a better approximation to the modelled weld area.



(a) 3-1: Parameters from table 5.7



(b) 3-2: Parameters from table 5.8

Figure 5.16: Butt joint type III: Test results

Due to the increased number of passes and thereby increased heat input, both test pieces was subjected to distortion of about three degrees.

Figure 5.17 show a weld face comparison between test 3-1 and 3-2. Even though 3-2 generated a more flattened top layer it came at the expense of a coarse weld face. Test 3-2, to the right in the figure, clearly show the print for each oscillation of the asymmetrical trapezoid weave. Test 3-1 has smoother edges at the weld toe areas.

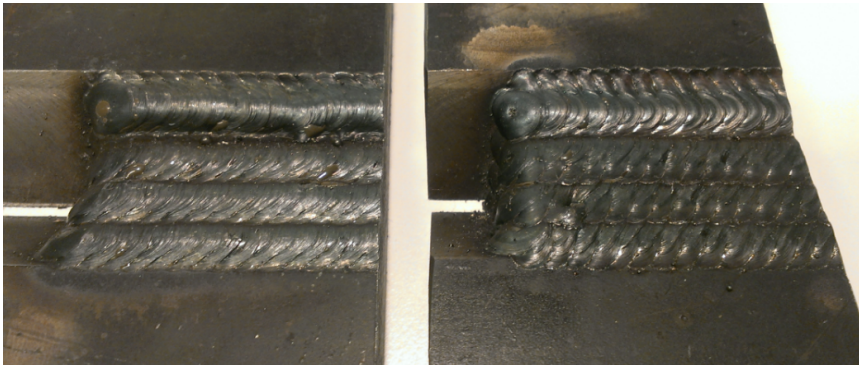


Figure 5.17: Butt joint type III: 3-1 and 3-2 weld face

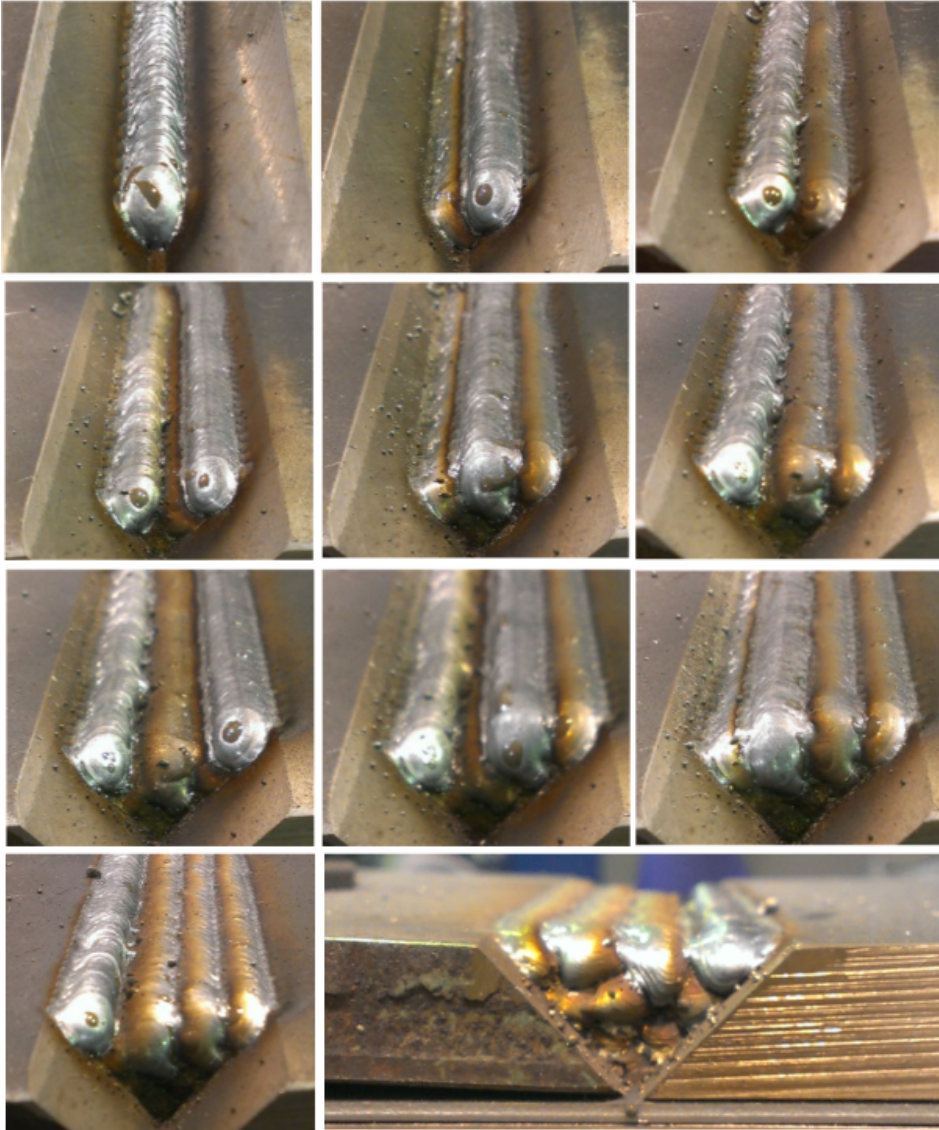


Figure 5.18: Butt joint type III: 3-1 weld pass progression

5.5 Butt Joint Type IV

Butt joint type IV is the last geometry tested in this thesis. The 15 mm thick section has a 90 degree bevel angle and a 5 mm root face. Hence, root passes are needed. Also, cover passes will be welded on these butt joints. The schematic layout for filler passes is given in figure 5.19, equivalent to the three first layers in test 3-1 and 3-2. Two sets of butt joint type IV is tested, denoted 4-1 and 4-2.

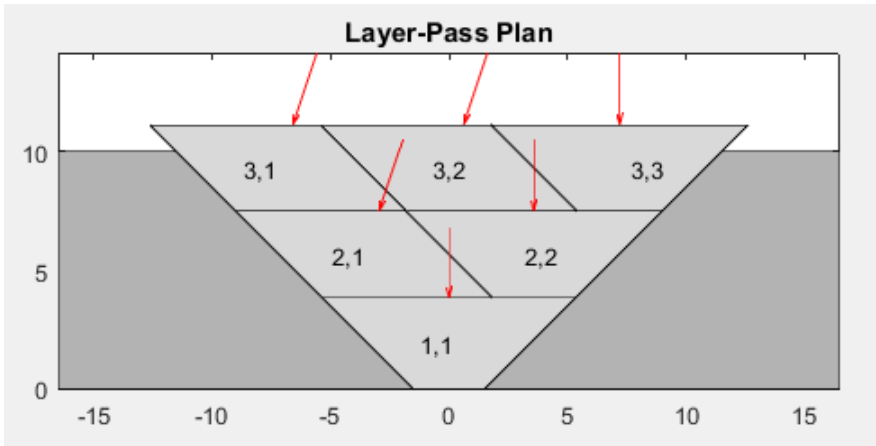


Figure 5.19: Butt joint type IV: Schematic layout

4-1

The feed speed of 3.5 m/min (58.3 mm/s) at the root pass is kept from test 2-3 as it seemed to produce a desirable bead height. However, an increased frequency of 2 Hz is implemented. For the filler passes, feed speed 5.3 m/min (88.3 mm/s) is kept from test 3-2. A cover pass is introduced for the first time of the experiments. According to the schematic layout, the third layer will slightly reach above the surface of the blocks. Previous test on block type II and III has shown higher beads at the edges with a downfall towards the middle. Therefore, a cover pass with weaving amplitude of 5 mm is implemented. This should be wide enough to tie in all three subsequent passes in layer three and produce a convex weld face. The welding speed is reduced to 0.12 m/min (2.0 mm/s) to allow a stable wide weaving. All other test parameters for 4-1 is given in table 5.9.

Figure 5.20a show the welded cross section of test 4-1. The increased root pass weaving frequency generated a too rapid robot movement, leading to a poor weld pass. The arc was unstable and poor fusion with the base metal was observed. All filler passes, however, was welded with a stable arc though a little more spatter was noted compared to 3-1 and 3-2. A slightly concave weld face was observed before the cover pass was laid. Asymmetrical trapezoid weaving with the configuration as shown in test 3-2 was implemented for this test as well. The weaving direction was however, configured the wrong way on the second trapezium, contributing to a more concave weld face. Even though the cover pass

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
Root	-	58.3	3	0	0	0	1.0	TRIW	2.0
1	1	88.3	3	0	3.83	0	3.3	TRIW	1.0
2	1	88.3	3	-2.96	7.49	18.7	3.7	ATRAW	1.0
2	2	88.3	3	3.58	7.49	0	3.4	ATRAW	1.0
3	1	88.3	3	-6.59	11.12	18.5	3.7	ATRAW	1.0
3	2	88.3	3	0.61	11.12	18.5	3.7	ATRAW	1.0
3	3	88.3	3	7.20	11.12	0	3.4	ATRAW	1.0
Cover	1	88.3	2	0	15.0	0	5.0	TRIW	1.0

Table 5.9: Butt joint type IV: 4-1 test parameters

was narrow and tall, it tied in all subsequent passes and created a convex surface. The feed speed appear to be adjusted a bit too low as too little filler metal was deposited.

4-2

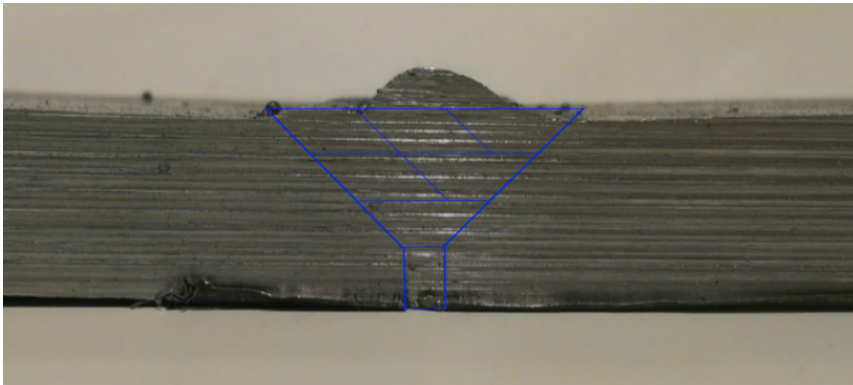
The following changes are made from test 4-1: The weaving frequency for the root pass is set back to 1 Hz and the weaving amplitude is decreased to 0.8 mm. The feed speed for all filler passes is increased to 5.4 m/min (90 mm/s), adjusting the deposition coefficient a_H to 1.11. The incorrect weaving direction for the second trapezium is corrected and a wider cover pass is implemented. All test parameters is given in table 5.10.

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
Root	-	58.3	3	0	0	0	0.8	TRIW	1.0
1	1	90	3	0	3.83	0	3.3	TRIW	1.0
2	1	90	3	-2.96	7.49	18.7	3.7	ATRAW	1.0
2	2	90	3	3.58	7.49	0	3.4	ATRAW	1.0
3	1	90	3	-6.59	11.12	18.5	3.7	ATRAW	1.0
3	2	90	3	0.61	11.12	18.5	3.7	ATRAW	1.0
3	3	90	3	7.20	11.12	0	3.4	ATRAW	1.0
Cover	1	90	2	0	15.0	0	6.5	TRIW	1.0

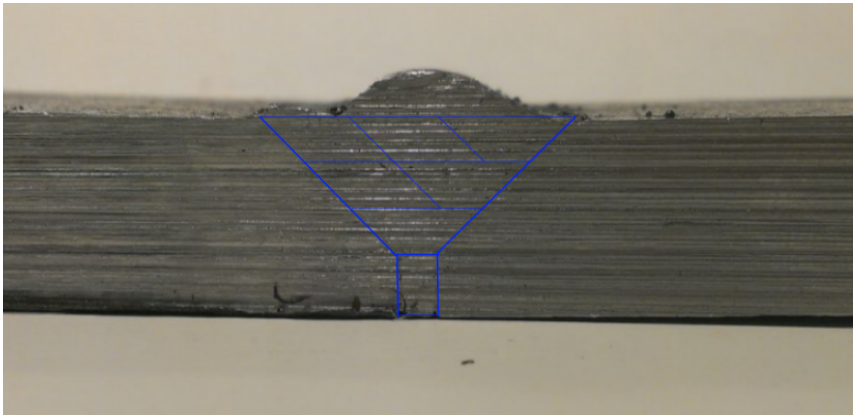
Table 5.10: Butt joint type IV: 4-2 test parameters

The cross section of the welded joint is shown in figure 5.20b. During welding of 4-2, a more stable root pass was observed. The root bead was a little high in the middle suggestion that a decreased weaving amplitude was undesirable. A virtually flat weld face was observed after the filler passes was laid. The wider cover pass produced a convex and more uniform bead compared to test 4-1. Still, a wider amplitude may advantageously be implemented as the bead still is fairly narrow. Also, the cover pass is a bit shifted to the right due to poor positional calibration of the base metal before the pass was laid. The increased feed speed and adjusted deposition efficiency coefficient appeared to have given the most accurate weld area so far. Figure 5.21 show a comparison between the two joints,

4-1 to the left and 4-2 to the right. Both joints show good fusion at the weld toe, though a little coarse edges from the asymmetrical trapezoid weaving.



(a) 4-1: Parameters from table 5.9



(b) 4-2: Parameters from table 5.10

Figure 5.20: Butt joint type IV: Test results

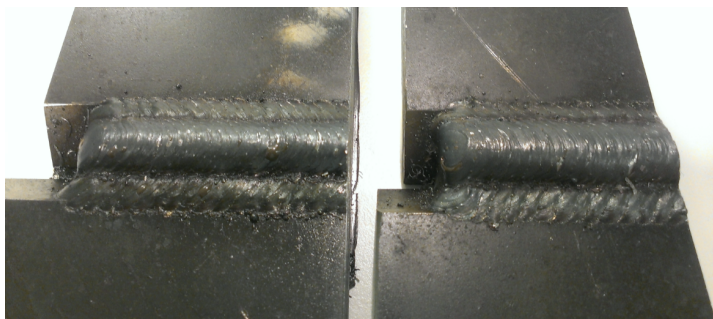


Figure 5.21: Butt joint type IV: 4-1 and 4-2 weld face

5.6 Final Tests

The final tests is constituted of three welding sequences with different blocks, respectively butt joint type II, III and IV. All testing up to now have contributed to adjust parameters to fit the mathematical model and generate good welds. On the final tests, the robot welding system is set in automatic mode, meaning that all layers and passes are completed in one sequence. This includes root pass and cover passes as well. This ultimately means, as discussed earlier, that the operator is unable to adjust the feed speed between passes due to the way the system is set up. The three final tests is denoted *2F*, *3F* and *4F*. Even though they are presented in an ascending sequence, test *3F* was the last joint welded. All three tests have been recorded using a video camera and are included in the digital appendix.

The following configurations are common for all final tests: Feed speed is set to 5.4 m/min (90 mm/s), with a deposition efficiency coefficient $a_H = 1.11$. After testing of different feed speeds, this value seemed to give the most accurate bead area relative to the schematic layout. A schematic grid will not be featured as before in the final result figures. The asymmetrical trapezoid weaving configuration from test 3-2 will be used as this produced the most flat weld face. Weaving frequency is held constant at 1.0 Hz, except for some of the cover passes. In regards to the root pass in test *2F* and *4F*, the increased feed speed is offset by an increase in welding speed to 0.28 m/min (4.6 mm/s).

2F

The final test layout for butt joint type II contains one root pass, four filler passes and a cover pass. The filler pass layout is the same as shown by the schematic layout from test 2-1, 2-2 and 2-3, in figure 5.11. The cover pass is set with a weaving amplitude of 7.5 mm. Preliminary testing of cover pass with approximately the same values produced a good convex weld face, see figure 5.7a. All other test parameters are given in table 5.11.

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
Root	-	90	4.6	0	0	0	1.0	TRIW	1.0
1	1	90	3	0	4.62	0	2.2	TRIW	1.0
2	1	90	3	0	7.27	0	3.7	TRIW	1.0
3	1	90	3	-2.59	11.12	23.1	2.9	ATRAW	1.0
3	2	90	3	3.40	11.12	0	2.5	ATRAW	1.0
Cover	1	90	2	0	12.0	0	7.5	TRIW	1.0

Table 5.11: Butt joint type II: Final test parameters

The resulting cross section of the welded joint is shown in figure 5.23a. During the weld process, an unstable arc was observed at the welding of all trapezium passes and at the cover pass. This was a surprising result as the same parameters provided an stable arc and good weld beads in previous tests. This is clearly apparent in the movie *WeldTest2F*. The arc is pulsating throughout the welding of the trapezium in second layer, third layer and at the cover pass. Welding of the parallelogram was more stable, though not perfect. As it turned out, the robot base had been configured for a zero root face. This means a 5

mm shorter stick-out length for the torch during the whole test process. The shorter arc length most probably led to the unstable arc. The discovery of this was not made before after the final test 4F. Nonetheless, good fusion with the base metal was obtained and a convex weld face can be seen in figure 5.23a.

3F

The final test layout for butt joint type III contain ten filler passes and two cover passes. Filler passes follow the same schematic layout as given in figure 5.14. Similar joints from tests 3-1 and 3-2 both had a concave weld face. Two cover passes are planned due to the width of the groove to ensure a convex weld bead. The first cover pass has an weaving amplitude of 15 mm. The widest successfully weaving test so far have been at 11 mm at 0.5 Hz, shown in the preliminary tests. A 10.5 mm weaving width at 0.75 Hz also produced adequate results in the preliminary tests. A 0.5 Hz frequency is set for the first cover pass. The second cover pass is set at 9 mm with frequency 0.75 Hz. All other test parameters are given in table 5.12.

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
1	1	90	3	0	3.83	0	3.3	TRIW	1.0
2	1	90	3	-2.96	7.49	18.7	3.7	ATRAW	1.0
2	2	90	3	3.58	7.49	0	3.4	ATRAW	1.0
3	1	90	3	-6.59	11.12	18.5	3.7	ATRAW	1.0
3	2	90	3	0.61	11.12	18.5	3.7	ATRAW	1.0
3	3	90	3	7.20	11.12	0	3.4	ATRAW	1.0
4	1	90	3	-10.22	14.75	18.5	3.7	ATRAW	1.0
4	2	90	3	-3.00	14.75	18.5	3.7	ATRAW	1.0
4	3	90	3	4.21	14.75	18.5	3.7	ATRAW	1.0
4	4	90	3	10.83	14.75	0	3.4	ATRAW	1.0
Cover	1	90	1.5	0	17.0	0	15.0	TRIW	0.5
Cover	2	90	1.5	0	18.0	0	9.0	TRIW	0.75

Table 5.12: Butt joint type III: Final test parameters

Figure 5.23b show the welded cross section of joint 3F, with its weld face displayed in figure 5.22b. The figures show good fusion and a uniform weld face. An stable arc and good fusion was observed throughout the welding process. Small amounts of spatter occurred. The two cover passes created a almost perfectly convex weld face with good tie-in of all subsequent weld passes. A movie *WeldTest3F* of this weld sequence is available in the digital appendix. Shorter periods of a pulsating arc can be seen in the first pass and the first cover pass. Nonetheless, this was the most stable weld sequence and the most representative of the three final tests as the robot base was set to the correct position.

4F

The final test layout of butt joint type IV contain one root pass, six filler passes and a cover pass. Root pass parameters are set equivalent to that of test 2F. Schematic layout for filler passes are identical to test 4-1 and 4-2, given in figure 5.19. A single cover pass is set at 0.75 Hz weaving with an amplitude of 10.5 mm. All other parameters are given in table 5.13

Layer	Pass	v_1	v_2	Δy	Δz	$\Delta\theta$	W	Type	Hz
Root	-	90	4.7	0	0	0	1.0	TRIW	1.0
1	1	90	3	0	3.83	0	3.3	TRIW	1.0
2	1	90	3	-2.96	7.49	18.7	3.7	ATRAW	1.0
2	2	90	3	3.58	7.49	0	3.4	ATRAW	1.0
3	1	90	3	-6.59	11.12	18.5	3.7	ATRAW	1.0
3	2	90	3	0.61	11.12	18.5	3.7	ATRAW	1.0
3	3	90	3	7.20	11.12	0	3.4	ATRAW	1.0
Cover	1	90	1.5	0	12.0	0	10.5	TRIW	0.75

Table 5.13: Butt joint type IV: Final test parameters

Figure 5.23c show the final cross section of the welded joint. Test 4F produced, like 2F and 3F, a uniform weld with good fusion. However, 4F experienced the same error with the robot base adjustment as test 2F. Hence, a 5 mm shorter arc length resulting in an unstable arc occurred during trapezium passes. This is also evident in the movie *WeldTest4F* where the first trapezium filler pass was highly pulsating. The trapezium weld in layer two also experienced a pulsing arc to some extent, while parallelogram welds were stable.

Figure 5.22 show a comparison of the three weld faces. 2F, figure 5.22a, clearly display the worst appearance of the three. The shortened stick-out length generated problems during the weld, producing spatter and unstable arc ignition. Test 4F had the same 5 mm error in the robot base. However the wider groove angle seemed to offset some of the effects as the arc was more stable and a more uniform weld face was obtained. Still, a good amount of spatter is seen at this specimen compared to 3F, shown in figure 5.22b.

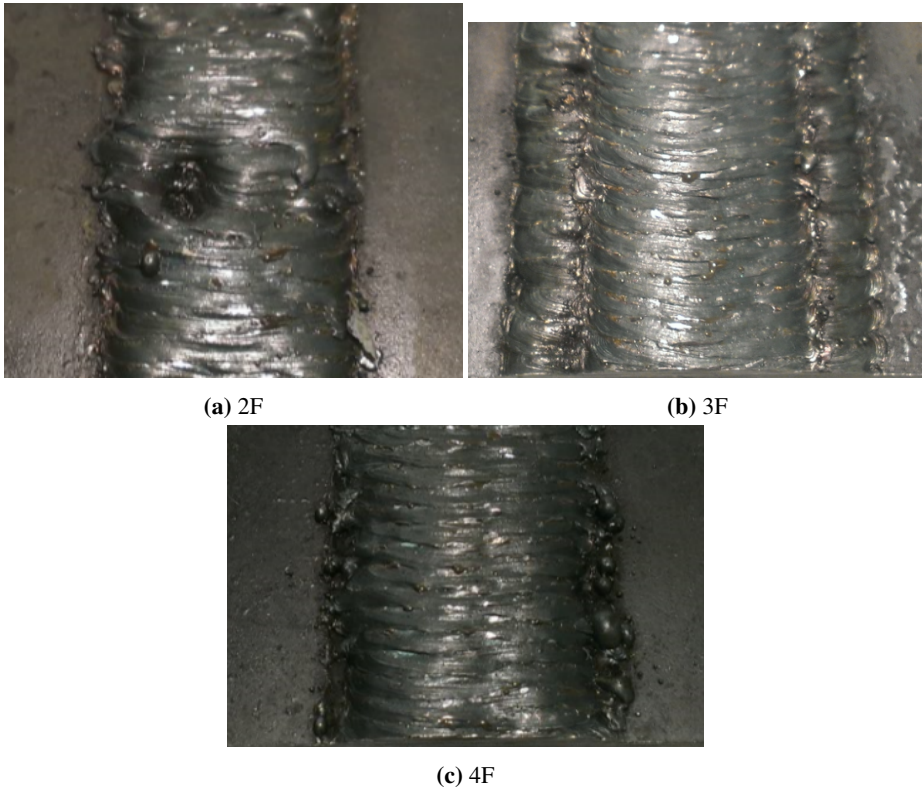
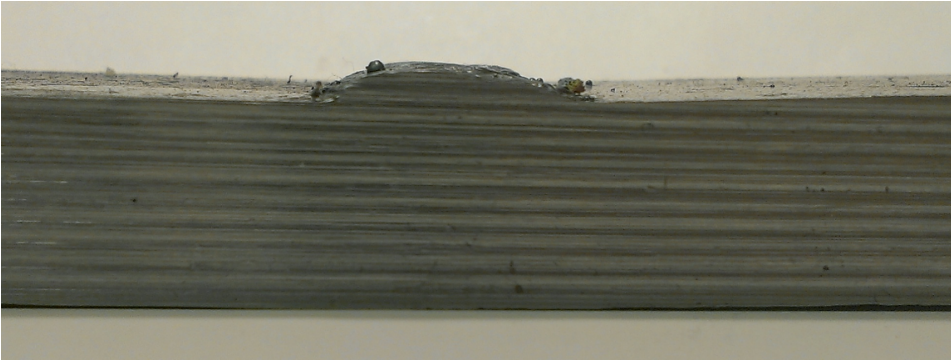


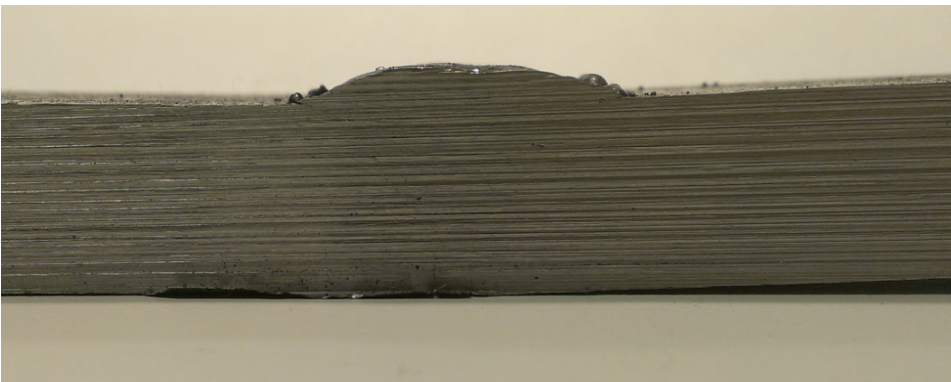
Figure 5.22: Final test weld faces



(a) 2F: Parameters from table 5.11



(b) 3F: Parameters from table 5.12



(c) 4F: Parameters from table 5.13

Figure 5.23: Final test results

5.7 Third Party Visual Inspection

A few of the welded joints was brought to Tromek AS, a mechanical workshop in Trondheim. They employ certified welders and have years of expertise and competence in the area. As a third party quality control, they were asked to evaluate some of the butt joints welded. The joints brought was 3-1, 3-2, 2F, 3F and 4F. Note that this was only a visual inspection and Tromek AS is not a certified quality control agency for welding. Nonetheless, the following comments were made about the welded joints.

First of all, they remarked that MAG welding is not well suited for welding thick butt joints. FCAW is the better choice for welding thick sections as it can deposit larger beads. Secondly, a full penetration butt joint is generally welded with backing that allows a root reinforcement. All tests in this laboratory implementation were conducted with flat backing plates, leaving no room for root reinforcement. A typical backing used is shown in figure 5.24.

In regards to the conducted welding, they noted that there was good fusion with side walls and in between layers for all exhibited butt joints. The beads were uniform and well made. In a comparison between 3-1 and 3-2, they concluded that 3-1 was the better weld. This was due to the coarse edges of 3-2, and a better overlap between the beads in the top layer for 3-1. They recommended to add an extra bead in between the third parallelogram and the trapezium (between bead 4,3 and 4,4 in figure 5.14). The beads should overlap more to create a flat uniform surface.

Lastly, they commented the weaving width of the cover passes for joint 2F, 3F and 4F. Normally, this large weaving is never allowed by different welding standards. This is generally restricted to two or three times the filler wire diameter, and to certain welding positions. The wide weaving was one of the main causes of shrinkage and distortion.



Figure 5.24: Backing plate

5.8 Remarks

A certain set of parameters have been found during the course of all testing, producing seemingly good results. For constant welding speed, the two parameters feed speed and deposition coefficient efficiency have been adjusted to match the schematic layout. Best matching results was found at $a_H = 1.11$. This is as mentioned earlier, a peculiar case. The electrode used is not an high efficiency electrode capable of metal recovery above 100%. This calls into question the validity of the equation 3.2. It could be the result of a set static and dynamic errors, inaccurate measurements, or simply the wrong use of the term deposition efficiency coefficient. a_H may in real life represent a number of different factors: Heat transfer effects, shrinkage, distortion, metal density change, metal recovery, etc. Relevant sources of errors are discussed below. The deposition efficiency coefficient can be measured in a timed test, or by measuring the change in mass for both the welded joint and filler wire drum before and after welding.

Another parameter of significance to the end result have been the weaving factor δ . A few tests was done with varying values for δ . The value $\delta = 2$ was chosen relatively early as it seemed to produce adequate results. Weaving widths generated ensured good fusion between the beads and side walls, covering a wide enough area.

Distortion was experienced at a few of the joints, especially butt joint type III and IV, in addition to specimens with cover passes. The heat input due to the number of passes and wide weaving in cover passes led to a certain amount of shrinkage in the joint. As mentioned in chapter 2, there are ways to avoid shrinkage. One way is to preheat the butt joints. This was not done do to lack of heating equipment. Another way is to preset the joints in a offset position to compensate the distortion. As the author had no perception of the amount of shrinkage that would occur, it was omitted. Also, the effects of distortion was to small to significantly affect the model.

All welding in the experiment have been done in a side to side sequence. Yang et al. (2014) argued that the layers are easier to keep flat in a side to center welding configuration, as a shorter inter pass time for cooling is needed. No specific approach was taken to test and measure the impact of different inter pass times during the experiments. Whether the effects of heat input and cooling times can significantly affect the result and thereby the model at these dimensions is not known to the author. However, a side to center welding sequence can provide better results and should be investigated further.

The original plan was to implement non-destructive testing (NDT) for weld quality assurance. Simple etch testing would be conducted to reveal fusion lines and determine the level of fusion. NDT is a vital tool for quality testing in the manufacturing industry, especially within the offshore sector. Even though visual inspection by qualified personnel is done, it is not a sufficient practice.

Source of errors

There are potentially a few error sources that should be noted. They are as follows:

- The precision of butt joints: All test pieces have been manufactured at mechanical workshop at IPK. Some of the test pieces was returned for more processing as the precision of the geometry was inadequate. The observed faults were related to the

root face which deviated along the piece. It was only apparent in two cases. Small deviations in root face affects the volume of the groove, thus affecting the conformity between the model and the welded joint. A higher root face (>5 mm) on the joints will give less weld area to fill. The deposited weld area may therefore appear larger than what is actually deposited, triggering the decision to decrease feed speed and increase a_H . All test cases seemed to indicate the same result. Thus, it is unlikely that the high deposition efficiency coefficient is due to poorly manufactured parts, as all parts would have to possess the same geometrical error.

- Another error may be related to the precision of the welding system, especially considering the wire feeder. The wire feeder is designed to provide a steady wire feed to the torch. As the developed procedure uses wire feed speed as one of the main parameters, it will have major influence on the welded joint. A static deviation from the desired feed speed may be possible. This was not tested prior to the welding. But if the wire feeder always provides a higher feed speed than what is programmed, more weld metal will be deposited. Again, this will lead to seemingly more deposited metal and to higher values for a_H .
- Poorly calibrated joint positions might induce experimental errors. Before each weld test, the robot was taken through the program without the welding system activated to ensure correct positioning. Visual inspection was the only means of calibration and is only so precise. It was done as thoroughly as possibly, however, a slight positional deviation is possible. This could lead to a small offset from the base origin, error in the root gap distance and also a weld path that start in the middle and is shifted to one of the sides. Estimating the size of this error is difficult. Certain steps were taken to minimize them. A solid steel plate measuring 3 mm was placed between the base metal pieces to ensure the correct root gap. Coordinates marking the origin were noted on the fixture for position reference.

Chapter 6

Conclusion and Future Work

This thesis has presented a method for solving an industrial challenge. An automated procedure for multiple-pass welding of V-grooved butt joints have been created and tested.

A brief discussion on multiple-pass welding of large offshore structures was done in section 2.4. A few examples from the industry have been given, including jacket manufacturing and ship building. Different studies and research on automated multiple-pass planning have been reviewed. This is outlined in section 2.5. Equations and concepts from this research have been used in the development of the automated procedure in this thesis.

The parameter controlled procedure creates a welding program based on the geometry of a V-grooved butt joint. The solution is described in chapter 3. Automated multiple-pass planning have been visualized in a programming software where weld beads are approximated as sets of parallelograms and trapeziums. Simulation of the welding process have been done in a simulation software and been translated into an executable robot program.

The procedure have been implemented in the robotics laboratory at IPK where several experiments have been conducted. Test procedures and equipment used is outlined in chapter 4. The robot welding system used have been configured for MAG welding. Four different V-groove butt joint geometries have been welded to test and adjust the accuracy of the automated procedure, given in chapter 5. The testing have included root pass, filler passes and cover passes. Conformity between the modelled- and the welded joints have been achieved. Good fusion and uniform welds without visual defects was obtained during the welding experiments.

Future Work

The system developed in this thesis is a functioning thick plate welding procedure using robotics. However, there is potential for improvement on different levels of the system. Here is a list of topics for further research:

- At the moment, the offline robot programming do not support ArcTech commands. Much time was spent on editing welding commands in the robot program on the robot controller. A better solution for generating robot programs with welding commands is needed. A possibility is to go through the software WorkVisual.
- In this thesis it is assumed that geometrical properties of the joint is set by the operator. A system that creates a model based on visual data, for example point clouds from a 3D camera, would be useful.
- Further testing of the system would be beneficial. Both in terms of parameters adjustments, various joint geometries and NDT for quality assurances.
- Inaccuracies in joint fit-up and positioning, distortion and variable geometries, raise the need for seam tracking. Real-time feedback for checking whether enough filler metal have been laid in a pass, and at the right position, is critical to maintain weld integrity. Most robotic welding systems is to date not able to do this, including the system developed here. Implementation of a well functioning seam tracking system would be a major development.

Bibliography

- ASME, 2010. ASME Boiler and pressure vessel code - Section IX: Welding and brazing qualifications.
URL <https://law.resource.org/pub/us/code/ibr/asme.bpvc.ix.2010.pdf>
- AWS, 2001. Welding Handbook: Welding Science & Technology, Ninth Edition. Vol. 1. American Welding Society.
- Bowditch, W. A. B. . K. E. B. . M. A., 2005. Welding Technology Fundamentals. The Goodheart-Willcox Company, Inc.
- Davies, A. C., 2003. The science and practice of welding, Tenth Edition. Vol. 2. CUP Archive.
- ESAB, 2000. Basic welding filler metal technology: Lesson II - common electric arc welding processes. Tech. rep., ESAB.
URL http://www.esabna.com/euweb/awtc/lesson2_1.htm
- ESAB, 2004. Product data sheet ok autrod 12.64. Tech. rep., ESAB, accessed 06.06.2016.
URL <http://www.eurocardis.com/hojastecnicas/hilossoldadura/OKAutrod12.64.PDF>
- Fronius International GmbH, 2010. Fronius International GmbH.
URL http://www.fronius.com/cps/rde/xchg/fronius_international/hs.xsl/1001_ENG_HTML.htm
- Huang, T.-Y., Fan, L.-W., Bao, Y.-N., Shi, H., Zhang, K., 2015. Multi-pass route planning for thick steel plate using laser welding with filler wire. In: Robotic Welding, Intelligence and Automation. Springer, pp. 551–560.
- Isao, M., et al., 1979. A program for setting co2 horizontal fillet welding parameters with given leg lengths and welding speed. Journal of the Japan Welding Society 48 (7), 94–98.

-
- ISO, 1972. ISO 2401:1972: Covered electrodes - determination of the efficiency, metal recovery and deposition coefficient.
- ISO, 1998. ISO 857-1:1998: Welding and allied processes - vocabulary - part 1: Metal welding processes.
- ISO, 2011. ISO 6947:2011: Welding and allied processes - welding positions.
- Jeffus, L., 1999. *Welding: Principles and Applications*, fourth edition Edition. Delmar Publishers.
- KUKA Roboter GmbH, 2012. KUKA.ArcTech Basic 1.0. KUKA Roboter GmbH.
- Kvaerner, 2015. Kvaerner. Accessed 30.05.2016.
URL <http://www.kvaerner.com/>
- Li, D. Y., Yi, X. D., Yang, D. Q., Zhang, G. J., 2015. Weld formation on overhead position for thick plate of high-strength low-alloy steel by double-sided arc welding. In: Chen, P. (Ed.), *Material Science and Environmental Engineering: Proceedings of the 3rd Annual 2015 International Conference on Material Science and Environmental Engineering (ICMSEE2015)*, Wuhan, Hubei, China. CRC Press, pp. 19–24.
- Li, K., Dai, S.-J., Sun, L.-X., Yue, H., 2001. Filling strategy of multi-pass welding. *Hanjie Xuebao(Transactions of the China Welding Institution)(China)(China)* 22 (2), 46–48.
- Moriyasu, M., Hiramoto, S., Ohmine, M., 1993. Development of a multi-pass welding program for arc welding robots and its application to heavy electrical section pieces. *Transactions of the Japan Welding Society* 24 (1), 24–29.
- Nakayama, H., 1994. Expert process planning system of CIM for shipbuilding. In: *Proceedings of International Conference on Computer Applications in Shipbuilding*. pp. 12.55–12.66.
- NTS, December 2000. NORSOK M-101: Structural steel fabrication.
- Ocean Farming, 2015. Havbasert fiskeoppdrett. Tech. rep., SalMar.
URL http://classic.vitaminw.no/kunde/Salmar09/FilVedlegg/Ocean-Farming_flyer_3sider.pdf
- Pan, Z., Polden, J., Larkin, N., Duin, S. V., Norrish, J., 2012. Recent progress on programming method for industrial robots. *Robotics and Computer-Integrated Manufacturing* 28, 87–94.
- Pires, J. N., Loureiro, A., Bölmsjö, G., 2006. *Welding Robots: Technology, System Issues and Applications*. Springer.
- Pires, J. N., Loureiro, A., Godinho, T., Ferreira, P., Fernando, B., Morgado, J., 2003. Welding robots. *Robotics & Automation Magazine, IEEE* 10 (2), 45–55.
- Soderstrom, E., Mendez, P., 2008. Metal transfer during gmaw with thin electrodes and ar-co² shielding gas mixtures. *WELDING JOURNAL-NEW YORK-* 87 (5), 124.

Welding Institute, 1975. Standard Data for Arc Welding. British Welding Research Association.

Yang, C., Ye, Z., Chen, Y., Zhong, J., Chen, S., 2014. Multi-pass path planning for thick plate by dsaw based on vision sensor. *Sensor Review* 34 (4), 416–423.

Zhang, H., Lu, H., Cai, C., Chen, S., 2011. Robot path planning in multi-pass weaving welding for thick plates. In: *Robotic Welding, Intelligence and Automation*. Springer, pp. 351–359.

Zhang, H., Zhang, G., Cai, C., YIN, Z., WU, L., 2009. Self-defining path layout strategy for thick plate arc welding robot. *Transactions of the China Welding Institution* 30 (3), 61–64.

Zou, Z., Li, Y., Yin, S., 2000. Welding and engineering application of high-strength low-alloy steel.

Appendices

A MATLAB Script: *sequencer.m*

```
1 % Multiple-pass path layout script: sequencer.m
2 %
3 % This script takes the geometry of a V-butt joint and suggests
4 % a welding program setup with multiple layers and passes.
5 % Output is a schematic view of the cross section containing layers
6 % and passes for filler passes. Root pass and cover pass is not included.
7 %
8 %   Geometry of given workpiece and input to script:
9 %       Rootgap           g
10 %       Depth of groove   totdepth
11 %       Groove angle      beta
12 %       Filler wire diameter D
13 %       Welding speed     V2
14 %       Wire feed speed   V1
15 %
16 %   External functions needed:
17 %   - crossectionarea.m
18 %   - delta_h_i.m
19 %   - delta_theta_para.m
20 %   - delta_y_para.m
21 %   - delta_y_trapes.m
22 %   - passlayerprop.m
23 %   - weaving_para.m
24 %   - weaving_trap.m
25 %
26 %   Bjoern Emil Evensen                               IPK, NTNU
27 %   Created:                                           10.03.16
28 %   Last modified:                                     15.05.16
29
30 clear
31 clc
32
33 %Use this if constant values are desired
34 %v1 = 1;           % Feed speed           [mm/s]
35 %v2 = 50;          % Welding speed        [mm/s]
36 %I = 200;          % Welding current      [V]           If needed
37 %D = 1.0;          % Wire diameter        [mm]
38 %g = 3;            % Rootgap              [mm]
39 %beta = 90;        % Bevelangle           [degrees]
40 delta = 2;         % Weaving parameter [mm]
41 a.H = 1.11;        % Deposition efficiency coefficient
42 % totdepth = 10; % Depth of bevel
43
44 %Create matrices for data input
45 paramtable = zeros(15,8);
46 S = zeros(10,10); % Empty matrix for crosssectional weld area all passes
47 hi = zeros(10,1); % Height of layer i
48 delta_hi = zeros(10,1); % Height of layer relative to reference path
49 mi = zeros(10,1); % Number of passes in layer i
50 i = 1;
51 mi(1) = 1;
52 weaving_width = zeros(10,10);
53 theta = zeros(10,10);
54
55
```

```

56 % Basic parameter input
57 str = 'Welcome to this multilayer butt weld sequencer. Lets establish some
      geometric properties.';
58 disp(str)
59 prompt = 'Input rootgap [mm]:           '; g = input(prompt);
60 prompt = 'Input groove angle [degrees]: '; beta = input(prompt);
61 prompt = 'Input wire diameter [mm]:     '; D = input(prompt);
62 prompt = 'Input depth of bevel [mm]:    '; totdepth = input(prompt);
63 %Weaving parameter delta as input?
64
65
66 % First pass
67 str = 'First pass parameters'; disp(str)
68 prompt = 'Welding speed [mm/s]:         '; V2 = input(prompt);
69 prompt = 'Feed speed [mm/s]:           '; V1 = input(prompt);
70
71 S(1,1) = crossectionarea(V1,V2,D,a_H);
72 delta_hi(1,1) = delta_h_i(g,beta,1,1,S);
73 hi(1,1) = h_i(g,beta,1,1,S,hi);
74 weaving_width(1,1) = weaving_trap(hi,beta,S,mi,delta,1);
75
76 popdelty = delta_y_trapes(1,mi,S,hi);
77 popdeltz = delta_hi(1,1);
78
79 %Fill in parameters from root pass to weld table
80 paramtable(1,1)=1; paramtable(1,2)=1;paramtable(1,3)=V1;
81 paramtable(1,4)=V2; paramtable(1,5)=popdelty;
82 paramtable(1,6)=popdeltz; paramtable(1,7)=theta(1,1);
83 paramtable(1,8)=weaving_width(1,1);
84
85 %Draw schematic layout of first pass with torch position
86 xtop = tand(beta/2)*totdepth;
87 x = [-(xtop+5+(g/2)) -(g/2) -(g/2+xtop) -(g/2+xtop+5)];
88 y = [0 0 totdepth totdepth]; fill(x,y,[.7 .7 .7])
89 hold on
90 x = [(xtop+5+(g/2)) (g/2) (g/2+xtop) (g/2+xtop+5)];
91 y = [0 0 totdepth totdepth]; fill(x,y,[.7 .7 .7])
92 title('Layer-Pass Plan')
93 x = [-(g/2) g/2 g/2+hi(1)*tand(beta/2) -(g/2+hi(1)*tand(beta/2))];
94 y = [0 0 hi(1) hi(1)]; fill(x,y,[.85 .85 .85])
95 string = '1,1';
96 text(-0.4*g, hi(1)/2, string);
97
98 axis equal tight
99
100 %Check if more passes are needed
101 if delta_hi(i) < totdepth
102     %Parameters for rest of welding
103     %str = 'Pass parameters'; disp(str)
104     %prompt = 'Welding speed [mm/s]:         '; V2 = input(prompt);
105     %prompt = 'Feed speed [mm/s]:           '; V1 = input(prompt);
106
107
108
109 %While-loop that breaks when the total height of layers have exceeded the
110 %top of the groove.
111 row = 2;

```

```

112 while delta_hi(i) < totdepth
113     %Use these if individual parameters are desired.
114     %str = 'Pass parameters'; disp(str)
115     %prompt = 'Welding speed [mm/s]:           '; V2 = input(prompt);
116     %prompt = 'Feed speed [mm/s]:             '; V1 = input(prompt);
117
118     i = i+1;      %Next layer
119     fprintf('Starting layer %i.\n',i);
120
121     %Calculate number of passes proposition in layer i.
122     %Suggestion of minimum nb. of passes
123     while 1
124         suggestion = passlayerprop(S,mi,i,hi,V1,V2,D,delta,g,beta,a.H);
125         if suggestion == 1000
126             disp('Weaving width exceeded. No solution within given range.')
```

```

127             suggestion = 0;
128             break;
129         elseif suggestion == 100
130             disp('Layer thickness exceeded.')
```

```

131             suggestion = 0;
132             break;
133         else
134             break;
135         end
136     end
137
138     if suggestion == 0;
139         disp('Not able to recommend number of passes. Consider using other
140             parameters.')
```

```

141         else
142             fprintf('Minimum number of passes recommended is %i.\n',suggestion)
143         end
144         prompt = 'How many passes would you like to make in this layer?     ';
145         mi(i) = input(prompt); %Recommended number?
146
147         if mi(i) > 1
148             for j = 1:mi(i)
149                 S(i,j) = crossectionarea(V1,V2,D,a.H);
150             end
151
152             for j = 1:mi(i)
153                 if j == mi(i)
154                     %trapezium action
155                     hi(i) = h_i(g,beta,i,mi,S,hi);
156                     delta_hi(i) = delta_h_i(g,beta,i,mi,S);
157                     popdeltz = delta_hi(i);
158                     popdelty = delta_y_trapes(i,mi,S,hi);
159                     weaving_width(i,j) = weaving_trap(hi,beta,S,mi,delta,i);
160
161                     %Fill in parameters for trapez weld to weld table
162                     paramtable(row,1)=i; paramtable(row,2)=j; paramtable(row,3)=V1;
163                     paramtable(row,4)=V2; paramtable(row,5)=popdelty;
164                     paramtable(row,6)=popdeltz;
165                     paramtable(row,7)=0; paramtable(row,8)=weaving_width(i,j);
166
167                     %Draw trapezium
168                     sum = 0;

```

```

168         for k=1:mi(i)
169             sum = sum + S(i,k);
170         end
171
172         x = [ -((g/2)+delta_hi(i-1)*tand(beta/2))+(j-1)*sum/(hi(i)*mi(i)
173         )) ...
174             (g/2)+delta_hi(i-1)*tand(beta/2) ...
175             (g/2)+delta_hi(i)*tand(beta/2) ...
176             -((g/2)+delta_hi(i)*tand(beta/2))+(j-1)*sum/(hi(i)*mi(i))
177         ];
178
179         y = [ delta_hi(i-1) delta_hi(i-1) delta_hi(i) delta_hi(i) ];
180         fill(x,y,[.85 .85 .85])
181         string = sprintf('%i,%i',i,j);
182         mid = x(1)+0.25*x(2);
183         text(mid,delta_hi(i)-(hi(i)/2),string);
184         row = row +1;
185
186     else
187         %Shape is parallelogram
188         hi(i) = h_i(g,beta,i,mi,S,hi);
189         delta_hi(i) = delta_h_i(g,beta,i,mi,S);
190
191         popdelty = delta_y_para(S,beta,mi,hi,delta_hi,i,j,g);
192         popdeltz = delta_hi(i);
193         weaving_width(i,j) = weaving_para(S,beta,mi,hi,delta,i);
194         theta(i,j) = delta_theta_para(hi,S,beta,mi,i);
195
196         paramtable(row,1)=i; paramtable(row,2)=j;
197         paramtable(row,3)=V1;
198         paramtable(row,4)=V2; paramtable(row,5)=popdelty;
199         paramtable(row,6)=popdeltz;
200         paramtable(row,7)=theta(i,j);
201         paramtable(row,8)=weaving_width(i,j);
202
203     %Draw parallelogram
204     sum = 0;
205     for k=1:mi(i)
206         sum = sum + S(i,k);
207     end
208
209     x = [ -((g/2)+delta_hi(i-1)*tand(beta/2))+(j-1)*sum/(mi(i)*hi(i)
210     )) ...
211         -((g/2)+delta_hi(i-1)*tand(beta/2))+j*sum/(mi(i)*hi(i))
212     ...
213         -((g/2)+delta_hi(i)*tand(beta/2))+(sum*(j)/(mi(i)*hi(i)))
214     ...
215         -((g/2)+delta_hi(i)*tand(beta/2))+(j-1)*sum/(mi(i)*hi(i))
216     ];
217
218     y = [ delta_hi(i-1) delta_hi(i-1) delta_hi(i) delta_hi(i) ];
219     fill(x,y,[.85 .85 .85])
220     string = sprintf('%i,%i',i,j);
221     mid = x(1)+abs(0.2*x(2));
222     text(mid,delta_hi(i)-(hi(i)/2),string);
223
224     row = row + 1;
225 end

```

```

219     end
220 else
221     %Trapezium calculations.
222     %in this case (mi=1), delta y is zero. Pose angle is always zero.
223     S(i) = crossectionarea(V1,V2,D,a-H);
224     hi(i) = h_i(g,beta,i,mi,S,hi);
225     delta_hi(i) = delta_h_i(g,beta,i,mi,S);
226     popdeltz = delta_hi(i);
227     weaving_width(i,1) = weaving_trap(hi,beta,S,mi,delta,i);
228
229     %Fill in parameters for trapezium weld to weld table
230     paramtable(row,1)=i; paramtable(row,2)=1; paramtable(row,3)=V1;
231     paramtable(row,4)=V2; paramtable(row,5)=0; paramtable(row,6)=popdeltz
;
232     paramtable(row,7)=0; paramtable(row,8)=weaving_width(i,1);
233
234     x = [ -((g/2)+delta_hi(i-1)*tand(beta/2)) (g/2)+delta_hi(i-1)*tand(
beta/2) ...
235         (g/2)+delta_hi(i)*tand(beta/2) -((g/2)+delta_hi(i)*tand(beta
/2)) ];
236     y = [ delta_hi(i-1) delta_hi(i-1) delta_hi(i) delta_hi(i) ];
237     fill(x,y,[.85 .85 .85])
238     string = sprintf('%i,1',i);
239     text(0,delta_hi(i)-(hi(i)/2),string);
240
241     row = row + 1;
242 end
243
244 end
245
246 end
247
248 %Draw arrows to show the position and pose of weld torch
249 for rows = 1:size(paramtable,1)
250     if paramtable(rows,1) == 0
251         break
252     end
253
254     p1 = [ paramtable(rows,5)+3*tand(paramtable(rows,7)) paramtable(rows,6)
+3];
255     p2 = [ paramtable(rows,5) paramtable(rows,6) ];
256     dp = p2-p1;
257     quiver(p1(1),p1(2),dp(1),dp(2),0,'MaxHeadSize',0.5,'color','r')
258
259 end

```

B Python Script: *jointmodel.py*

```
1 # Script jointmodel.py
2 #
3 # Script for generating a 3D model of a V-grooved butt joint
4 # based on parameters set by the user.
5 # Script creates a graphical user interface in 3DAutomate
6 # and generates model at the command 'Regenerate model'.
7 #
8 # Bjoern Emil Evensen                IPK, NINU
9 # Created                            22.01.2016
10 # Last modified                      01.04.2016
11
12 from vcScript import *
13 from random import *
14 from vcFeature import *
15 from math import *
16 import vcVector, vcMatrix
17 import time
18
19 comp = getComponent()
20 featur = comp.findFeature('Featurz')
21
22 geo_cont = featur.Geometry
23 geo_cont.clear()
24
25 if not featur:
26     featur = comp.RootFeature.createFeature(Vc_GEOMETRY, 'Featurz')
27 if featur.Geometry.GeometrySetCount == 0:
28     pcs = featur.Geometry.createGeometrySet(Vc_TRIANGLESET)
29 else:
30     pcs = featur.Geometry.getGeometrySet(0)
31
32 #
33 # Get/Set component parameters:
34 #
35 rootgapProp = comp.createProperty(Vc_INTEGER, 'Root gap'); rootgapProp.
    Value = 5
36 bottomlengthProp = comp.createProperty(Vc_INTEGER, 'Length bottom plate');
    bottomlengthProp.Value = 150
37 bottomwidthProp = comp.createProperty(Vc_INTEGER, 'Width bottom plate');
    bottomwidthProp.Value = 75
38 bevelangle1Prop = comp.createProperty(Vc_INTEGER, 'Bevel angle side 1');
    bevelangle1Prop.Value = 30
39 bevelangle2Prop = comp.createProperty(Vc_INTEGER, 'Bevel angle side 2');
    bevelangle2Prop.Value = 30
40 platethicknessProp = comp.createProperty(Vc_INTEGER, 'Plate thickness');
    platethicknessProp.Value = 15
41 rootheightProp = comp.createProperty(Vc_INTEGER, 'Root height');
    rootheightProp.Value = 5
42
43
44 def createbottomplate(rootgap, bottomwidth, bottomlength):
45     #Right side square
46     v1, v2, v3, v4 = vcVector.new(), vcVector.new(), vcVector.new(),
    vcVector.new()
47     v1.X, v1.Y, v1.Z = rootgap/2, 0, 0
```

```

48 v2.X, v2.Y, v2.Z = rootgap/2, bottomlength, 0
49 v3.X, v3.Y, v3.Z = rootgap/2+bottomwidth, 0, 0
50 v4.X, v4.Y, v4.Z = rootgap/2+bottomwidth, bottomlength, 0
51 p1 = pcs.addPoint(v1); p2 = pcs.addPoint(v2); p3 = pcs.addPoint(v3); p4
    = pcs.addPoint(v4)
52 pcs.addTriangle(p1,p2,p3); pcs.addTriangle(p3,p2,p4)
53
54 #Left side square
55 v1s, v2s, v3s, v4s = vcVector.new(), vcVector.new(), vcVector.new(),
    vcVector.new()
56 v1s.X, v1s.Y, v1s.Z = -1*((rootgap/2)+bottomwidth), 0, 0
57 v2s.X, v2s.Y, v2s.Z = -1*(rootgap/2), bottomlength, 0
58 v3s.X, v3s.Y, v3s.Z = -1*(rootgap/2), 0, 0
59 v4s.X, v4s.Y, v4s.Z = -1*((rootgap/2)+bottomwidth), bottomlength, 0
60 p1s = pcs.addPoint(v1s); p2s = pcs.addPoint(v2s); p3s = pcs.addPoint(v3s
    ); p4s = pcs.addPoint(v4s)
61 pcs.addTriangle(p1s,p4s,p2s); pcs.addTriangle(p3s,p1s,p2s)
62
63 def createoutsidewalls(bottomwidth, bottomlength, platethickness):
64     #Right side square
65     v1, v2, v3, v4 = vcVector.new(), vcVector.new(), vcVector.new(),
        vcVector.new()
66     v1.X, v1.Y, v1.Z = rootgap/2+bottomwidth, 0, 0
67     v2.X, v2.Y, v2.Z = rootgap/2+bottomwidth, 0, platethickness
68     v3.X, v3.Y, v3.Z = rootgap/2+bottomwidth, bottomlength, 0
69     v4.X, v4.Y, v4.Z = rootgap/2+bottomwidth, bottomlength, platethickness
70     p1 = pcs.addPoint(v1); p2 = pcs.addPoint(v2); p3 = pcs.addPoint(v3); p4
        = pcs.addPoint(v4)
71     pcs.addTriangle(p2,p1,p3); pcs.addTriangle(p2,p3,p4)
72
73     #Left side square
74     v1s, v2s, v3s, v4s = vcVector.new(), vcVector.new(), vcVector.new(),
        vcVector.new()
75     v1s.X, v1s.Y, v1s.Z = -1*(rootgap/2+bottomwidth), 0, 0
76     v2s.X, v2s.Y, v2s.Z = -1*(rootgap/2+bottomwidth), 0, platethickness
77     v3s.X, v3s.Y, v3s.Z = -1*(rootgap/2+bottomwidth), bottomlength, 0
78     v4s.X, v4s.Y, v4s.Z = -1*(rootgap/2+bottomwidth), bottomlength,
        platethickness
79     p1 = pcs.addPoint(v1s); p2 = pcs.addPoint(v2s); p3 = pcs.addPoint(v3s);
        p4 = pcs.addPoint(v4s)
80     pcs.addTriangle(p1,p2,p3); pcs.addTriangle(p3,p2,p4)
81
82 def createbevel(rootgap, rootheight, bottomlength, bottomwidth, platethickness
    , bevelangle1, bevelangle2):
83     alpha=platethickness-rootheight
84
85     ###RIGHT SIDE###
86     #Front right triangle + frontsurface
87     v1fr, v2fr, v3fr, v4fr, v5fr = vcVector.new(), vcVector.new(), vcVector.
        new(), vcVector.new(), vcVector.new()
88     v1fr.X, v1fr.Y, v1fr.Z = rootgap/2, 0, rootheight
89     v2fr.X, v2fr.Y, v2fr.Z = rootgap/2+(alpha*tan(radians(bevelangle1))),
        0, rootheight
90     v3fr.X, v3fr.Y, v3fr.Z = rootgap/2+(alpha*tan(radians(bevelangle1))), 0,
        platethickness
91     v4fr.X, v4fr.Y, v4fr.Z = rootgap/2+bottomwidth, 0, rootheight
92     v5fr.X, v5fr.Y, v5fr.Z = rootgap/2+bottomwidth, 0, platethickness

```

```

93  p1fr = pcs.addPoint(v1fr); p2fr = pcs.addPoint(v2fr); p3fr = pcs.
    addPoint(v3fr)
94  p4fr = pcs.addPoint(v4fr); p5fr = pcs.addPoint(v5fr)
95  pcs.addTriangle(p1fr,p2fr,p3fr)
96  pcs.addTriangle(p4fr,p5fr,p3fr)
97  pcs.addTriangle(p2fr,p4fr,p3fr)
98
99  #Back right triangle + back surface
100 v1br, v2br, v3br, v4br, v5br = vcVector.new(), vcVector.new(), vcVector.
    new(), vcVector.new(), vcVector.new()
101 v1br.X, v1br.Y, v1br.Z = rootgap/2, bottomlength, rootheight
102 v2br.X, v2br.Y, v2br.Z = rootgap/2+(alpha*tan(radians(bevelangle2))),
    bottomlength, rootheight
103 v3br.X, v3br.Y, v3br.Z = rootgap/2+(alpha*tan(radians(bevelangle2))),
    bottomlength, platethickness
104 v4br.X, v4br.Y, v4br.Z = rootgap/2+bottomwidth, bottomlength, rootheight
105 v5br.X, v5br.Y, v5br.Z = rootgap/2+bottomwidth, bottomlength,
    platethickness
106 p1br = pcs.addPoint(v1br); p2br = pcs.addPoint(v2br); p3br = pcs.
    addPoint(v3br)
107 p4br = pcs.addPoint(v4br); p5br = pcs.addPoint(v5br)
108 pcs.addTriangle(p2br,p1br,p3br)
109 pcs.addTriangle(p5br,p4br,p3br)
110 pcs.addTriangle(p4br,p2br,p3br)
111
112 #Right side bevel surface
113 pcs.addTriangle(p1br,p1fr,p3br)
114 pcs.addTriangle(p1fr,p3fr,p3br)
115
116 #Top right surface
117 pcs.addTriangle(p3br,p3fr,p5br)
118 pcs.addTriangle(p3fr,p5fr,p5br)
119
120
121 ###LEFT SIDE###
122 #Front left triangle
123 v1fl, v2fl, v3fl, v4fl, v5fl = vcVector.new(), vcVector.new(), vcVector.
    new(), vcVector.new(), vcVector.new()
124 v1fl.X, v1fl.Y, v1fl.Z = -1*(rootgap/2), 0, rootheight
125 v2fl.X, v2fl.Y, v2fl.Z = -1*(rootgap/2+(alpha*tan(radians(bevelangle1)))
    ), 0, rootheight
126 v3fl.X, v3fl.Y, v3fl.Z = -1*(rootgap/2+(alpha*tan(radians(bevelangle1)))
    ), 0, platethickness
127 v4fl.X, v4fl.Y, v4fl.Z = -1*(rootgap/2+bottomwidth), 0, rootheight
128 v5fl.X, v5fl.Y, v5fl.Z = -1*(rootgap/2+bottomwidth), 0, platethickness
129 p1fl = pcs.addPoint(v1fl); p2fl = pcs.addPoint(v2fl); p3fl = pcs.
    addPoint(v3fl)
130 p4fl = pcs.addPoint(v4fl); p5fl = pcs.addPoint(v5fl)
131 pcs.addTriangle(p2fl,p1fl,p3fl)
132 pcs.addTriangle(p4fl,p3fl,p5fl)
133 pcs.addTriangle(p4fl,p2fl,p3fl)
134
135 #Back left triangle
136 v1bl, v2bl, v3bl, v4bl, v5bl = vcVector.new(), vcVector.new(), vcVector.
    new(), vcVector.new(), vcVector.new()
137 v1bl.X, v1bl.Y, v1bl.Z = -1*(rootgap/2), bottomlength, rootheight

```

```

138 v2bl.X, v2bl.Y, v2bl.Z = -1*(rootgap/2+(alpha*tan(radians(bevelangle2))))
    ), bottomlength, rootheight
139 v3bl.X, v3bl.Y, v3bl.Z = -1*(rootgap/2+(alpha*tan(radians(bevelangle2))))
    ), bottomlength, platethickness
140 v4bl.X, v4bl.Y, v4bl.Z = -1*(rootgap/2+bottomwidth), bottomlength,
    rootheight
141 v5bl.X, v5bl.Y, v5bl.Z = -1*(rootgap/2+bottomwidth), bottomlength,
    platethickness
142 p1bl = pcs.addPoint(v1bl); p2bl = pcs.addPoint(v2bl); p3bl = pcs.
    addPoint(v3bl)
143 p4bl = pcs.addPoint(v4bl); p5bl = pcs.addPoint(v5bl)
144 pcs.addTriangle(p1bl,p2bl,p3bl)
145 pcs.addTriangle(p4bl,p5bl,p3bl)
146 pcs.addTriangle(p2bl,p4bl,p3bl)
147
148 #Left side bevel surface
149 pcs.addTriangle(p1fl,p1bl,p3bl)
150 pcs.addTriangle(p3fl,p1fl,p3bl)
151
152 #Top left surface
153 pcs.addTriangle(p3fl,p3bl,p5bl)
154 pcs.addTriangle(p5fl,p3fl,p5bl)
155
156 def createinsidewalls(bottomwidth,bottomlength,rootheight,rootgap):
157     v1, v2, v3, v4, v5, v6, v7, v8 = vcVector.new(), vcVector.new(),
        vcVector.new(), vcVector.new(), vcVector.new(),
        vcVector.new(), vcVector.new()
158     v1.X, v1.Y, v1.Z = rootgap/2, 0, 0
159     v2.X, v2.Y, v2.Z = rootgap/2, bottomlength, 0
160     v3.X, v3.Y, v3.Z = rootgap/2, 0, rootheight
161     v4.X, v4.Y, v4.Z = rootgap/2, bottomlength, rootheight
162     v5.X, v5.Y, v5.Z = rootgap/2+bottomwidth, 0, 0
163     v6.X, v6.Y, v6.Z = rootgap/2+bottomwidth, 0, rootheight
164     v7.X, v7.Y, v7.Z = rootgap/2+bottomwidth, bottomlength, 0
165     v8.X, v8.Y, v8.Z = rootgap/2+bottomwidth, bottomlength, rootheight
166     p1 = pcs.addPoint(v1); p2 = pcs.addPoint(v2); p3 = pcs.addPoint(v3); p4
        = pcs.addPoint(v4)
167     p5 = pcs.addPoint(v5); p6 = pcs.addPoint(v6); p7 = pcs.addPoint(v7); p8
        = pcs.addPoint(v8)
168
169     pcs.addTriangle(p2,p1,p3); pcs.addTriangle(p2,p3,p4)
170     pcs.addTriangle(p1,p5,p6); pcs.addTriangle(p1,p6,p3)
171     pcs.addTriangle(p7,p4,p8); pcs.addTriangle(p7,p2,p4)
172
173     v1s, v2s, v3s, v4s, v5s, v6s, v7s, v8s = vcVector.new(), vcVector.new(),
        vcVector.new(), vcVector.new(), vcVector.new(),
        vcVector.new(), vcVector.new()
174     v1s.X, v1s.Y, v1s.Z = -1*(rootgap/2), 0, 0
175     v2s.X, v2s.Y, v2s.Z = -1*(rootgap/2), bottomlength, 0
176     v3s.X, v3s.Y, v3s.Z = -1*(rootgap/2), 0, rootheight
177     v4s.X, v4s.Y, v4s.Z = -1*(rootgap/2), bottomlength, rootheight
178     v5s.X, v5s.Y, v5s.Z = -1*(rootgap/2+bottomwidth), 0, 0
179     v6s.X, v6s.Y, v6s.Z = -1*(rootgap/2+bottomwidth), 0, rootheight
180     v7s.X, v7s.Y, v7s.Z = -1*(rootgap/2+bottomwidth), bottomlength, 0
181     v8s.X, v8s.Y, v8s.Z = -1*(rootgap/2+bottomwidth), bottomlength,
        rootheight

```

```

182 p1 = pcs.addPoint(v1s); p2 = pcs.addPoint(v2s); p3 = pcs.addPoint(v3s);
    p4 = pcs.addPoint(v4s)
183 p5 = pcs.addPoint(v5s); p6 = pcs.addPoint(v6s); p7 = pcs.addPoint(v7s);
    p8 = pcs.addPoint(v8s)
184 pcs.addTriangle(p1,p2,p3); pcs.addTriangle(p3,p2,p4)
185 pcs.addTriangle(p5,p1,p6); pcs.addTriangle(p6,p1,p3)
186 pcs.addTriangle(p4,p7,p8); pcs.addTriangle(p2,p7,p4)
187
188 #Parameters
189 rootgap = rootgapProp.Value
190 bottomwidth = bottomwidthProp.Value
191 bottomlength = bottomlengthProp.Value
192 platethickness = platethicknessProp.Value
193 rootheight = rootheightProp.Value
194 bevelangle1 = bevelangle1Prop.Value
195 bevelangle2 = bevelangle2Prop.Value
196
197 #comp.deleteProperty('Regenerate model') #For use when deleting a
    property in the parameter tab.
198
199 def regenerateModel():
200     pcs.update()
201     featur.rebuild()
202     comp.rebuild()
203     getApplication().render()
204
205 def buildModel(arg):
206     for triangles in range(pcs.TriangleCount):
207         pcs.deleteTriangle(triangles)
208     pcs.update()
209     updateVariables()
210     createbottomplate(rootgap, bottomwidth, bottomlength)
211     createoutsidewalls(bottomwidth,bottomlength,platethickness)
212     createinsidewalls(bottomwidth,bottomlength,rootheight,rootgap)
213     createbevel(rootgap,rootheight,bottomlength,bottomwidth,platethickness,
        bevelangle1,bevelangle2)
214     regenerateModel()
215     print "Triangles; ", pcs.TriangleCount
216
217 def updateVariables():
218     global rootgap, bottomwidth, bottomlength, platethickness, rootheight,
        bevelangle1, bevelangle2
219     rootgap = rootgapProp.Value
220     bottomwidth = bottomwidthProp.Value
221     bottomlength = bottomlengthProp.Value
222     platethickness = platethicknessProp.Value
223     rootheight = rootheightProp.Value
224     bevelangle1 = bevelangle1Prop.Value
225     bevelangle2 = bevelangle2Prop.Value
226
227
228 arg = 0
229 buildModel(arg)
230
231 global regenModelProp
232 regenModelProp = comp.createProperty(VC.BUTTON, 'Regenerate model')
233 regenModelProp.OnChanged = buildModel

```

C Python Script: *sequencer.py*

```
1 # Script sequencer.py
2 #
3 # Script the calculates filler passes and respective
4 # values such as torch position , work angles , directional
5 # offsets , etc. Same base equation as sequencer.m from MATLAB
6 # Limitations :
7 # - No functionality for user input in script such as MATLAB.
8 # - Number of layer and passes are hardcoded.
9 # - No calculation for pass number suggestion
10 # - Welding parameters are hardcoded.
11 #
12 # Output is written to paramtable.csv
13 # - Change directory path before use in sequencer function!
14 #
15 # Bjoern Emil Evensen                IPK, NTNU
16 # Created                            10.04.2016
17 # Last modified                      05.05.2016
18
19 from vcScript import *
20 from random import *
21 from vcFeature import *
22 from math import *
23 import vcVector , vcMatrix
24 import time
25 import sys
26 import csv
27
28 comp = getComponent()
29
30 # Get joint model parameters :
31 rootgapProp = comp.getProperty('Root gap')                #rootgapProp.
32     Value = 5
33 bevelangle1Prop = comp.getProperty('Bevel angle side 1') #
34     bevelangle1Prop.Value = 30
35 bevelangle2Prop = comp.getProperty('Bevel angle side 2') #
36     bevelangle2Prop.Value = 30
37 platethicknessProp = comp.getProperty('Plate thickness') #
38     platethicknessProp.Value = 20
39 rootheightProp = comp.getProperty('Root height')         #rootheightProp
40     .Value = 5
41
42 #Define functions
43 def h_i(g, beta , i , mi , S , hi) :
44     h_iprev=0
45     if i > 0:
46         for k in range(0, i) :
47             h_iprev=h_iprev+hi[k]
48         return delta_h_i(g, beta , i , mi , S)–h_iprev
49
50 def delta_h_i(g, beta , i , mi , S) :
51     sum_S=0
52     for k in range(0, i+1) :
53         for l in range(max(mi)) :
54             sum_S=sum_S+S[k][l]
55     return (sqrt(pow(g, 2)+4*tan(0.5*beta)*sum_S)–g)/(2*tan(0.5*beta))
```

```

51
52 def crossectionarea (feedspeed , weldingspeed , wire_D , a_H ):
53     return (a_H*pi*feedspeed*pow(wire_D ,2))/(4*weldingspeed)
54
55 def weaving_trap (hi , beta , S , mi , delta , i ):
56     sum=0
57     for k in range (0 , i+1 ):
58         sum=sum+S[i][k]
59     return 0.5*hi[i]*tan (0.5*beta)+sum/(2*mi[i]*hi[i])-delta
60
61 def weaving_para (S , beta , mi , hi , delta , i ):
62     sum=0
63     for k in range (0 , max(mi)):
64         sum=sum+S[i][k]
65     return 0.5*sqrt(pow(hi[i]*tan (0.5*beta)+sum/(mi[i]*hi[i]),2) + pow(hi[i]
66         ],2))-delta;
67
68 def delta_y_trapes (i , mi , S , hi ):
69     sum=0
70     for k in range (0 , i+1 ):
71         sum=sum+S[i][k]
72     return ((mi[i]-1)*sum)/(2*mi[i]*hi[i])
73
74 def delta_y_para (S , beta , mi , hi , delta_hi , i , j , g ):
75     sum=0
76     for k in range (0 , i+1 ):
77         sum=sum+S[i][k]
78     var = hi[i]*tan (0.5*beta)+sum/(mi[i]*hi[i])
79     y=(0.5*(pow(var ,2)+pow(hi[i],2)))/(hi[i]*tan (0.5*beta)+sum/(mi[i]*hi[i])
80     )
81     sum_S=0
82     for k in range (0 , j-1 ):
83         sum_S=sum_S+S[i][k]
84     return y - 0.5*g - delta_hi[i]*tan (0.5*beta)+sum_S/hi[i]
85
86 def delta_theta_para (hi , S , beta , mi , i ):
87     sum=0
88     for k in range (0 , i+1 ):
89         sum=sum+S[i][k]
90     return atan (hi[i]/(hi[i]*tan (0.5*beta)+sum/(mi[i]*hi[i])))
91
92 ###Main routine
93 def sequencer (beta_deg , g , totthickness , rootheight ):
94     #Create matrices for value input
95     hi=[ 0 for i in range (10) ]
96     mi=[ 0 for i in range (10) ]
97     V1=[ 0 for i in range (10) ]
98     V2=[ 0 for i in range (10) ]
99     #sum.S=[ 0 for i in range (10) ] #Save time by calculating this in
100     sequencer()?
101     delta_hi=[ 0 for i in range (10) ]
102     weaving_width=[[ 0 for i in range (10) ] for j in range (10) ]
103     theta=[[ 0 for i in range (10) ] for j in range (10) ]
104     S=[ [ 0 for i in range (10) ] for j in range (10) ]
105     paramtable = [ [ 0 for i in range (8) ] for j in range (20) ]

```

```

105 #Define/set variables/parameters
106 totdepth=totthickness-rootheight
107 D = 1 #Wire diameter
108 delta = 2 #Weaving parameter
109 i = 0 #Iteration parameter
110 a_H = 1.11 #Deposition efficiency coefficient
111
112 #First pass
113 mi[0] = 1
114 V1[0] = 100
115 V2[0] = 3
116 #Set parameters for all passes and layers (pass numbers, welding speed
    and feed speed)
117 mi[1],mi[2],mi[3],mi[4],mi[5],mi[6],mi[7],mi[8],mi[9]=2,2,4,5,0,0,0,0,0
    #Hardcoded pass numbers
118 V1[1],V1[2],V1[3],V1[4],V1[5],V1[6],V1[7],V1[8],V1
    [9]=100,100,100,100,100,100,100,0,0 #Feed speed
119 V2[1],V2[2],V2[3],V2[4],V2[5],V2[6],V2[7],V2[8],V2[9]=3,3,3,3,3,1,1,0,0
    #welding speed
120
121 beta=radians(beta_deg) #Convert to radians
122 S[i][0]=crossectionarea(V1[i],V2[i],D,a_H)
123 delta_hi[i]=delta_h_i(g,beta,i,mi,S)
124 hi[i]=h_i(g,beta,i,mi,S,hi)
125 weaving_width[i][0]=weaving_trap(hi,beta,S,mi,delta,i)
126 popdelty=delta_y_trapes(i,mi,S,hi)
127 popdeltz=delta_hi[i]
128 print('Root pass calculated')
129 paramtable[i][0],paramtable[i][1],paramtable[i][3],paramtable[i][4]=1,1,
    V2[i],popdelty
130 paramtable[i][5],paramtable[i][6],paramtable[i][7]=popdeltz,theta[i][0],
    weaving_width[i][0]
131 paramtable[i][2]=V1[i]
132 #print paramtable
133
134 if delta_hi[i]<totdepth:
135     row=1
136     while delta_hi[i]<totdepth:
137         i=i+1
138         print("Starting layer %d." %i)
139         if mi[i]>1:
140             for j in range(0,mi[i]):
141                 S[i][j]=crossectionarea(V1[i],V2[i],D,a_H)
142                 for j in range(1,mi[i]+1):
143                     if j==mi[i]:
144                         #Trapeium calculations
145                         delta_hi[i]=delta_h_i(g,beta,i,mi,S)
146                         hi[i]=h_i(g,beta,i,mi,S,hi)
147                         weaving_width[i][j-1]=weaving_trap(hi,beta,S,mi,delta,i)
148                         popdelty=delta_y_trapes(i,mi,S,hi)
149                         popdeltz=delta_hi[i]
150                         paramtable[row][0],paramtable[row][1],paramtable[row][3],
    paramtable[row][4]=i+1,j,V2[i],popdelty
151                         paramtable[row][5],paramtable[row][6],paramtable[row][7]=
    popdeltz,0,weaving_width[i][j-1]
152                         paramtable[row][2]=V1[i]
153                         row=row+1

```

```

154         else :
155             #Parallelogram action
156             delta_hi [ i ]= delta_h_i ( g , beta , i , mi , S )
157             hi [ i ]= h_i ( g , beta , i , mi , S , hi )
158             popdelty= delta_y_para ( S , beta , mi , hi , delta_hi , i , j , g )
159             popdeltz= delta_h_i [ i ]
160             theta [ i ][ j -1 ]= delta_theta_para ( hi , S , beta , mi , i )
161             weaving_width [ i ][ j -1 ]= weaving_para ( S , beta , mi , hi , delta , i )
162             paramtable [ row ][ 0 ] , paramtable [ row ][ 1 ] , paramtable [ row ][ 3 ]= i +1 , j
, V2 [ i ] ;
163             paramtable [ row ][ 4 ] , paramtable [ row ][ 5 ]= popdelty , popdeltz ;
164             paramtable [ row ][ 6 ] , paramtable [ row ][ 7 ]= theta [ i ][ j -1 ] ,
weaving_width [ i ][ j -1 ]
165             paramtable [ row ][ 2 ]= V1 [ i ]
166             row= row +1
167         else : #Layer contain only one pass. Calculate trapezium variables
168             S [ i ][ 0 ]= crossectionarea ( V1 [ i ] , V2 [ i ] , D , a , H )
169             delta_hi [ i ]= delta_h_i ( g , beta , i , mi , S )
170             hi [ i ]= h_i ( g , beta , i , mi , S , hi )
171             weaving_width [ i ][ 0 ]= weaving_trap ( hi , beta , S , mi , delta , i )
172             popdeltz= delta_h_i [ i ]
173             paramtable [ row ][ 0 ] , paramtable [ row ][ 1 ] , paramtable [ row ][ 3 ] ,
paramtable [ row ][ 4 ]= i +1 , 1 , V2 [ i ] , 0
174             paramtable [ row ][ 5 ] , paramtable [ row ][ 6 ] , paramtable [ row ][ 7 ]= popdeltz
, 0 , weaving_width [ i ][ 0 ]
175             paramtable [ row ][ 2 ]= V1 [ i ]
176             row= row +1
177
178     print paramtable
179     print ( 'Weld data table written to file.' )
180
181     ### Write to file !! (Shows to alternatives. Using .CSV file extension for
simplicity
182     #table= str ( paramtable )
183     #f= open ( 'C:\Users\Bj rnEmil\Desktop\\paramtable.txt' , 'w' )
184     #f. write ( table )
185     #f. close ( )
186
187     ### CHANGE DIRECTORY HERE! ###
188     f = open ( 'C:\Users\Bj rnEmil\Desktop\\paramtable.csv' , 'w' )
189     writer = csv. writer ( f )
190     writer. writerows ( paramtable )
191
192 #Parameters
193 rootgap = rootgapProp. Value
194 platethickness = platethicknessProp. Value
195 rootheight = rootheightProp. Value
196 bevelangle1 = bevelangle1Prop. Value
197 bevelangle2 = bevelangle2Prop. Value
198
199 def generateweld ( arg ) :
200     updateVariables ( )
201     sequencer ( bevelangle1 *2 , rootgap , platethickness , rootheight )
202
203
204 def updateVariables ( ) :
205     global rootgap , platethickness , rootheight , bevelangle1 , bevelangle2

```

```
206 rootgap = rootgapProp.Value
207 platethickness = platethicknessProp.Value
208 rootheight = rootheightProp.Value
209 bevelangle1 = bevelangle1Prop.Value
210 bevelangle2 = bevelangle2Prop.Value
211
212 arg = 0
213 generateweld(arg)
214
215 global genweldProp
216 genweldProp = comp.createProperty(VC.BUTTON, 'Generate welding sequence')
217 genweldProp.OnChanged = generateweld
```

D Python Script: *robotmotion.py*

```
1 # Script for generating motion statements: robotmotion.py
2 # The script takes in matrix 'paramtable' (as a .csv file)
3 # generated by the weld sequencer. Creates a robot motion program
4 # with optional triangle weaving welding of joint based on
5 # 'paramtable' values.
6 # Adds a root pass and a cover pass at the top of the root face
7 # and at the upper surface height of the joint model.
8 # A short break is added right before ARC ON command is set on
9 # the robot controller.
10 #
11 # Sequence of operation:
12 #     1. Regenerate model
13 #     2. Generate welding sequence
14 #     3. Run
15 #
16 # Bjoern Emil Evensen                IPK, NINU
17 # Created                             12.04.2016
18 # Last modified                       25.05.2016
19
20 from vcScript import *
21 from math import *
22 import vcMatrix
23 import csv
24
25 comp=getComponent().findBehaviour('Executor')
26 mainroutine=comp.MainRoutine
27
28 app=getApplication()
29 workpiece=app.findComponent('weldpiece')
30
31 def OnRun():
32
33     ###Read csv file will print [['value','value'],[.....],[ ] etc]
34     ### Change directory here for retrieval of 'paramtable.csv'
35     paramtable = list(csv.reader(open('C:\Users\Bj rnEmil\Desktop\
36         paramtable.csv')))
37     table = [ [ 0 for i in range(8) ] for j in range(20) ]
38
39     #Translating .csv into 2D array of floats (paramtable defined as string
40     #at the moment)
41     for i in range(len(paramtable)):          #number of rows
42         for j in range(len(paramtable[0][:])):#number of columns
43             table[i][j]=float(paramtable[i][j]) #parse string to float
44
45     ###Workpiece model position and parameter information
46     m = workpiece.PositionMatrix #Position matrix of joint model
47     #Position and rotation about z-axis
48     OrigoX ,OrigoY ,OrigoZ=m.P.X,m.P.Y,m.P.Z
49     rotZ=m.WPR.Z
50     #print m.N.X, m.O.X, m.A.X, m.P.X
51     #print m.N.Y, m.O.Y, m.A.Y, m.P.Y
52     #print m.N.Z, m.O.Y, m.A.Z, m.P.Z
53     print m.WPR.X,m.WPR.Y,m.WPR.Z
```

```

54  rootheightProp = workpiece.getProperty('Root height')
55  lengthProp = workpiece.getProperty('Length bottom plate')
56  widthProp = workpiece.getProperty('Width bottom plate')
57  thicknessProp = workpiece.getProperty('Plate thickness')
58  rootheight = rootheightProp.Value
59  length = lengthProp.Value
60  width = widthProp.Value
61  thickness = thicknessProp.Value
62
63  weld_length = length - 2*10 #Small offset from edge og piece
64  frequency = 1 #Frequency of weaving oscillation [Hz]
65
66  my_statement=mainroutine.createStatement(VC_STATEMENT_HOME)
67
68  #Set base with origin at end of weldpiece
69  mtx=vcMatrix.new()
70  mtx.translateAbs(OrigoX, OrigoY, OrigoZ+rootheight) #Z value is because
71  of height of pedestole
72  mtx.rotateRelX(0)
73  mtx.rotateRelY(0)
74  mtx.rotateRelZ(rotZ)
75  my_statement=mainroutine.createStatement(VC_STATEMENT_DEFINE_BASE)
76  my_statement.Base='BASE_DATA[2]'
77  my_statement.Position=mtx
78  my_statement.Relative=True
79
80  #TCP for welding pistol in lab. Switch to TOOL_DATA[2] for implemetation
81  #mtx=vcMatrix.new()
82  #mtx.translateRel(38,121,0,290.53)
83  #mtx.rotateRelX(0)
84  #mtx.rotateRelY(45)
85  #mtx.rotateRelZ(-90)
86  #my_statement=mainroutine.createStatement(VC_STATEMENT_DEFINE_TOOL)
87  #my_statement.Tool='TOOL_DATA[2]'
88  #my_statement.Position=mtx
89  #my_statement.Relative=True
90
91  mtx=vcMatrix.new()
92  mtx.translateRel(0,10,thickness+100)
93  mtx.rotateRelX(0)
94  mtx.rotateRelY(180)
95  mtx.rotateRelZ(-180)
96  my_statement=mainroutine.createStatement(VC_STATEMENT_PTPMOTION)
97  my_statement.Base='BASE_DATA[2]'
98  #my_statement.Tool='TOOL_DATA[2]'
99  my_statement.Tool='TCPFrame'
100  my_statement.Target=mtx
101
102  ###ROOT PASS START
103  my_statement=mainroutine.createStatement(VC_STATEMENT_DELAY)
104  my_statement.Delay=0.2
105
106  mtx=vcMatrix.new()
107  mtx.translateRel(table[i][4],10,0) #y=10 for a little offset from edge
108  mtx.rotateRelX(0)
109  mtx.rotateRelY(180)

```

```

110 mtx.rotateRelZ(-180)
111 my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
112 my_statement.Base='BASE_DATA[2]'
113 #my_statement.Tool='TOOL_DATA[2]'
114 my_statement.Tool='TCPFrame'
115 my_statement.Target=mtx
116
117 mtx=vcMatrix.new()
118 mtx.translateRel(table[i][4],length-10,0) #y=10 for a little offset from
    edge
119 mtx.rotateRelX(0)
120 mtx.rotateRelY(180)
121 mtx.rotateRelZ(-180)
122 my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
123 my_statement.Base='BASE_DATA[2]'
124 #my_statement.Tool='TOOL_DATA[2]'
125 my_statement.Tool='TCPFrame'
126 my_statement.Target=mtx
127
128 my_statement=mainroutine.createStatement(VC.STATEMENT_DELAY)
129 my_statement.Delay=0.2
130
131 #Retract from weld
132 mtx=vcMatrix.new()
133 mtx.translateRel(0,length-10,thickness+100)
134 mtx.rotateRelX(0)
135 mtx.rotateRelY(180)
136 mtx.rotateRelZ(-180)
137 my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
138 my_statement.Base='BASE_DATA[2]'
139 #my_statement.Tool='TOOL_DATA[2]'
140 my_statement.Tool='TCPFrame'
141 my_statement.Target=mtx
142 ###ROOT PASS END
143
144 ### FILLER PASSES
145 ###Loop through welding sequence!
146 for i in range(0,len(table)):
147     #Count all welding sequencers, skip empty
148     if table[i][0] != 0:
149         osc = frequency*(weld.length/table[i][3])
150         x_inc = table[i][7]*cos(table[i][6]) #X displacement due to
weaving
151         y_inc = 0.25*(weld.length/osc) #Y increment
152         z_inc = table[i][7]*sin(table[i][6]) #Z displacement due to
weaving
153
154         mtx=vcMatrix.new()
155         mtx.translateRel(0,10,thickness+100)
156         mtx.rotateRelX(0)
157         mtx.rotateRelY(180)
158         mtx.rotateRelZ(-180)
159         my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
160         my_statement.Base='BASE_DATA[2]'
161         #my_statement.Tool='TOOL_DATA[2]'
162         my_statement.Tool='TCPFrame'
163         my_statement.Target=mtx

```

```

164 my_statement=mainroutine.createStatement(VC.STATEMENT_DELAY)
165 my_statement.Delay=0.2
166
167
168 mtx=vcMatrix.new()
169 mtx.translateRel(table[i][4],10,table[i][5]) #y=10 for a little
offset from edge
170 mtx.rotateRelX(0)
171 mtx.rotateRelY(180+(table[i][6])*(180/pi))
172 mtx.rotateRelZ(-180)
173 my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
174 my_statement.Base='BASE_DATA[2]'
175 #my_statement.Tool='TOOL_DATA[2]'
176 my_statement.Tool='TCPFrame'
177 my_statement.Target=mtx
178
179 ### Script for triangle weaving
180 #ARC ON _START WEAVING
181 #weld_inc = 0
182 #while weld_inc < weld_length:
183
184 #   mtx.translateAbs(x_inc , y_inc , -z_inc)
185 #   mtx.rotateRelX(0)
186 #   mtx.rotateRelY(0)
187 #   mtx.rotateRelZ(0)
188
189 #   my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
190 #   my_statement.Tool='TOOL_DATA[2]'
191 #   my_statement.Base='BASE_DATA[2]'
192 #   my_statement.Target=mtx
193
194 #   mtx.translateAbs(-x_inc , y_inc , z_inc)
195 #   mtx.rotateRelX(0)
196 #   mtx.rotateRelY(0)
197 #   mtx.rotateRelZ(0)
198 #   my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
199 #   my_statement.Tool='TOOL_DATA[2]'
200 #   my_statement.Base='BASE_DATA[2]'
201 #   my_statement.Target=mtx
202
203 #   x_inc = -1*x_inc
204 #   z_inc = -1*z_inc
205
206 #   weld_inc =weld_inc+2*y_inc
207
208 #my_statement=mainroutine.createStatement(VC.STATEMENT_DELAY)
209 #my_statement.Delay=2
210
211
212 mtx=vcMatrix.new()
213 mtx.translateRel(table[i][4],length-10,table[i][5]) #y=10 for a
little offset from edge
214 mtx.rotateRelX(0)
215 mtx.rotateRelY(180+(table[i][6])*(180/pi))
216 mtx.rotateRelZ(-180)
217 my_statement=mainroutine.createStatement(VC.STATEMENT_LINMOTION)
218 my_statement.Base='BASE_DATA[2]'
#my_statement.Tool='TOOL_DATA[2]'

```

```

219     my_statement.Tool='TCPFrame'
220     my_statement.Target=mtx
221
222     my_statement=mainroutine.createStatement(VC_STATEMENT_DELAY)
223     my_statement.Delay=0.2
224
225     #Retract from weld
226     mtx=vcMatrix.new()
227     mtx.translateRel(0,length-10,thickness+100)
228     mtx.rotateRelX(0)
229     mtx.rotateRelY(180)
230     mtx.rotateRelZ(-180)
231     my_statement=mainroutine.createStatement(VC_STATEMENT_LINMOTION)
232     my_statement.Base='BASE_DATA[2]'
233     #my_statement.Tool='TOOL_DATA[2]'
234     my_statement.Tool='TCPFrame'
235     my_statement.Target=mtx
236     ###HOT PASS AND FILLER PASSES END
237
238
239     ###COVER PASS
240     mtx=vcMatrix.new()
241     mtx.translateRel(0,10,thickness+100)
242     mtx.rotateRelX(0)
243     mtx.rotateRelY(180)
244     mtx.rotateRelZ(-180)
245     my_statement=mainroutine.createStatement(VC_STATEMENT_LINMOTION)
246     my_statement.Base='BASE_DATA[2]'
247     #my_statement.Tool='TOOL_DATA[2]'
248     my_statement.Tool='TCPFrame'
249     my_statement.Target=mtx
250
251     my_statement=mainroutine.createStatement(VC_STATEMENT_DELAY)
252     my_statement.Delay=0.2
253
254     mtx=vcMatrix.new()
255     mtx.translateRel(table[i][4],10,thickness-rootheight+2) #y=10 for a
        little offset from edge
256     mtx.rotateRelX(0)
257     mtx.rotateRelY(180)
258     mtx.rotateRelZ(-180)
259     my_statement=mainroutine.createStatement(VC_STATEMENT_LINMOTION)
260     my_statement.Base='BASE_DATA[2]'
261     #my_statement.Tool='TOOL_DATA[2]'
262     my_statement.Tool='TCPFrame'
263     my_statement.Target=mtx
264
265     mtx=vcMatrix.new()
266     mtx.translateRel(table[i][4],length-10,thickness-rootheight+2) #y=10 for
        a little offset from edge
267     mtx.rotateRelX(0)
268     mtx.rotateRelY(180)
269     mtx.rotateRelZ(-180)
270     my_statement=mainroutine.createStatement(VC_STATEMENT_LINMOTION)
271     my_statement.Base='BASE_DATA[2]'
272     #my_statement.Tool='TOOL_DATA[2]'
273     my_statement.Tool='TCPFrame'

```

```
274 my_statement.Target=mtx
275
276 my_statement=mainroutine.createStatement(VC_STATEMENT_DELAY)
277 my_statement.Delay=0.2
278
279 #Retract from weld
280 mtx=vcMatrix.new()
281 mtx.translateRel(0,length-10,thickness+100)
282 mtx.rotateRelX(0)
283 mtx.rotateRelY(180)
284 mtx.rotateRelZ(-180)
285 my_statement=mainroutine.createStatement(VC_STATEMENT_LINMOTION)
286 my_statement.Base='BASE_DATA[2]'
287 #my_statement.Tool='TOOL_DATA[2]'
288 my_statement.Tool='TCPFrame'
289 my_statement.Target=mtx
290 ###COVER PASS END
291
292
293 my_statement=mainroutine.createStatement(VC_STATEMENT_HOME)
294
295 def OnReset():
296     for k in range(0,1000):
297         mainroutine.deleteStatement(len(mainroutine.Statements)-1)
```

E Digital Appendix

The digital appendix is submitted with the electronic document as *BjornEmilEvensen-MasterThesis-DigitalAppendix.zip*. It includes:

Matlab:

sequencer.m

With Subfunctions:

crosssectionarea.m

delta_h_i.m

delta_theta_para.m

delta_y_para.m

delta_y_trapes.m

h_i.m

passlayerprop.m

weaving_para.m

weaving_trap.m

Visual Components Model:

weldpieceSIM.vcm

Containing Python Source Code:

jointmodel.py

sequencer.py

robotmotion.py

Welding Movies:

WeldTest2F.mp4

WeldTest3F.mp4

WeldTest4F.mp4