

DEGREE PROJECT FOR MASTER OF SCIENCE WITH SPECIALIZATION IN ROBOTICS
DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY WEST

Industrial robot motion control for joint tracking in laser welding

- Master degree thesis

Jiaming Gao



A THESIS SUBMITTED TO THE DEPARTMENT OF ENGINEERING SCIENCE
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE WITH SPECIALIZATION IN ROBOTICS
AT UNIVERSITY WEST
2016

Date:	June 20, 2016
Author:	Jiaming Gao
Examiner:	Fredrik Sikström
Advisor:	Morgan Nilsen, University West
Programme:	Master Programme in Robotics
Main field of study:	Automation with a specialization in industrial robotics
Credits:	60 Higher Education credits (see the course syllabus)
Keywords:	Robot motion control, Real-time, Laser welding, EGM, Joint tracking
Template:	University West, IV-Master 2.7
Publisher:	University West, Department of Engineering Science S-461 86 Trollhättan, SWEDEN Phone: + 46 520 22 30 00 Fax: + 46 520 22 32 99 Web: www.hv.se

Summary

Laser welding is used in modern industrial production due to its high welding speed and good welding performance comparing to more traditional arc welding. To improve the flexibility, robots can be used to mount the laser tool. However, laser welding has a high requirement for the accuracy in positioning the laser tool. There are three main related variables which affect the laser welding accuracy: robot path accuracy, workpiece geometry and fixture repeatability. Thus, joint tracking is very important for laser welding to achieve high quality welds.

There are many joint tracking systems which were proposed in recent years. After receiving the joint information, a control system is necessary to control the robot motion in real-time. The open control system for the industrial robot is one trend for the future. A lot of methods and systems are proposed to control the robot motion. Some systems can achieve a high accuracy in the experiments. However, it is still hard to apply them in practical industrial production. Thus more commercial solutions appear to overcome the robot motion problem nowadays. They are very useful to realize practical applications.

ABB EGM path correction module, a new function of Robotware, is one of the commercial solutions for robot motion control in real time. In the experiments presented in this work, a computer is used to simulate a sensor to create a path correction signal.

To test its feasibility for the laser welding application, many experiments are conducted. One was to test the robot path repeatability when there is no correction message sent to the robot. Another was to test the level of accuracy EGM can achieve during the correction process. Different types of paths and three different speeds were separately carried out. The results showed that it is possible to use the EGM in the laser welding application. In the EGM feasibility test, there exists deviation in the z-direction. Since these deviations are less than 0.2mm, it will have a minor influence the laser welding performance, implying that the EGM path correction can be applied in practical production.

Preface

This thesis project is the final thesis of the Master Programme in Robotics at University West. It is also connected with the research project of Robust in-process joint finding (RobIn), which is one of the projects for flexible production in Produktion2030. The experimental work in this project were conducted at Production Technology Centre (PTC) in Trollhättan. All of the test equipment is supported by PTC.

First I would like to thank my supervisor Morgan Nilsen for the help and suggestions during the whole thesis process. Next thank Fredrik Sikström for giving me advice when doing the experiments and Fredrik Danielsson for discussions about the literature review. When preparing the EGM experiment, I received technical support from Svante Augustsson, Mattias Bennulf, Xiaoxiao Zhang. At last, thank Ørjan Mæhre and Jon Tjerngren for help about EGM.

Affirmation

This master degree report, *Industrial robot motion control for joint tracking in laser welding*, was written as part of the master degree work needed to obtain a Master of Science with specialization in Robotics degree at University West. All material in this report, that is not my own, is clearly identified and used in an appropriate and correct way. The main part of the work included in this degree project has not previously been published or used for obtaining another degree.

Jiaming Gao

2016/10/13

Signature by the author

Date

Jiaming Gao

Contents

Preface

SUMMARY	III
PREFACE	IV
AFFIRMATION	V
CONTENTS	VI
SYMBOLS AND GLOSSARY	VIII

Main Chapters

1 INTRODUCTION	1
1.1 PROBLEM DESCRIPTION	1
1.2 AIM.....	1
1.3 OUTLINE.....	2
2 RELATED WORK (BACKGROUND)	3
2.1 LASER WELDING	3
2.2 JOINT TRACKING FOR ROBOTIC WELDING	4
2.3 EXTEND ROBOT CONTROL SYSTEM.....	5
2.4 CONTROL SYSTEM FOR ROBOT MOTION.....	7
2.5 COMMERCIAL SOLUTION.....	10
2.6 CONCLUSION FOR THE LITERATURE REVIEW.....	11
3 EXTERNALLY GUIDED MOTION(EGM)	12
3.1 EGM PATH CORRECTION	12
3.2 SENSOR PROTOCOL.....	12
3.3 CONFIGURATION FOR EGM PATH CORRECTION	13
4 TEST EXPERIMENT	16
4.1 ROBOT PATH REPEATABILITY TEST	16
4.2 EGM FEASIBILITY TEST FOR LASER WELDING	19
5 RESULTS.....	23
5.1 RESULTS OF ROBOT PATH REPEATABILITY TEST.....	23
5.2 RESULTS OF EGM FEASIBILITY TEST.....	25
6 DISCUSSION	30
6.1 DISCUSSION FOR ROBOT PATH REPEATABILITY TEST RESULTS.....	30
6.2 DISCUSSION FOR EGM FEASIBILITY TEST RESULTS	31
6.3 SPEED ANALYSIS FOR EGM FEASIBILITY TEST	32
7 CONCLUSION	34
7.1 FUTURE WORK AND RESEARCH.....	34

7.2	GENERALIZATION OF THE RESULT	34
8	REFERENCES.....	36

Appendices

- A. RAPID CODE IN IRB 4400 INDUSTRIAL ROBOT FOR EGM TEST
- B. C++ CODE FOR CREATING THE PATH CORRECTION MESSAGE
- C. MATLAB CODE EXAMPLE TO ANALYSIS DATA

Symbols and glossary

YAG	Yttrium aluminum garnet (Y ₃ Al ₅ O ₁₂), is a synthetic crystalline material of the garnet group. The host material of solid state laser is normally using it.
CCD	Charge-coupled device, is a device that operates the movement of electrical charge. It can convert the electrical charge into a digital value. The CCD image sensors can get a high-quality image, so it has a wide range of applications.
CMOS	Complementary metal-oxide semiconductor, is used for constructing integrated circuits. It is applied in many areas, CMOS image sensor is one of the applications in digital circuits.
CAN bus	Controller Area Network, is a vehicle bus standard and message-based protocol. It is designed for communication between microcontrollers and devices without a host computer.
PWM	Pulse-width modulation, is a modulation technique. It can encode information into a pulsing signal. It is usually applied to control the power supply of electrical equipment.
AIC	Add-in card or expansion card, is a printed circuit board that plugs into an expansion slot or other connector to provide the extra facility. It is mainly used in functionality for the system.
EGM	Externally Guided Motion (EGM) is part of the functions of Robotware. It contains two features: EGM Position Guidance and EGM Path Correction.
UdpUc	User Datagram Protocol Unicast Communication, is one of the core members of the Internet protocol which is the sending of messages to a single network destination identified by a unique address.

1 Introduction

1.1 Problem description

Laser welding is a welding technique used for joining components by melting materials together with a high-power laser beam. Due to its high welding speed, small heat-affected zone and high heating and cooling rate, laser welding is used in modern industrial production. Robot laser welding is a common application due to the robot flexibility.

This thesis work is connected with the research project Robust in-process joint finding (RobIn), which is one of the projects for flexible production in Produktion2030. The aim of the RobIn project is to develop a new and advanced laser welding solution for flexible manufacturing processes [1]. It is trying to find a new joint tracking system for laser welding. Finally, it would help the industries to save production time and improve the welding quality. However, there are two main challenges in RobIn: one is that current joint finding systems are not robust enough to fulfil the demanding applications, another is that the space requirement for sensor devices is a limitation in the interaction area where welding take place. Both regional companies and national universities are involved in contributing to solving these challenges. After getting the joint information from joint tracking system from RobIn, this thesis work focuses on how to use this information to control the welding trajectory in real-time to improve the welding quality.

To change the trajectory automatically, there are two common methods: the first is to add an adjustable extra axis which carries the laser tool; the second method directly control and adjust the robot motion. The methods and solutions using adjustable extra axis will not be considered in this work.

Added to the joint tracking during the welding process, a joint tracking system should control the robot motion, in real-time, and hence be able to control the robot trajectory. However, most industrial robot controllers do not have this function and are not open control systems. So a new control method and system are necessary to be developed to achieve real-time and high accuracy robot motion control.

1.2 Aim

The aim of this thesis project is to develop test methods for evaluating the ABB EGM of Robotware control system module applicability to laser welding applications and to test this control system module's performance for robotic laser welding application and find its limitations. The detailed aims is as follows:

- Investigate how the ABB EGM of Robotware control system module can be used for path correction
- Develop a test method for evaluating the control module applicability to laser welding applications
- Test this control system module performance for robotic laser welding application and find its limitations

- Answer the question: Is the EGM a feasible solution for robot motion control in laser beam welding applications?

1.3 Outline

In this report, chapter 1 mainly describes the aim of this thesis project. The literature review is highlighted in chapter 2. Externally Guided Motion will be introduced in chapter 3. Chapter 4 gives a detailed introduction about how to conduct the experiment. The test results are shown in chapter 5. Chapter 6 is discusses the test results. Conclusions are shown in chapter 7. Chapter 8 is the reference of the report.

2 Related work (Background)

This chapter gives the basic information about laser welding. It mainly introduces the literature study about joint tracking, the extension of robot control systems and robot motion control systems. The conclusion about literature review is presented at the end of this chapter.

2.1 Laser welding

2.1.1 Overview of laser welding

Laser welding is a welding technique used for joining components by melting materials together with a high-power laser beam. The laser beam is a high power density heat source, allowing for narrow, high-speed and deep welding. In [2], there are four kinds of lasers: CO₂ laser, lamp-pumped YAG laser, diode-pumped YAG laser and diode laser. Table 1 compares these different types of laser. There are also some other available lasers such as fiber laser.

Table 1. Comparison of laser types

Laser type	CO ₂ laser	Lamp-pumped YAG laser	Diode-pumped YAG laser	Diode laser
Advantages	Easy high power uprating	Fiber trans-mission capability	High efficiency	Low cost, small size, high efficiency
Disadvantages	No fiber trans-mission capability	Poor efficiency	Costly	Low beam quality

There are two typically types of temporal modes in the laser beam during welding, continuous and pulsed laser beam. The use of each type is application specific.

2.1.2 Laser welding characteristics

Due to its many advantages, laser welding is used in modern industrial production. Robot laser welding is a common application due to the robot flexibility. The main characteristics of laser welding are [2]:

- High-speed
- Deep-penetration
- Reducing thermal distortion and heat effected zone
- Higher accuracy and higher efficiency

Though laser welding has many advantages, there are some disadvantages which make it not conventional in industrial application.

The disadvantages of laser welding:

- The need for strict and precise joint control
- Weld materials requirements
- High-cost equipment

With the development of technology, some of the demerits of laser welding will be conquered.

2.2 Joint tracking for robotic welding

2.2.1 Overview of joint tracking techniques

In comparison to the other welding processes like arc welding, laser welding can achieve higher accuracy, better welding performance and higher requirements about path accuracy. There are three variables: robot path accuracy, work piece geometry and fixture repeatability. In industrial production, joint tracking is commonly used to realize laser welding accuracy.

There are two main methods for joint tracking: contact and non-contact sensors. For the contact sensor, mechanical guidance such as a fixed guide tip mounted on the wirefeed mechanism or a tracking wheel is used to track the welding joint. This will touch the workpiece and must follow linear joint. The non-contact sensor is mainly based on the triangulation principle. There are many advantages with the non-contact sensor: the sensor does not touch the joint between the two materials; high temporal resolution and reliable joint detection. Optical sensors such as CCD or CMOS camera are common to measure the joint at different distances. [3]

There are also two ways for sensor arrangement: fixed to the robot hand and fixed to the welding head. Fixing to the robot hand, the robot hand is between sensor and laser tool. The joint position which is detected by the sensor needs to transfer through the robot hand to the laser tool. The number of errors would increase during the transferring. The first sensor arrangement is an open loop control, fixing to the welding head is a closed loop control. Considering the accuracy, it is usually mounted in the welding head, which could reduce frame transfer error.

There have been many joint tracking methods and systems proposed in recent years. This chapter will introduce some feasible joint tracking systems.

2.2.2 3D and 2D visual information fusion vision sensor system

The joint is narrow and burred with a gap width less than 0.1mm in some welding cases. When considering a 3D curved joint and heated induced deformation during the welding process, there will make it difficult for a normal vision system to get high precision joint information.

A new 3D and 2D visual fusion vision sensor system (named HUST-SM) is proposed to be applied as a laser welding joint-tracking in [4]. An 8-axes machine tool is used to carry the vision sensor system and perform motion control to get the 3D surface joint information.

In the HUST-SM, the vision system can get joint information which contains joint width, joint position and the normal direction of the local joint surface. Next, the system converts this information from local coordinates to the workpiece coordinates and makes a joint model. Then the system is used to control the axes motion to plan laser focal point motion path. Finally, the system detects joint position and calculates online right direction to compensate the joint trajectory offset.

There are two experiments to test this system. One is HUST-SM sensor test, another is laser welding experiment. In both tests, the process used 45 steel as the welding material and joint gap width is less than 0.1mm. In the HUST-SM sensor test, the

HUST-SM results would be compared with TESA-VISION instrument (An image measuring equipment) results. In the laser welding experiment, the author compares the welded joint between compensated path and uncompensated path.

In the HUST-SM sensor test, the results show that the HUST-SM system could get precise joint information. In the laser welding experiment, the results show that it can get better welding quality with the compensated path.

2.2.3 Automatic seam tracking system based on laser visual sensing

There are many seam tracking methods, and laser-based vision sensor is commonly used because of its high speed, accuracy and robustness. Two laser sensors are used to acquire the seam information in [5].

Luo [5] applies this system in a five-axes servo robot which carries a laser-based vision sensor in front of the torch. The frame-grabber board or a data-acquisition board and an eight-axes motion-controller card are added in the computer to acquire seam information and control the robot.

Two laser displacement sensors with two laser position-sensitive-detectors inside are used to scan the workpiece. It can scan around 35mm and get 1000 points in 20ms and return to the start point in 10ms. There exists distorted signals due to the shiny surface, but the groove profile could be built from the sensor.

The structure of this system is as follows:

- Acquiring the feature template.
- Automated to find the start welding point.
- Automated calibration.
- Acquiring seam information.
- Path control and parameter modification.

To find the start welding point, a feature template is needed. The robot will automatically adjust and find the start welding point. There are some missing points in the feature finding process, so two-point linear interpolation is used to find missing point. Finally, a seam welding path conversion algorithm is applied to convert the seam path to the robot welding trajectory.

Experiment was carried out to test this system. The accuracy of seam tracking less than 0.4mm is realized in this system. Feature template and auto calibration algorithm are proposed to find seam and start welding point.

2.3 Extend robot control system

2.3.1 Overview of extend robot control system

Many joint tracking systems are presented in the above chapter. To transfer the joint position to the robot path, a robot control system is required to execute and rectify the robot trajectory. Nowadays, most industrial robot controllers are not open control systems. Thus, extending the robot control system is necessary to deal with the joint tracking data and control the robot.

The extended robot system is normally developed as an open system. Robots are commonly used in industrial production. The robot should be suitable for different manufacturing tasks. An open control system could help the robot connect with exter-

nal devices and deal with external signals. Blomdell and Bolmsjö proposed two fundamental questions for the open system: current industrial controllers should be useful as components in future advanced robot systems; Commercial/optimized systems should be structured to allow flexible extensions, even on a hardware and real-time level. [6]

2.3.2 Extending industrial robot control system

Most of the industrial robot controllers are not open system, so it is hard to directly control robot motion and get feedback to let the robot be more flexible in industrial applications. In the design and implementation, ABB S4CPlus controller is used to develop and test in [6].

In the design of extending control system, high-level (user-level) usage of low-level (primitive) sensors and low-level (motion control) usage of high level (force, vision, etc.) sensors are used to achieve good performance and flexibility. Peripheral connection interface system bus and Ethernet communication are used to connect S4CPlus controller for sharing memory. For the safety and quality consideration, an application software which could be test dummy sensor data without actual hardware should be developed to enhance hardware reliability. An added board should be used to store external common data before allowing access to the robot controller. A neutral definition of exposed variables should be used to define shared data. The external software should get internal state information and be cross-checked before effecting the robot control. It is important to always keep the internal safety functions activated.

In ABB controller, it supports the sensor's feedback to the correct robot program and path by correction generation. In [6], in order to compatible with standard RAPID, XML comments are used to communicate between controller and external added board. MATLAB/Simulink is used for design and simulation.

In the experiment, robot force-controlled deburring is a case study carried out by a company. The experimental results show that the contact force is around 150N during the grinding process. The experiment proves that the extending controller works in the industrial robot. A disturbance in the result is because of the resonance in the work-piece.

This new extended industrial robot control system is proposed and applied in the real experiment. Shared memory, integrated high-level and low-level control, external sensor and board control, engineered view to design and deploy make this new extended control system more practical and applicable in the real industrial robot. The extended control system makes the industrial robot applicable in more areas and effective in achieving the task requirements.

2.3.3 Real-time open robot control system

The main manipulator robot controller is still a closed architecture system. It is hard to change and reconfigure. So an open control system which allows the addition of new software and hardware is necessary.

The control system, proposed in [7], is based on distributed architecture, called AIC. It controls each joint of the robot through PWM converter and interface. CAN bus is an open protocol and is used in this system for real-time data transfer. A host computer is connected to each AIC through the CAN bus. Then each AIC directly drives robot DC motor.

Open Robot Control Software project which supports open robot control software package is used in this proposed control system. This project depends on four C++

libraries: The Real-time Toolkit which operates on real-time, on-line robotic application; the components Library which offers some component models; The Kinematics and Dynamics Library which could support the calculation kinematic chains in real-time and The Bayesian Filtering Library which provides an independent framework for inference in Dynamic Bayesian Networks.

Some component models are created for interface between Janus and Open Robot Control Software. It includes AIC, controller and bridge. The AIC component model is created to command the AIC card and see the communication details. The PID component model is created as an independent PID controller to connect AIC. The Bridge component model is created to convert data formats between robot oriented in Open Robot Control Software and joint-oriented in PID or AIC.

The proposed control system is implemented in the Janus robot which consists of a two-armed robot and a stereo vision system. The trajectory tracking experiment is carried out to test this proposed control system. The desired trajectory and the measured trajectory are compared

From the experimental result, the measured trajectory is close to the desired trajectory. This proves that this new open control system is feasible. Based on Open Robot Control Software, the control system is more flexible. It can realize real-time robot motion control by controlling the robot joint.

2.4 Control system for robot motion

At present, many robot control systems are proposed and developed to control the robot motion. This chapter shows some research about robot motion control system in recent years.

2.4.1 Control system with real-time seam tracking method

In [8], the author presents two new systems: one is the vision sensor system which could get clear and real-time seam information; the other is the control system which is fast and steady to correct robot (torch) path.

The vision sensor system contains a CCD camera, image acquisition card, automatic transmission mechanism, dimmer-filter system etc. This system can realize many functions, like remote controlling, seam tracking etc. The dimmer-filter system is divided into four tiers: shading, dimming, filtering and the extending layer, to eliminate the disturbance and adapt to more environments.

In the image processing, an improved detection algorithm is proposed to detect the seam edge. A verification of image processing is carried out to test the new algorithm's precision. From the verification results, the image processing tolerance is about $\pm 0.169\text{mm}$.

A multi-thread program is developed for control software system. The main thread controls arc start/end, initialization and image display. The sub-thread is for image acquiring and processing. Segmented self-adaptive PID controller is proposed to control the seam tracking process. This new PID controller can automatically choose the parameters on the basis of the offset between seam and torch. The input "e(t)" is offset between seam and torch and the output "u(t)" is rectifying voltage. This control system combines with vision sensor system to realize real-time seam tracking.

The welding experiment is carried out to test this system. The experimental device contains robot system, vision sensor devices, the isolation unit, the weld power supply

and the computer. There are two teaching trajectories: straight line trajectory and fold line trajectory. Both of them are tested to assess this new method feasibility.

In the straight line teaching trajectory experiment, the result shows that the tolerance of tracking is within $\pm 0.27\text{mm}$. In the fold line teaching trajectory experiment, the tolerance of tracking is within $\pm 0.3\text{mm}$. Both of them fulfil the real-time seam tracking requirement. The experiment proved that this new real-time seam tracking method and control system are feasible and stable.

2.4.2 Control system for improving ABB industrial robot based on external sensing

Due to the bandwidth limitations, longer duty time is required to acquire and reflect the external feedback. For many application, real-time, fast feedback and robot reflection are very important. Thus, the author proposed a new control system to achieve high-performance motion control based on external feedback and applied it to ABB industrial robot. [9]

A binary protocol (LabComm) is used in this system to handle interface changes and software incompatibilities. Data Description Language which represents data sent by LabComm, generates marshalling/DE marshalling code for some languages such as C, Java etc. While one-way communication is required to carry out data logging, throttling guarantees the sensor data package reaches the low-level motion in time.

Open Robot Control Architecture (ORCA) is a two-way protocol built on the top of LabComm protocol and Matlab/Simulink Real Time Workshop is used to generate code for ABB robot. So a tool is developed for converting code from Matlab C-code to ORCA. For the Simulink Controller, C-code generation is used in common blocks. The controller can be simulated any time before applying in the real robot. A driver routine is necessary to receive and send the data by LabComm protocol, when input and output signal adding to the Simulink controller.

In the experiment, an iterative learning control of the parallel kinematic robot is tested. By decreasing the deviation with reference trajectory in each iteration, the robot tries to learn a specific motion. Updating the robot joint position references are through correction of the term between the iteration. From the experiment results, it is clear that the tracking error has a large reduction after one iteration. The latency between detection and motor reaction is less than 1ms. In [9], the author proposed a new control system to handle external feedback and control robot motion. The experiment has proved this system is feasible.

2.4.3 Trajectory-based control architecture

In [10], the sensor is mounted in front of the laser focal point to measure the seam trajectory to realize real-time seam tracking. The sensor is considered to use a camera and imaged-based sensor. There are two kinds of control architectures: position-based control and image-based control. However, the time delay in the control system and the different cycle time between sensor and robot would influence the accuracy. Synchronisation in the control system is very important.

The author proposed a new solution that the sensors' measurements related to the robot position are stored in a buffer. In the buffer, a real-time trajectory generator is used to control the laser focal point to follow the seam trajectory with a predefined speed. Some filters and corrections are used to reduce the measurements fluctuations and smooth the robot motion.

Tool Trajectory Buffer is used to store the robot tool locations needed for the trajectory. These locations will update in the real-time robot motion. Real-time Setpoint Generator interpolates the locations and computes location setpoints every 4ms. Inverse Geometric Model computes the robot joint angle. These locations will be sent to Joint Motion Controller to control robot motion. Then, the sensor image is handled by the Image Processing. The robot joint information and position are obtained and synchronised together to compute the seam locations which are stored in Seam Trajectory Buffer.

There are some applications of this trajectory-based control approach: teaching a known seam trajectory; teaching an unknown seam trajectory; real-time tracking a known seam trajectory and real-time tracking an unknown seam trajectory.

As for the seam model, the trajectory could be regarded as a continuous curve in 3D space. Many discrete points are 3D vector, located in the seam trajectory. Cubic spline is used to interpolate new locations in every segment.

In the real-time seam tracking, there are two steps to carry out. In the first step, the sensor would begin to work at the starting point of the nominal seam trajectory. The actual seam trajectory would be measured by the sensor. When the laser focal point arrives at seam trajectory, it would begin to weld in the actual seam trajectory which is measured by the sensor ahead of the robot. In the second step, the laser focal point continues to weld the actual seam trajectory. Until the sensor passes the end point of the nominal trajectory, no new locations will be updated in the actual seam trajectory. The laser focal point continues to weld until it arrives at the end point.

In the real-time seam tracking, the orientation of actual locations would be computed in the Seam Trajectory Buffer. Then, filtering is used to remove the noise.

In the experiment, two kinds of seam trajectory: line and curved sine, are tested. The velocity used in the experiment is 100mm/s for line and 50mm/s for curved.

In the experiment, each kind of trajectory consists of teaching and tracking. In the real-time line trajectory teaching experiment, using both sensor tool frame and laser tool frame have high accuracy within 0.1mm. In the tracking experiment, using sensor tool frame also maintains very high accuracy. There are some fluctuations by the laser tool frame but still mainly within 0.1mm. The results are similar to teaching experiment and most do not need orientation correction.

In the real-time curved sine trajectory teaching experiment, the sensor tool frame is used for the test and the measured results are mainly within 0.2mm. When using laser tool frame, the accuracy is about 0.3mm which is beyond the required 0.2mm. In the tracking experiment, the accuracy of using sensor tool frame and laser tool frame are about 0.4mm and 0.5mm, which also does not fulfil the requirement. Graaf [10] thinks these errors are caused by the geometric robot errors. Since there are many locations that needed orientation correction, the robot also needs the corresponding orientation to get to the measured location.

A real-time seam tracking algorithm and a trajectory-based control architecture are presented and tested in [10]. In the control approach, the buffer is used to store predefined and actual locations information. Real-time Setpoint generator, synchronise method, sensor image and filtering are used to get precise actual seam trajectory. Through the experiments, this trajectory-based control approach is shown to be an appropriate method for real-time seam tracking. However, more work needs to be done to improve the accuracy.

2.4.4 Robot visual control system based on a local network with a multi-level hierarchy

There are three kinds of visual control methods: position-based, image-based and hybrid methods in [11]. The hybrid methods contain both of the former methods' advantages to control the translation in image space and rotation in Cartesian space.

In [11], the author proposed structured light stereovision which combines structured light and stereovision together to measure the weld seam. A hierarchical visual control system which contains human-machine interface (HMI) level, motion planning level, motion control level and servo control level is proposed to achieve real-time control. Human-machine interface level is designed for HMI, task planning and image processing. It also gives the basic motion information such as position and pose parameters. Next, motion planning level calculates and provides position values for all joint motors according to these parameters. Then, motion control level receives all position values from the motion planning level and measures the actual position values as a feedback. The servo control level directly controls the robot through servo amplifiers. The HMI level is located in a master computer and the other three levels are located in an open robot controller. Due to the small exchange between master computer and open robot controller and fast local network, this visual control system can work in real-time.

To test this new control system, tracking and jointing experiments are executed. The tracking speed is set to 0.03m/s and the jointing speed is set to 0.003m/s. The weld seam is a V-groove weld. The experimental results show this new visual control system works effectively. The robot can realize real-time seam tracking and motion control. It proposed some good methods for seam parameter extraction and seam tracking. The experiment had already proved the visual control system works stably and efficiently.

2.5 Commercial solution

To improve the welding quality, a sensor is necessary for robot welding. It can be used for joint finding, joint tracking, adaptive control and quality monitoring. In the joint finding and joint tracking, some commercial sensors and control systems are introduced.

2.5.1 Joint tracking system

In [12], there are four available commercial sensors used for joint tracking. Robo-Find system is based on laser vision system for off-line joint finding. It locates, detects and measures weld joints and controls the robot to adjust trajectory less than 1s.

Power-Trac system can achieve real-time joint tracking and off-line joint finding which are also based on a laser vision system. The trajectory of the torch is continuously rectified during the welding process.

Laser pilot is used for joint tracking and joint finding. It can correct positioning errors and thermal distortion errors.

Circular Scanning System Weld-Sensor emits a laser beam through an off-axis lens onto the surface and receives the distance information through triangulation method. It could potentially measure the joint information in the welding process.

Permanova WT04 Laser Welding Tool System [13] is based on a joint tracking module. The built-in vision control of this system allows joint tracking very close to the welding zone. More modules can be integrated in this system.

2.5.2 Commercial control system

ABB Weldguide III consists of two external sensors for welding current and arc voltage. It can realize fast and accurate path correction and could be integrated with different transfer modes. It has a basic mode in which the distance is maintained, an advanced mode which could identify and modify variations in joint tolerance, and multi-pass modes which track the first pass and store the actual tracked path.

ABB Externally Guided Motion of Robotware control system helps operators control the robot motion more precisely [14]. It can use the input position from the external sensor to modify the robot path. It can realize real-time control because the updating time is between 4 to 20 milliseconds.

Precitec WeldMaster systems [15] is developed for real-time process control and quality monitoring of a laser beam. It can read all the measured data in real-time and monitor the laser welding process.

2.6 Conclusion for the literature review

From the literature study, there are many research projects about joint tracking and robot control system.

For joint tracking, the non-contact sensor is often used to acquire the joint information, such as laser sensor, CCD camera etc. These sensors prefer to be fixed on the welding head due to its higher accuracy. There are usually two or more sensors used for joint tracking system. For example, 3D and 2D visual information fusion vision sensor system for the laser welding joint-tracking. It combines the 3D triangular method and the 2D image measurement method to get precise joint information. The combination of different sensors is the trend in industry and would make joint tracking more precise in the future.

Extending control system for the industrial robot is necessary for robot motion control. An open control system could allow flexible extensions. Thus, the open control system for the industrial robot is one for the future, even if most of the industrial robot controllers are not open now.

For the robot motion control, most of the research uses an extending robot control system to realize motion control. For example, Trajectory-based control architecture and a multi-thread program for control software system have been proposed for robot motion control. All of these proposed methods and systems can be controlled in real-time and perform effectively in the experiment. However, comparing to the laser welding requirements, the results of the test show that high accuracy was not achieved. None of these methods can be applied in the practical production.

Nowadays, many commercial solutions appear to solve the robot control problem. These commercial solutions could be very helpful in realizing the practical application. A real-time robot motion control system which is based on commercial solutions is possible for laser welding in the future.

3 Externally Guided Motion(EGM)

Externally Guided Motion (EGM) is part of Robotware which is an ABB robot control software. It contains two features: EGM Position Guidance and EGM Path Correction [16]. EGM Position Guidance allows the robot to move to a given position which is given from an external device. It is mainly used for taking and placing objects in special locations. Using correction messages which are sent from the external robot mounted devices, EGM Path Correction guides the robot along the correct path. The application of this feature is to track the joint or objects. For this project work, EGM Path Correction is used to test and control the robot motion in real-time.

3.1 EGM Path Correction

EGM Path Correction can correct a programmed robot path with a maximum update frequency around 24 Hz. For the laser welding application, the tracking system sensor has to be mounted on the robot and it must be possible to calibrate the sensor frame. The path correction is executed in the path coordinate system. However, for the EGM Path Correction, the path can only correct in y- and z-directions, it cannot change the orientation and correct in x-direction [16].

There are three states when using EGM to control the robot. In the EGM_STATE_DISCONNECTED, the setup is not active. In the EGM_STATE_CONNECTED, the setup has been made, but no movement is done. In the EGM_STATE_RUNNING, the robot movement is executing.

There are two kinds of EGM setup interfaces for input data selection. When selecting a signal interface, EGMSetupAI, EGMSetupAO, EGMSetupGI are used. EGMSetupUC is the instruction for an UdpUc (User Datagram Protocol Unicast Communication) interface. However, the output data is only available for the UdpUc interface. The explanations see Table 2.

Table 2. The explanations of EGM input and output instruction.

Instruction	Description
EGMSetupAI	Establish analogue input signals for EGM
EGMSetupAO	Establish analogue output signals for EGM
EGMSetupGI	Establish general input signals for EGM
EGMSetupUC	Establish UdpUc protocol for EGM

There are two different modes, joint mode and pose mode, to control the robot motion in EGM Path Correction. Axes angles are given to control the robot motion in joint mode. For the pose mode, reference frames such as Tool frame, Work object frame, Correction frame and Sensor frame are necessary.

3.2 Sensor protocol

The sensor protocol is designed to communicate sensor data regularly between the robot controller and sensors. Google Protocol Buffers [17] are used for encoding and User Datagram Protocol (UDP) is used for transport protocol. The sensor starts to send messages until it receives the first message from the robot controller. The first

message is a data message and messages can be sent independently. However, a sender of a UDP message is sent continuously even if the receiver's queue may be full.

Google Protocol Buffers or Protobuf, are an efficient way to serialize/de-serialize data. The EGM .proto file has already defined the EGM sensor protocol data structures. Following the defined structures, the compiler can generate serialized/de-serialized code. After handling by the compiler, the application creates a message, calls serialization method and then sends the message. The Protobuf is language neutral, so many programming languages can be used. Due to its speed and language neutrality, Protobuf is used to encode.

User Datagram Protocol is one of the important members of the Internet protocol suite in transport layers. There is no handshake in UDP, so the delivery, ordering, or duplicate protection cannot be ensured. If the application has requirements about error checking and correction, UDP should not be used. However, the data is sent in real-time with high frequency through the UDP. Thus, UDP is chosen as the transport protocol in EGM communication.

3.3 Configuration for EGM Path Correction

3.3.1 Working principle

The Figure 1 is showing how the EGM Path Correction works.

Here is a simple description about working principle of EGM Path Correction:

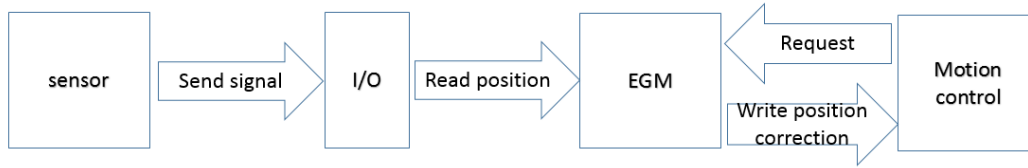


Figure 1 Working principle of EGM Path Correction

1. Sensor sends the signals to I/O module multiple of 4ms.
2. Robot sets up the interface for input data (Or establish communication between robot and sensor).
3. Motion control sends the request for EGM multiple of 4ms.
4. EGM reads position data (for path correction only y and z values) from I/O module.
5. EGM calculates the position corrections and writes position corrections to the Motion control.

3.3.2 System parameters setup in robot controller

There are many system parameters which needs to be configured before using the EGM. Some system parameters can affect the EGM behaviours, such as the parameters in External Motion Interface Data in 'Topic Motion' [18].

'EGMSetupUc' is used to setup the UdpUc protocol for EGM [19]. There is one argument, named 'ExtConfigName', which needs to be created before using the 'EGMSetupUc'. In 'ExtConfigName', three kinds of levels can be chosen. They are Raw which applies raw correction and is added just before the servo controllers, Filtering which applies extra filtering on the correction and Path which corresponds to path correction. Filtering is used for EGM Position Guidance and Path is used for EGM

Path Correction. Each level determines which of the corrections are applied in EGM. The default level is Filtering. Thus, it is necessary to create a new 'ExtConfigName' for path correction.

Besides the level, another two parameters can influence the EGM control system. Default Proportional Position Gain determines the default proportional gain of the EGM position feedback control. It affects how fast the response moves to the target position. The target position is given by the sensor corresponding to the current position. Higher values can lead to a faster response. Default Low Pass Filter Bandwidth Time determines the default value used to filter the speed contribution from EGM. Figure 2 shows the simple view of EGM control loop.

Another argument 'UCDevice' also needs to be pre-setup, 'UCDevice' is an UdpUc device name. UdpUc device is used by the robot controller to initiate the connection to

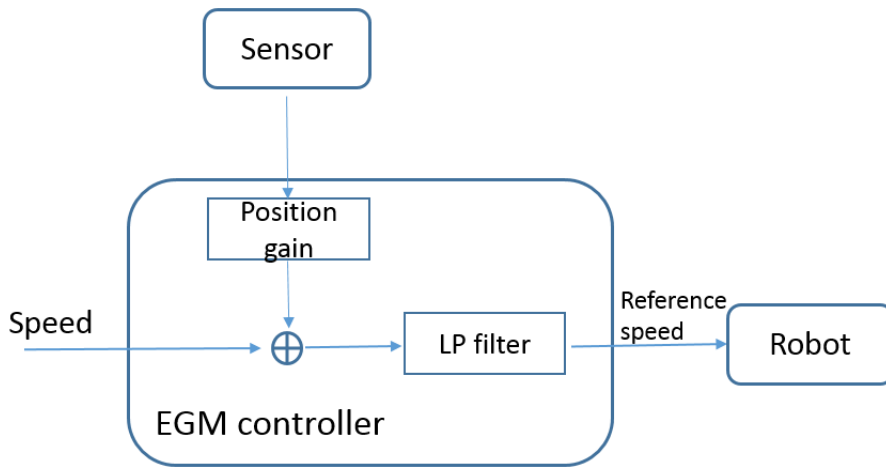


Figure 2 Simple view of EGM control system

the computer or sensor. The system parameters which are used to configure UdpUc device belong to the type 'Transmission Protocol' in 'Topic Communication'. The parameter type is set to 'UDPUC' and the parameter Serial Port is not used and shall be set to N/A. Remote Address is the IP address of the remote device, such as a computer or a sensor. Remote Port Number specifies the IP port number identified by the Remote Address in order to build the connection.

Other arguments in the 'EGMSetupUc' are chosen depending on the real test situation.

3.3.3 RAPID instructions about EGM

EGM contains many RAPID instructions, here some basic RAPID instructions were introduced which will be used in EGM path correction test. The RAPID code example can be seen in the appendix A.

'EGMGetId' and 'EGMReset' are used to reserve and reset EGM identity. This identity is applied in all other EGM RAPID instructions and functions to identify a specific EGM process.

'EGMSetupUc' is used to establish an UdpUc protocol for a specific EGM process. The robot must have 6 axes. The argument 'MecUnit' is the name of the robot that will be guided and 'EGMid' is the specific EGM process which is already defined and identified. For the argument 'ExtConfigName' and 'UCDevice', they have been setup as

system parameters in chapter 3.3.2. There are three options for 'EGMSetupUc' to decide which mode will be used for EGM. Both [\Joint] and [\Pose] are used to position guidance. Hence only [\PathCorr] mode will be applied in this thesis experiment. If [\PathCorr] mode is chosen, [\APTR] or [\LATR] must be selected as the sensor tracking type for path correction. [\APTR] is to setup an at-point-tracker type of sensor and [\LATR] is to setup a Look-ahead-tracker type of sensor.

'EGMActMove' activates a specific EGM process and defines static data for the EGM path correction movement. 'EGMid' is the specific EGM process which should be the same as used in the 'EGMSetupUc'. 'SensorFrame' is used to interpret the sensor data. [\SampleRate] is another argument to define input data reading sample rate and it should be a multiple of 24ms.

'EGMMoveL' and 'EGMMoveC' are the instructions to execute the EGM path correction. 'EGMMoveL' is to move the tool centre point (TCP) linearly to the given robot target and 'EGMMoveC' is to move the TCP circularly to the given robot target. These two instructions can design the robot path. If there is no correction data or information from the sensor or computer, the robot just executes the designed robot path. In this case, 'EGMMoveL' is the same as 'MoveL' and 'EGMMoveC' is the same as 'MoveC'. On the contrary, the robot will follow the path correction data to move.

'EGMGetState' is used to retrieve the state of the specific EGM process. It is a function in the RAPID to show the EGM process state. It can help to check whether EGM is connected and whether EGM is running.

The simple description for EGM RAPID instructions sees Table 3.

Table 3 Simply description for EGM instructions

Instructions	Description
EGMGetId	Reserves EGM identity
EGMReset	Resets EGM identity
EGMSetupUc	Establishes an UdpUc protocol for a specific EGM process
EGMActMove	Activates a specific EGM process and defines static data for the EGM path correction movement
EGMMoveL	Moves the TCP linearly to the given robot target
EGMMoveC	Moves the TCP circularly to the given robot target
EGMGetState	Retrieves the state of the specific EGM process

4 Test experiment

Externally Guided Motion (EGM) is a new function of ABB Robotware control system. When applying the EGM to laser welding, it is necessary to evaluate how fast the EGM can react and how accurate it is. Thus, some experiments are carried out to test the EGM performance.

The experiments are carried out in an industrial robot, specifically the ABB IRB 4400 industrial robot. A robot operating at different speeds could perform differently. Therefore, robot path repeatability, without path corrections, is also evaluated for ABB IRB 4400 industrial robot. The experiment is setup as Figure 3. UdpUc protocol is setup to build the communication between computer and robot. The fiber laser tool is mounted to the IRB 4400 industrial robot. When defining the laser tool in the robot, the max payload 60kg is used and the center of gravity is 900mm in z-direction.

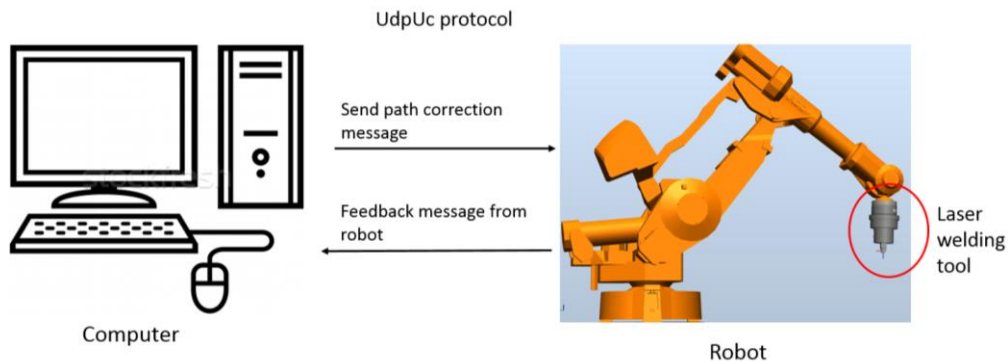


Figure 3 The experimental setup schematic graph

4.1 Robot path repeatability test

Robot path repeatability test is very important for laser welding. If there is no correction message sent to the robot, the nominal robot path will be executed. In that case, robot path repeatability will influence the laser welding accuracy. Thus, it is better to evaluate the path repeatability without using path corrections for this ABB industrial robot.

The test experiments are divided into three groups depending on the different shape of the designed robot path (or taught trajectory). These groups are straight line, fold line and curved line. A simple view of the three different designed robot paths is shown in Figure 4.

In the robot path repeatability test, three different robot speeds are used. They are 5mm/s, 10mm/s and 20mm/s. The reference work object is Wobj0. The robot position is saved every 0.1s during the test process. The example code can be seen in appendix A, trap routine is used to save the robot position in a removable disk.

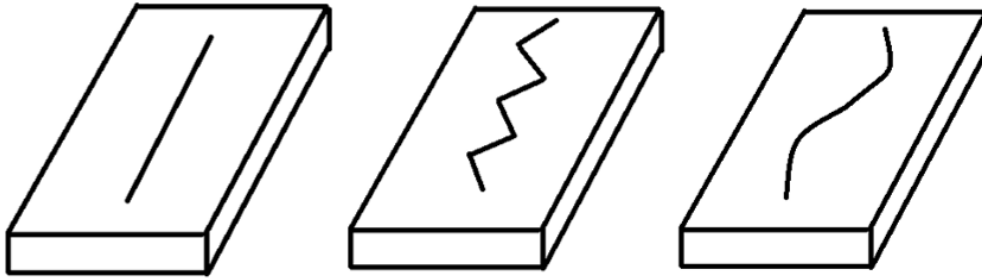


Figure 4 Three kinds of designed robot path: straight line, fold line and curved line

4.1.1 Straight line robot path

When designing straight line robot path, the robot only moves along the x-axis. There is no movement in y- and z-axis. The length of the straight line is 300mm. Using 'EGMMoveL' instruction to control the robot from robot target p20 and linearly move to target p60. The coordinate value of target p20 is (510, 1245, 235) and target p60 is (170, 1245, 235). Figure 5 shows the designed straight line robot path. When the robot starts to linearly move from the robot target, it will start to save the robot position in the specific file every 0.1s. It will end to save until robot reaches target p60. The detailed robot RAPID code is shown in appendix 1.A.

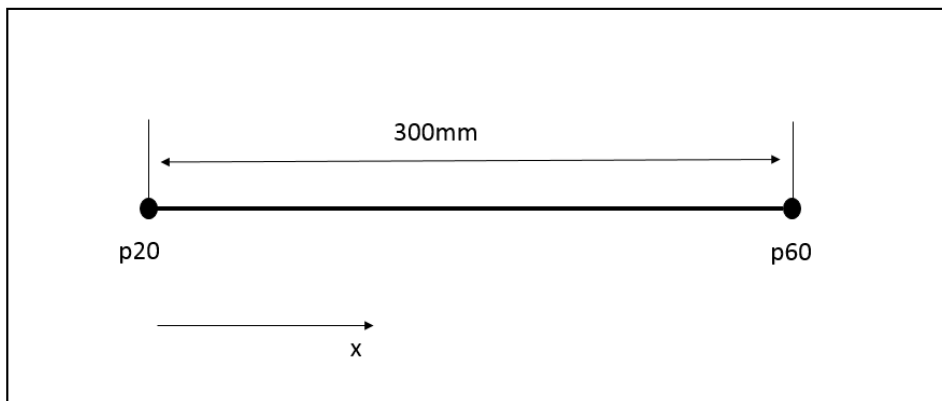


Figure 5 The designed straight line robot path

4.1.2 Fold line robot path

When designing fold line robot path, three types of experiments are carried out. The first test is movement in x- and y-direction, keep z direction constant. The second test is movement in x- and z-direction, keep y-direction constant. The third test is 3D movement with movement in x-, y- and z-direction. The length of the straight line is 300mm. Robot path starts from target p20 to p60. The coordinate value of target p20 is (510, 1245, 235) and target p60 is (170, 1245, 235). It will pass through target p30, p35, p40, p45 and p50. These targets have 50mm offset comparing the former robot target. The simplified diagram of the designed fold line robot path shows in Figure 6. 'EGMMoveL' instruction is used to generate the fold line path. Each type of experiment is also tested with three different speeds: 5mm/s, 10mm/s and 20mm/s.

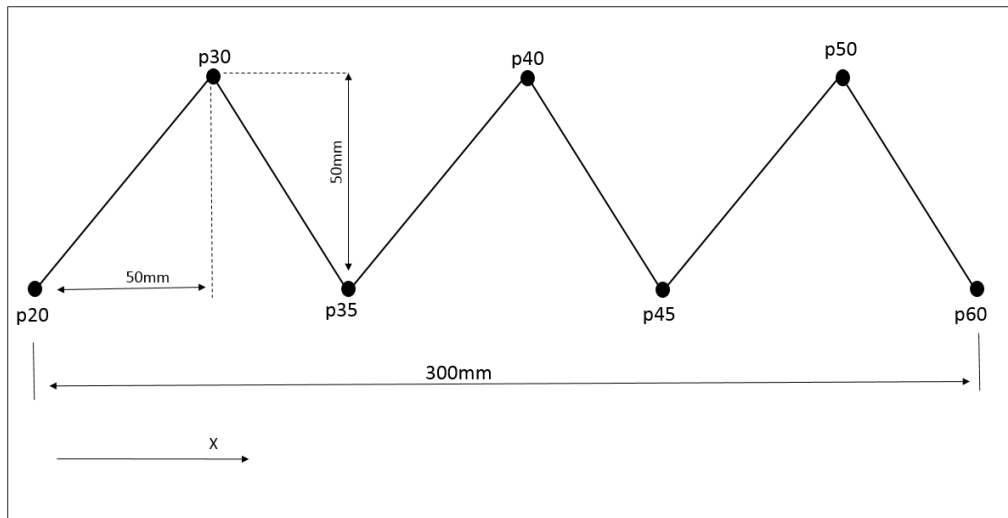


Figure 6 The designed fold line robot path

4.1.3 Curved line robot path

Here two types of experiments for curved line robot paths are presented. One test is movement in x- and y-direction, while keeping z-direction constant. Another test is movement in x- and z-direction, while keeping y-direction constant. The length of the straight line is 300mm. Robot path is start from target p20 to p60. In the designed curved line robot path, there are two different curvatures: one is from p20 to p40 and one is from p40 to p60. RAPID instruction 'EGMMoveC' is used to create the curved line robot path. Figure 7 shows the sketch of the designed curved line robot path.

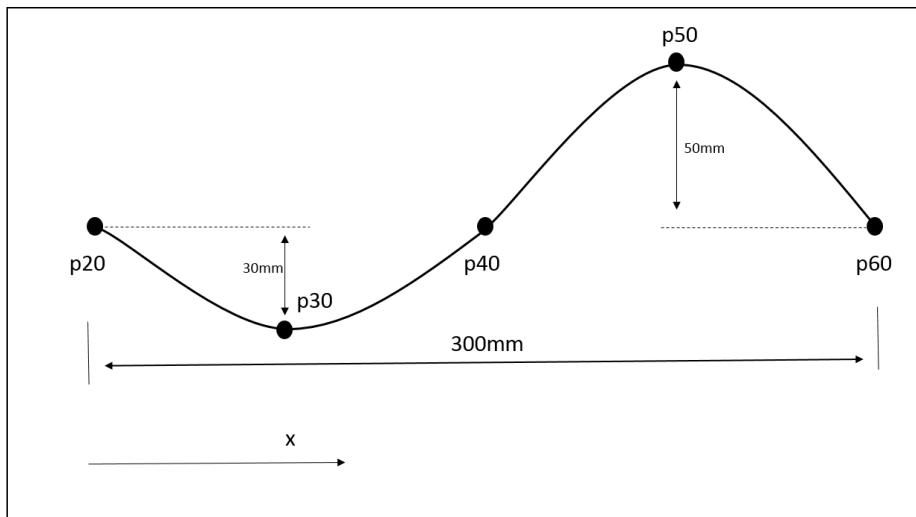


Figure 7 The designed curved line robot path

4.2 EGM feasibility test for laser welding

To test the EGM feasibility for laser welding, four kinds of experiment are carried out. There are three kinds of corrections conducted in the straight robot path. The last experiment is curved line correction in the curved robot path. All of the path correction signals are simulated by the computer. When doing some trials for EGM path correction, the laser tool starts to shake if the correction message is too large. For example, if the increment of correction message is larger than around 0.1mm in the straight line correction test with speed 20mm/s, the laser tool starts to shake. This shaking can influence the weld performance and is not permitted to occur during the laser welding process. Thus, all the correction paths are not changed too much to guarantee no obvious shake during the test process.

4.2.1 Create path correction signal

EGM has two kinds of interfaces setting for input data. One is for a signal interface, which can connect to a sensor and get values from an analogue input. Another is for UdpUc interface, which can directly connect to the computer. For signal interface, a sensor or a joint tracking system is needed to get joint information and send the actual robot target to the EGM controller. When using the signal interface to do the experiments, the actual joint shape should be created by the welding parts. The welding parts have difficulty making all kinds of joint shapes to do the experiment. Thus, the UdpUc interface is setup for all the text experiments. ABB support an EGM library for C++ to create path correction data. It can simulate a sensor to create a signal and send it to the robot in real-time through the Google Protocol Buffers [17] and UDP protocol. Thus, all of the test experiments are setup UdpUc interface in this thesis work.

The detailed process about how to create path correction data using C++ is shown in the Application manual Software IRC5 [16]. Here just a general description of how to create path correction information is given. Microsoft Visual Studio is used to program the C++ code. First, create a C++ win32 console application. Then download Google protocol buffers C++ and add it as include directory. Next, add ABB EGM source 'egm.pb.cc' to the project. Finally, program the C++ code using Google proto-buf source and instructions from 'egm.pb.cc'.

When programing the C++ code to simulate sensor, there is a simple example for a sequence. First, create a socket and 'listen' on all the interfaces. Secondly, receive and deserialize the message from the robot. Thirdly, create a sensor path correction message. Fourthly, send the path correction message to the robot. Finally, the whole process repeated in a loop until the robot ends the EGM or exits due to connection errors.

When creating a sensor path correction message, many instructions are available in 'egm.pb.cc'. Here only some basic instructions that are used in the experiment are introduced. 'EgmRobot' is used to deserialize the message from the robot or feedback from EGM. After deserializing the message, some information like sequence number, time and message type can be displayed in the console window. Next 'EgmSensorPathCorr' is used to simulate a real sensor. It contains two parts: 'EgmHeader' and 'EgmPathCorr'. In the 'EgmHeader', sequence number, time and message type should be added. In the 'EgmPathCorr', another instruction 'EgmCartesian' is used to set the path correction data. Due to the EGM path correction limitations, only y- and z-directions setting value works. The path correction data in y- and z-direction is the offset value referencing the nominal robot path. The path correction data in x-direction and orientation angle should be set to zero. Then the function 'SerializeToString' is used to

encode and serialize the created path correction message. Finally, these messages will be sent to the robot.

4.2.2 Straight line correction in straight robot path

When creating the correction signal, the correct path only rectifies in y-direction, z-direction correction is 0. This experiment is carried out in three different speeds: 5mm/s, 10mm/s and 20mm/s. The nominal robot path is from p20 to p60. The coordinate value of target p20 is (510, 1245, 235) and target p60 is (170, 1245, 235). The designed correction is also a straight line, but the ending target has the 10.3mm difference with p60 in y-direction. The designed correction path see Figure 8.

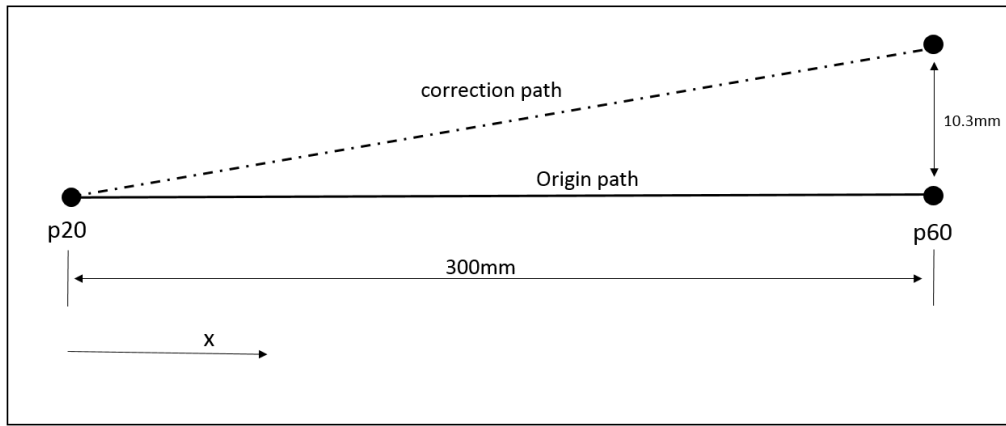


Figure 8 Straight line correction in nominal straight robot path

In the RAPID code, the 'SampleRate' is set to 24ms. 'SampleRate' is input data reading sample rate in EGM path correction and 24ms is the fastest reading sample rate. After finishing some tests, it finds that the sequence number from the robot feedback do not change at the same speed. Sequence number is the feedback from the robot, it reflects how many times the computer sends the path correction message to the robot. For this experiment, when the speed is 5mm/s, the sequence number is 413. It means path correction message is sent 413 times to the robot. Speed 10mm/s is corresponding to 206 and speed 20mm/s is corresponding to 103. Approximately every 145ms or 146ms the robot receives the path correction message. Considering only 1ms difference, sequence number is used to create the path correction message. For example, when the speed is 10 mm/s, correction message in this experiment is $0.05 \times \text{sequence number}$. It means the correction message increment increases 0.05mm in each time. The detailed C++ code is shown in appendix 1.B. Thus the last path correction message has around 10.3mm offset the p60. Since it cannot guarantee the exact sending time, there exists very small errors when creating the path correction message in this experiment.

4.2.3 Fold line correction in straight robot path

The nominal robot path is a straight line, and the path correction is a fold line in this experiment. The correct path only rectifies in y-direction, z-direction correction is 0. Three different speeds are also applied in here. Figure 9 is the simplified diagram for this experiment. The nominal robot path is from p20 to p60. The max offset value of path correction is 2.6mm. Sequence number is also used to create the path correction message.

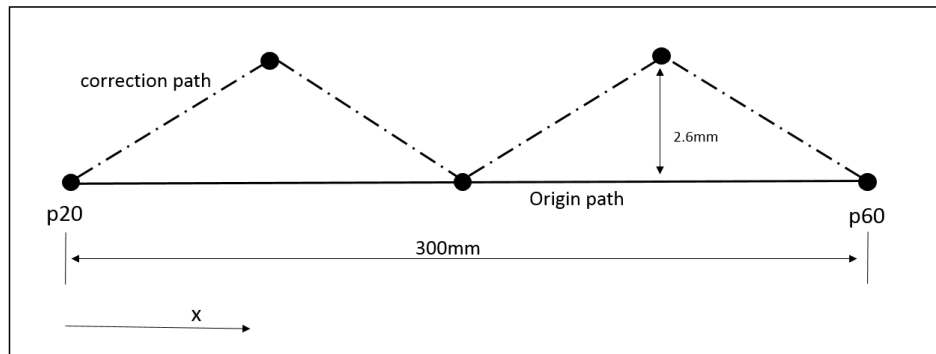


Figure 9 Fold line correction in nominal straight robot path

4.2.4 Curved line correction in straight robot path

To create the curved line correction, two curvatures are used in this experiment. Due to using the sequence number to create the correction messages for both the two curvatures, the difficulty is significantly increased. The nominal robot path is still the straight line. There are two curvatures: one is offset 3mm reference to the nominal path and the other offset 5mm. The correct path only rectifies in y-direction, z -direction correction is 0. The schematic diagram can be seen in Figure 10. Sequence number is still used to create the path correction message. But in the correction message creation there exists an error this is due to the small differences between the control message and real time actions.

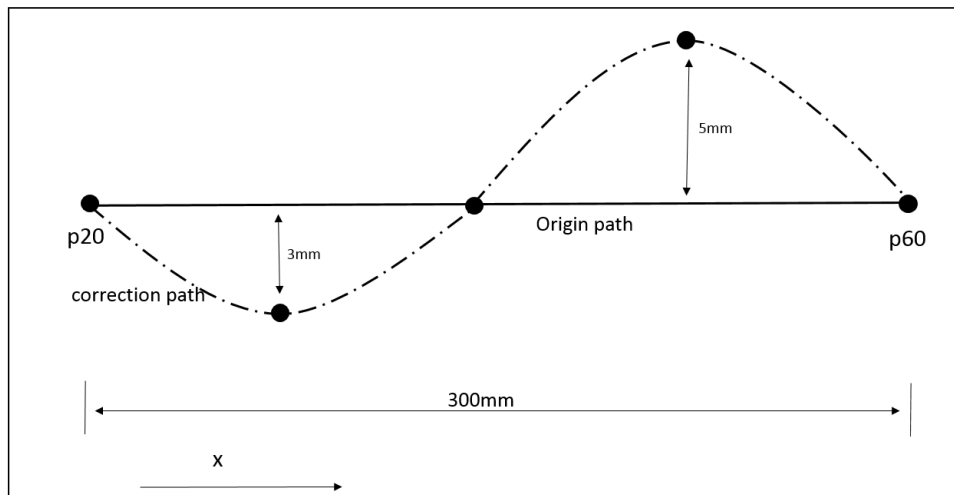


Figure 10 Curved line correction in nominal straight robot path

4.2.5 Curved line correction in curved robot path

The nominal robot path changes to curved line in this experiment. The correction path starts to rectify the robot path in the middle of the welding process. The correct path only rectifies in the y-direction; z-direction correction is 0. Robot path starts from target p20 to p60. RAPID instruction 'EGMMoveC' is used to create the curved line robot path. When using two 'EGMMoveC' instructions, the second instruction cannot work

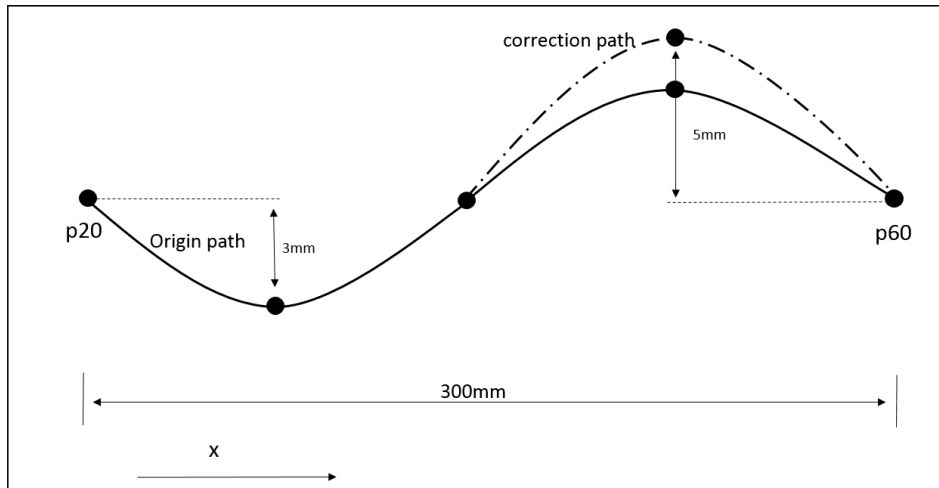


Figure 11 Curved line correction in nominal curved robot path
due to that the UdpUc connection should be rebuilt again. Thus, the first instruction
'EGMMoveC' is replaced by 'MoveC'. Figure 11 shows the designed ideas.

5 Results

5.1 Results of Robot path repeatability test

5.1.1 Part of results of Straight line robot path test

After finishing the straight line robot path test, a comparison of both x-y and x-z are conducted between three different speeds. The results are shown in Figure 12 and Figure 13. As the figures show, the accuracy of robot path repeatability decreases when the speed increases.

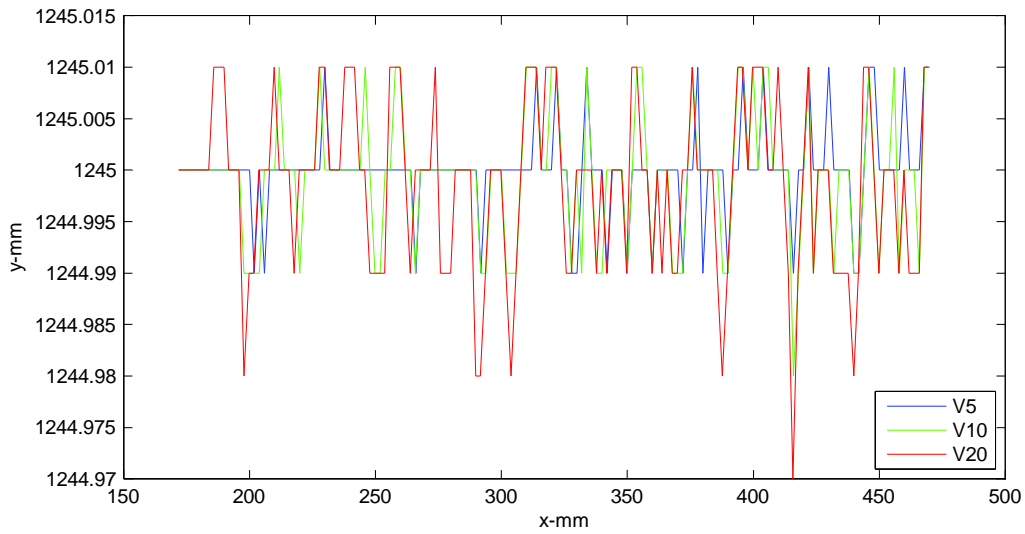


Figure 12 Straight line robot path (x-y graph)

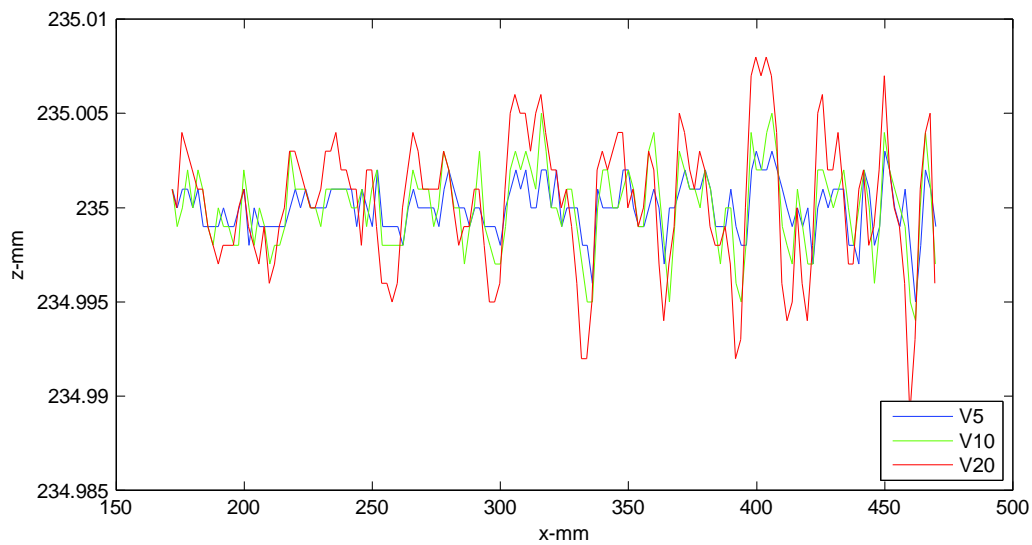


Figure 13 Straight line robot path (x-z graph)

5.1.2 Part of results of Fold line robot path test

Since the differences between the real robot path and designed robot path are very small, it is hard to compare the difference between different speeds in the figure. Here only an example with movements in x- and y-direction (z constant) with speed 10mm/s is shown. As the Figure 14 and Figure 15 show, the max different values in both y and z directions appear in the turning point.

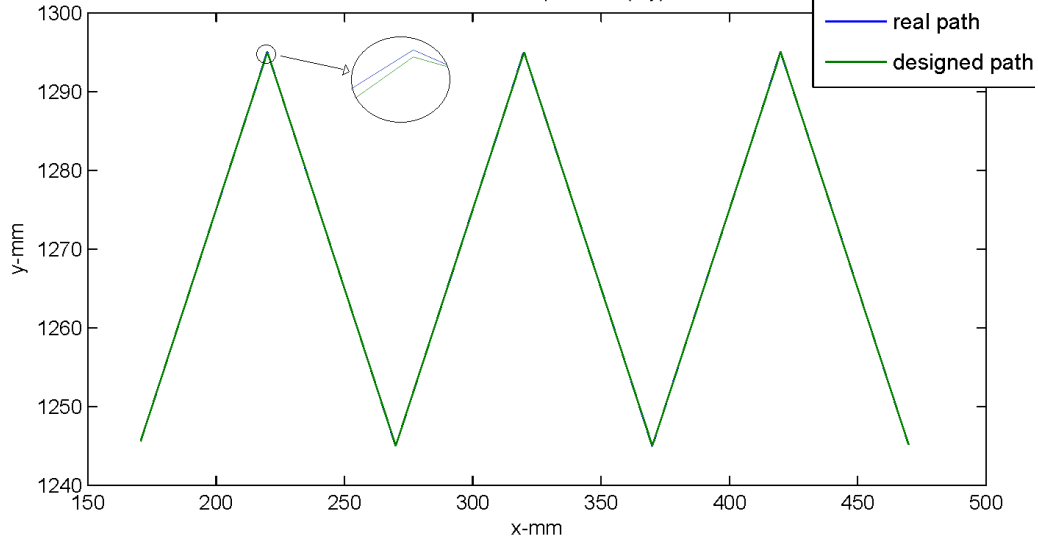


Figure 14 Fold line robot path movement in x- and y-direction with speed 10mm/s (x-y graph)

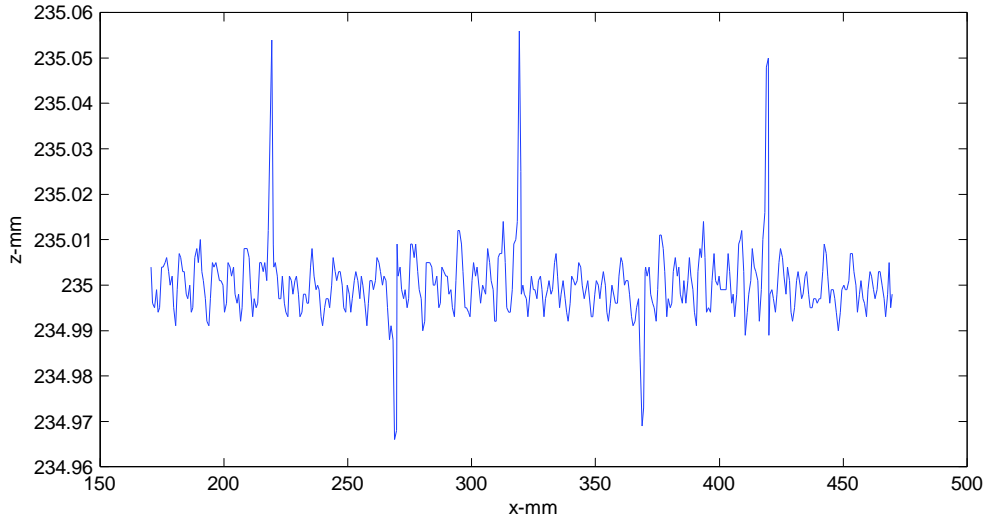


Figure 15 Fold line robot path movement in x- and y-direction with speed 10mm/s (x-z graph)

5.1.3 Part of result of Curved line robot path test

The same problem as with the fold line robot path occurs, it is difficult to show the difference between different speeds in the figure. Thus, here an example where movement in x- and y-direction (z constant) is at speed 10mm/s. Figure 16 shows the x-y graph and Figure 17 shows the x-z graph.

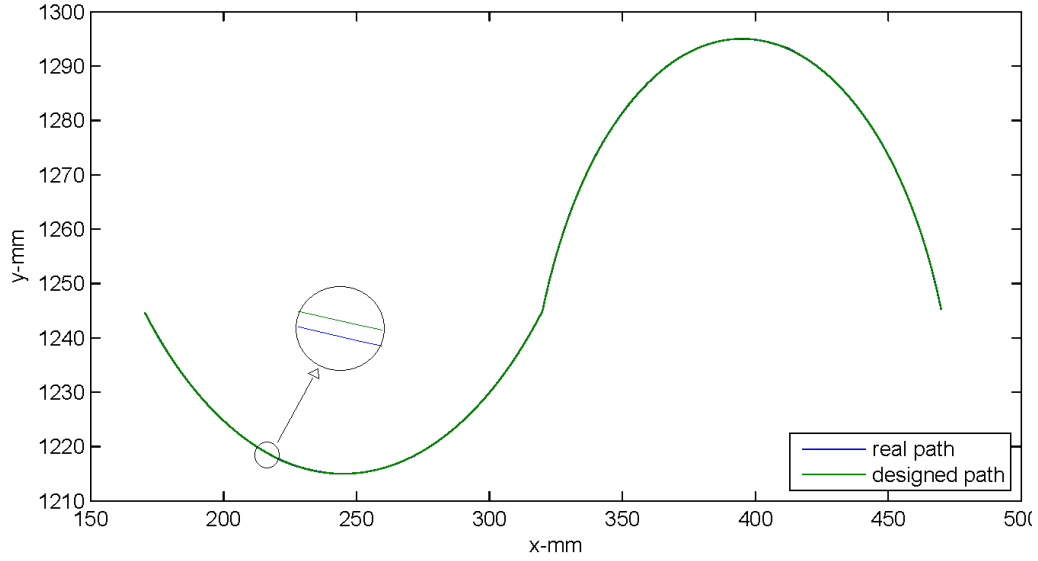


Figure 16 Curved line robot path movement in x and y direction with speed 10mm/s (x-y graph)

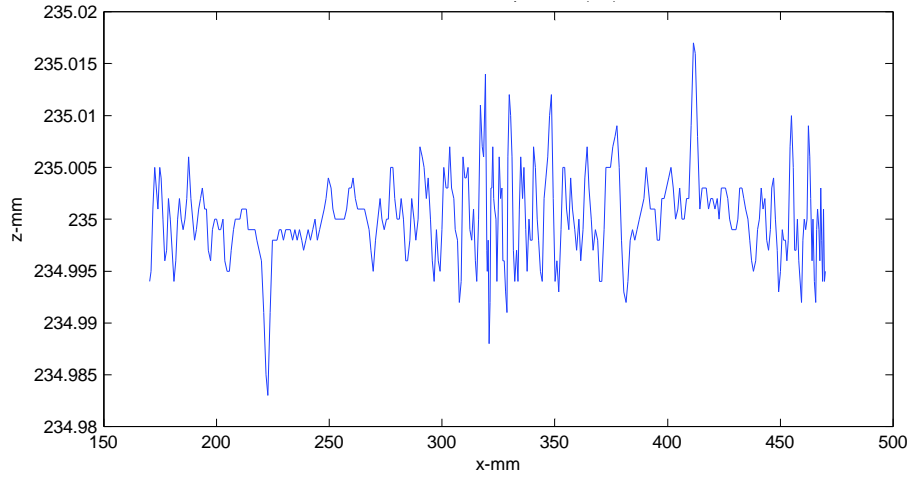


Figure 17 Curved line robot path movement in x and y direction with speed 10mm/s (x-z graph)

5.2 Results of EGM feasibility test

5.2.1 Part of results of Straight line correction test

The experiment is also carried out with three different speeds. The comparison between the three different speeds are shown in Figure 18 and Figure 19. Figure 18 only shows part of the comparison results, since it clearly indicates the difference.

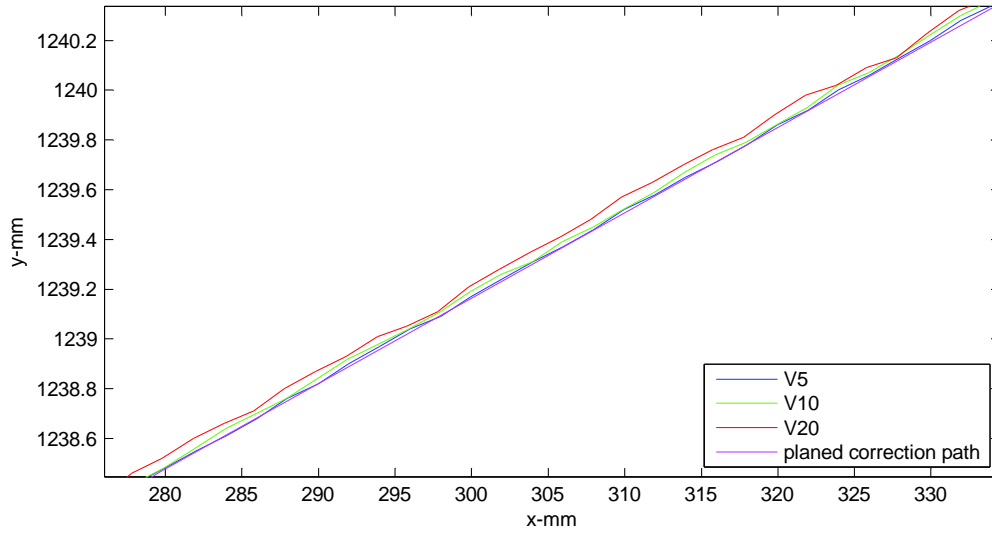


Figure 18 Part of results comparison about straight line correction in nominal straight robot path (x-y graph)

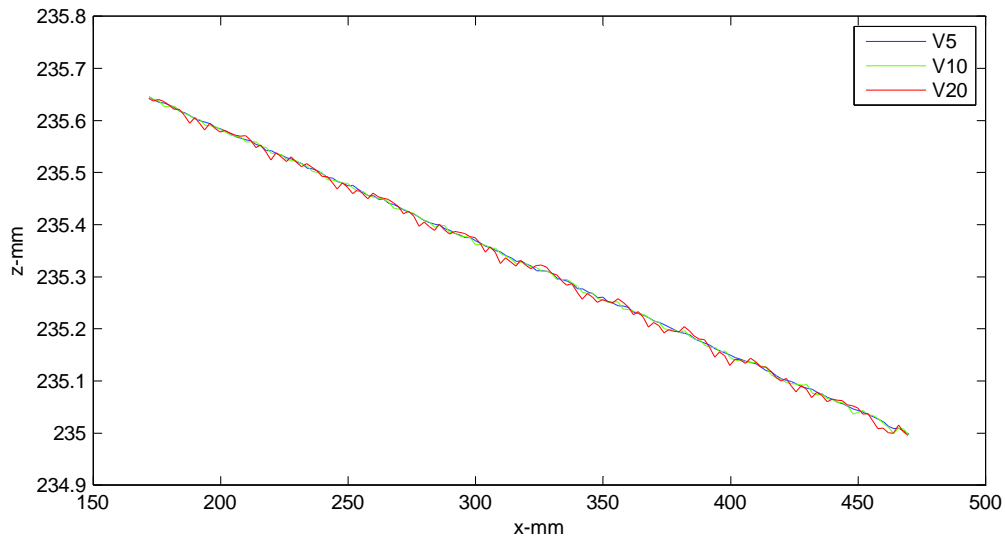


Figure 19 Comparison about straight line correction in nominal straight robot path (x-z graph)

5.2.2 Part of results of Fold line correction

After finishing the experiment, there is a comparison for different speeds. Figure 20 is the result in y-direction and Figure 21 is the result in z-direction.

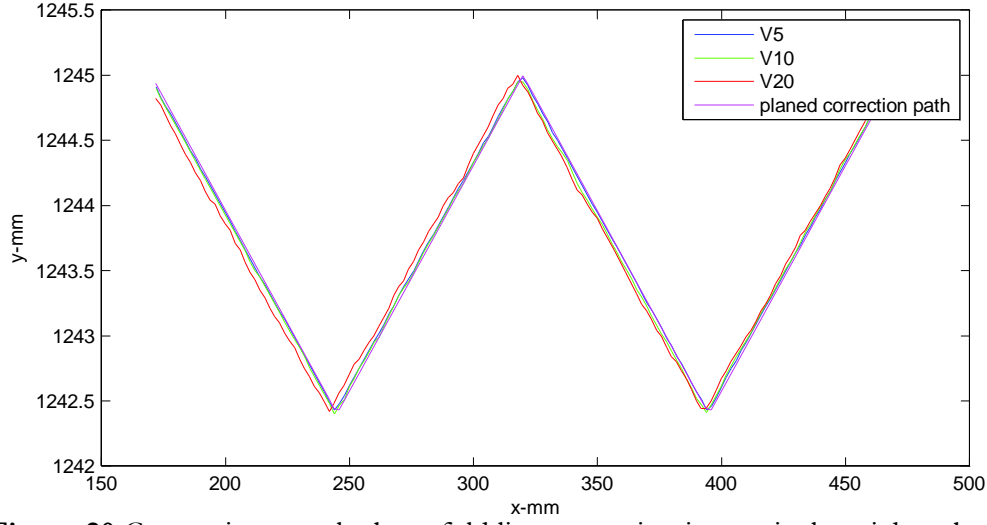


Figure 20 Comparison result about fold line correction in nominal straight robot path (x-y graph)

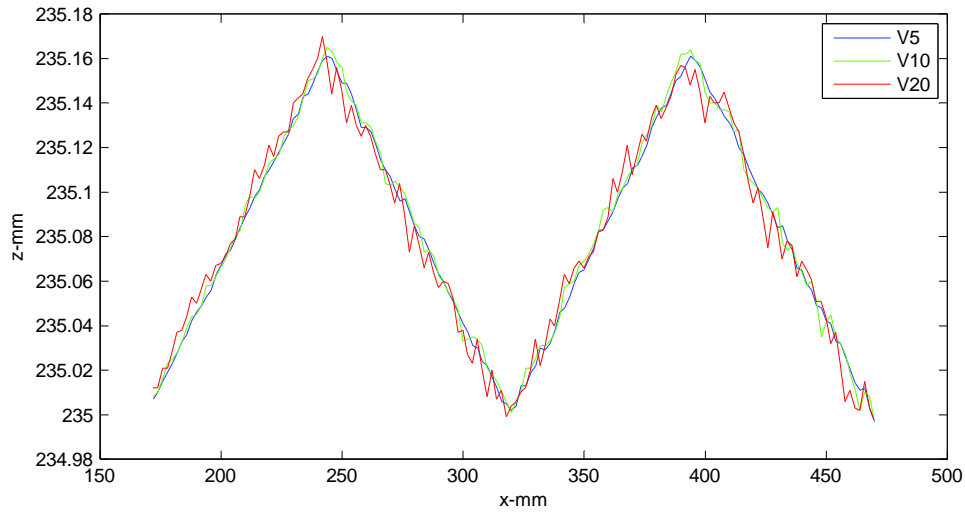


Figure 21 Comparison result about fold line correction in nominal straight robot path (x-z graph)

5.2.3 Part of results of Curved line correction

A comparison is conducted for the different speed. To make the figure clearer, Figure 22 only shows part of the comparison result. The comparison in z-direction is shown in the Figure 23.

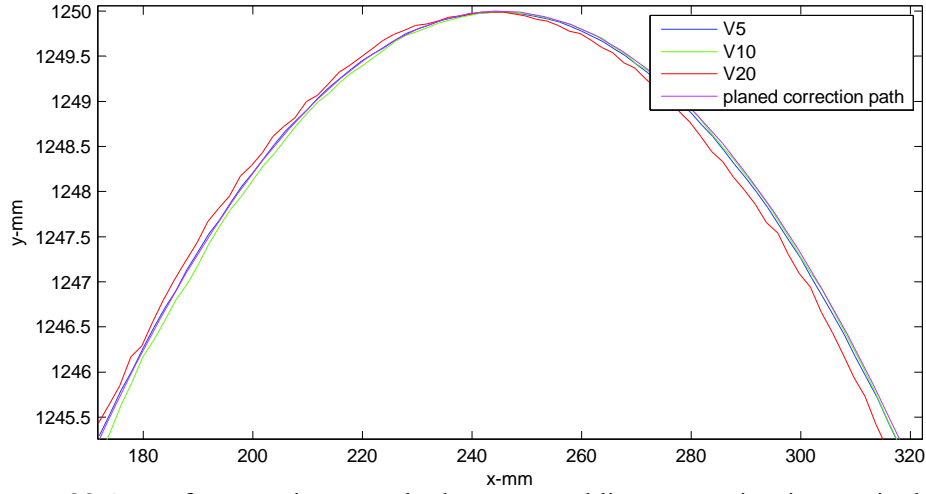


Figure 22 Part of comparison result about curved line correction in nominal straight robot path (x-y graph)

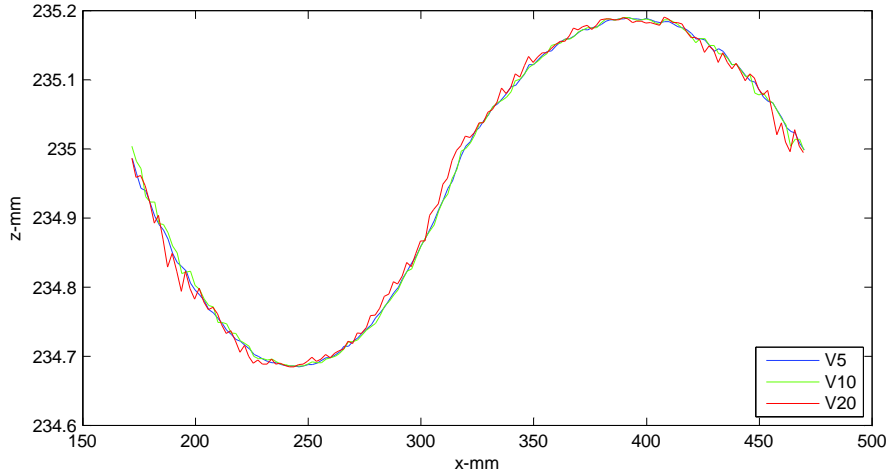


Figure 23 Comparison result about curved line correction in nominal straight robot path (x-z graph)

5.2.4 Part of results of Curved line correction in curved robot path

After finishing the experiment, there is a comparison for the three different speeds. Figure 24 only shows part of the comparison result. The comparison in z direction is seen in Figure 25. During the process from x510 to x320, there is no correction message sent to the robot. The position of the robot does not change. However, when there is a path correction from x320 to x170, the robot position not only changes in the y direction, it also has a small change in the z-direction. The changes in z-direction are opposite to the y-direction changes.

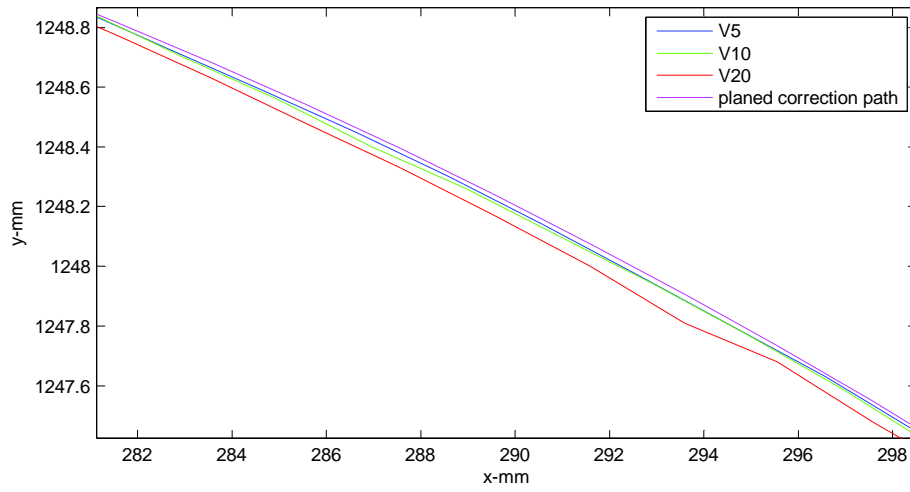


Figure 24 Part of comparison result about curved line correction in nominal curved robot path (x-y graph)

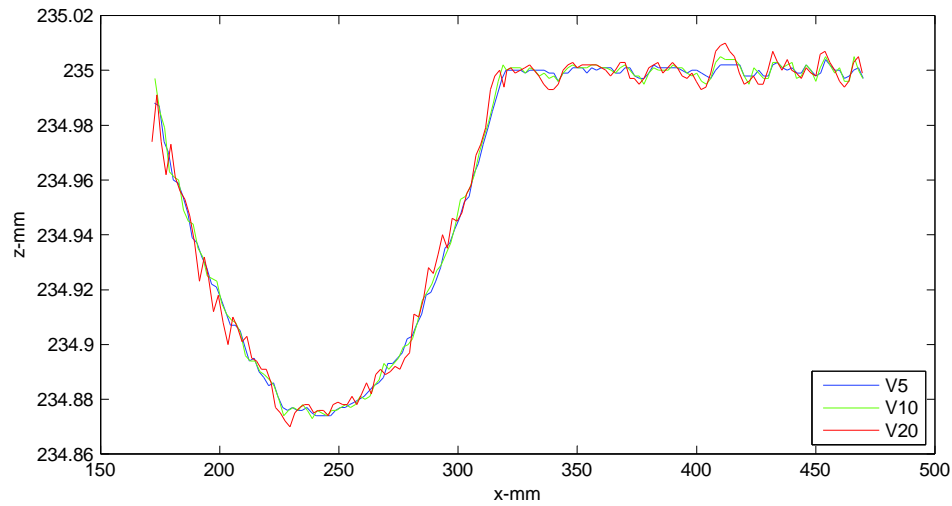


Figure 25 Comparison result about curved line correction in origin curved robot path (x-z graph)

6 Discussion

Many experiments are carried out in this thesis work. Some of the results are already graphed and presented in chapter 5. Here mainly discusses the experimental results.

6.1 Discussion for robot path repeatability test results

For robot path repeatability test, three types of lines are chosen to test how the line shapes influence the robot path repeatability. Each line is carried out with three different speeds in order to get to know how the speed effects for the robot path repeatability.

In each experiment, the max difference value between the real robot path and designed robot path is found in MATLAB. The MATLAB code example can be seen the appendix C. Table 4 shows all of the max difference values between the real robot path and designed robot path. As the table shows, straight line robot path has a very high accuracy. Curved line robot path can also achieve high accuracy. For curved line robot path, the movement in x and z directions(y constant) has a better performance than movement in x and y directions. For the fold line robot path, the results show that the robot has a poor accuracy. As the Figure 14 shows, the max difference value occurs at the turning point of the fold line. This is mainly due to the robot having to sharply decrease the speed and change the movement direction at the turning point. For each type of experiment, as the speed increased, the max difference value also increased. At the same time, the accuracy decreased.

Table 4 Max difference value between the real robot path and designed robot path in the robot repeatability test (Unit: mm)

Type	Straight line robot path	Fold line robot path (z constant)	Fold line robot path (y constant)	3D fold line robot path	Curved line robot path (z constant)	Curved line robot path (y constant)
Speed (mm/s)						
V5	0.0100	0.1170	0.1200	0.1243	0.0630	0.0372
V10	0.0200	0.1350	0.1580	0.1594	0.1021	0.0439
V20	0.0300	0.1710	0.2230	0.2030	0.1392	0.0537

Besides collecting the max difference values, the root mean square error for each experiment was also calculated. The results are shown in Table 5. It represents the standard deviation of the differences between predicted values and observed values. In the experiment, it is used to reflect the robot path repeatability. According to the results, straight line robot path achieves a high repeatability. The worst repeatability is the 3D fold line. Both z constant fold line and y constant fold line are 2D fold lines. Even if the accuracy between 2D fold line and 3D fold line is not much different, 3D fold line has worse repeatability than 2D fold line. Thus, the robot accuracy does not represent or reflect the robot path repeatability. As with the accuracy, the robot path repeatability has decreased with the increasing speed.

Table 5 Root mean square value between the real robot path and designed robot path in the repeatability (Unit: mm)

Type	Straight line robot path	Fold line robot path (z constant)	Fold line robot path (y constant)	3D fold line robot path	Curved line robot path (z constant)	Curved line robot path (y constant)
Speed (mm/s)						
V5	0.0045	0.0154	0.0127	0.0188	0.0110	0.0076
V10	0.0062	0.0215	0.0196	0.0268	0.0172	0.0104
V20	0.0083	0.0327	0.0317	0.0416	0.0263	0.0143

6.2 Discussion for EGM feasibility test results

For the EGM feasibility test, there are two kinds of nominal robot paths. One is straight line robot path, another is curved line robot path. Three types of correction paths are conducted in the nominal straight path. Only curved correction is carried out in the nominal curved path. All of correction paths are created in two dimensions.

The same as the discussion for robot path repeatability test results, the max difference value between the real correction path and planned correction path is calculated in each experiment. The results are shown in Table 6. In EGM feasibility test, the straight line correction path still achieved the high accuracy. In the fold line correction test, the results are not too different compared with other types. Due to the offset value of fold line being small in the test, the effect of the turning point is weak. However, the curved line correction path in straight robot path has poor accuracy. This is mainly because of the limitation of creating the two curvatures correction message. There are two different curvature correction paths in this experiment. It is hard to create the exact correction message when using the sequence number. Thus, two different curvatures and limitation of creating correction message can influence the results of the curved line correction path in straight robot path.

Table 6 Max difference value in y direction between the real correction path and planed robot path in the EGM feasibility test (Unit: mm)

Type	Straight line correction path in straight robot path (z constant)	Fold line correction path in straight robot path (z constant)	Curved line correction path in straight robot path (z constant)	Curved line correction in curved robot path (z constant)
Speed (mm/s)				
V5	0.0278	0.0499	0.1007	0.0408
V10	0.0483	0.0773	0.1905	0.0647
V20	0.0877	0.1442	0.3920	0.1172

The Root mean square error is presented in Table 7. Here Root mean square error is also used to reflect the robot path repeatability. The performance of accuracy and repeatability for the robot path decreased when the speed increased. From the Table 7, there are two values larger than the other values. They are fold line and curved line correction path in straight robot path with speed 20mm/s. For the same correction path, the correction message should be enlarged in each sending cycle when the speed

increases. Both the large changes and high speed can cause poor robot accuracy and unreliable repeatability during the movement process. It can also lead to making the laser tool shake. This should not occur during the welding process. Therefore, choosing the proper welding speed is very important when using the EGM path correction.

Table 7 Root mean square value in y direction between the real robot correction and planed robot path in the EGM feasibility test (Unit: mm)

Type	Straight line correction path in straight robot path (z constant)	Fold line correction path in straight robot path (z constant)	Curved line correction path in straight robot path (z constant)	Curved line correction in curved robot path (z constant)
Speed (mm/s)				
V5	0.0102	0.0246	0.0332	0.0116
V10	0.0234	0.0394	0.0493	0.0145
V20	0.0527	0.0924	0.1337	0.0340

There exists small changes in z-direction besides the path correction in y-direction, even if there is no correction message for z-direction during the experiment. In this case, the robot should keep constant in z-direction. As the Figure 20 and Figure 21 shows, the changes in z-direction is opposite to the path correction in y-direction. For example, when the correction message rectifies the path in $-y$ direction, the robot is corresponding to modify in $+z$ direction. However, the real reason is not investigated in this thesis project. Here two assumptions may cause the movement in z-direction. One of the reasons is the limitations of EGM modules, which has drawbacks. Another reason is the existence of some errors when setting and using the EGM path correction. To solve this problem in the future, path correction messages in z-direction should be sent to the robot to compensate the changes. Due to fibre laser tool which is used in this thesis work having a working tolerance in z-direction and the deviation in z-direction being very small, the deviation does not influence the welding performance.

6.3 Speed analysis for EGM feasibility test

Speed analysis for all the experiments was carried out to analyse how the speed changes when using the EGM Path Correction. The same speeds in the same type of robot path between robot path repeatability test and EGM feasibility test were compared. After the comparison, there was no difference between two kinds of tests for the same speed. An example is shown in Figure 26 and Figure 27. Figure 26 is the speed variation without correction in straight line robot path repeatability test, and Figure 27 shows the speed variation with straight line EGM path correction. From the graphs, the tolerances of the speed variation between them are almost the same. Thus, using the EGM path correction would not influence the speed variation during the welding process.

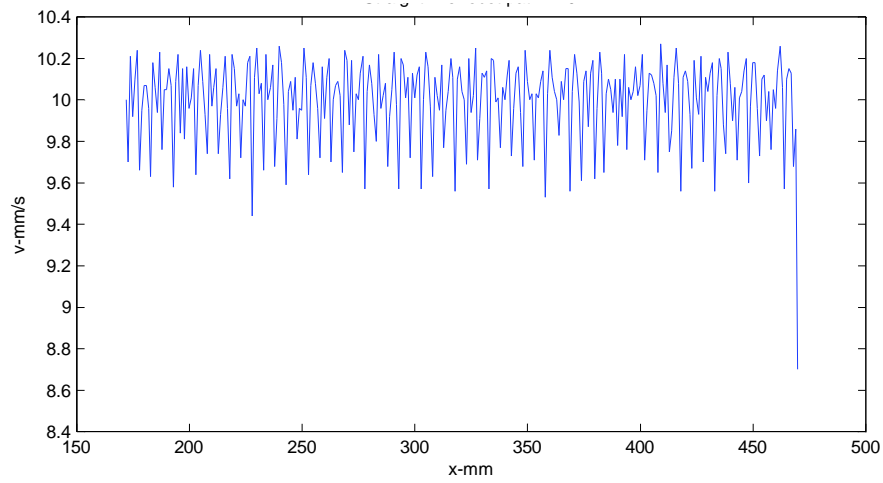


Figure 26 Speed variation in path repeatability test of straight line robot path with speed 10mm/s

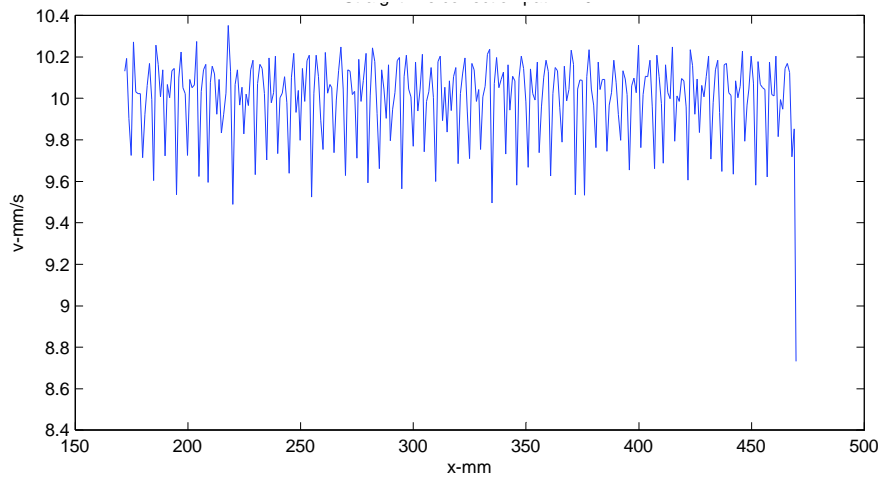


Figure 27 Speed variation in EGM feasibility test of straight line correction path with speed 10mm/s

7 Conclusion

The literature review shows that, there are many methods proposed to control the robot motion in real-time. However, when considering the requirements on accuracy for laser welding and considering the industrial feasibility, it is impossible to apply them in practical production. Thus, some commercial solutions are investigated to show the feasibility in the laser welding application.

ABB EGM in the Robotware control system module has been evaluated for robot motion control in laser welding application. Three types of experiments were carried out to test the industrial robot repeatability. Four kinds of experiments were proposed and conducted to test the feasibility of EGM path correction module. In each experiment, three different speeds were conducted to investigate how the speed influences the accuracy. The experimental results show that, the EGM path correction can achieve good accuracy and repeatability. For example, the laser tool which was used in the experiment, has a laser spot diameter of 1.12mm. The whole error during the welding process should be less than half of the diameter which is 0.56mm. Considering other errors like fixture repeatability, the test results can fulfil the laser welding requirements, except the test of curved line correction path in straight robot path. The deviation in z-direction should be investigated and eliminated in the future. Furthermore, EGM path correction would not influence the welding speed during the welding process.

From all the experiment results, the EGM path correction is feasible to be used in robot motion control of laser welding.

7.1 Future Work and Research

This thesis work only produced a basic test for EGM path correction. The results show that it is possible to apply the system in laser welding. However, there exists deviation in the z-direction during the path correction process. Future work will try to investigate the cause of the deviation in z-direction and try to find some methods to eliminate it.

In this thesis project, only different types of correction lines and speeds are investigated, in order to know how they affect the robot accuracy and repeatability. Other affected parameters should be considered when doing the experiment in the future. What's more, the real welding test should be conducted to test how they influence the welding performance.

Due to the thesis project limitation, only UdpUc protocol is used in the experiment and the correction message is created by a computer. The correction messages from the real sensor should be sent to the robot through analogue input/output in future experiments. This is closer to the practical production procedures.

7.2 Generalization of the result

The test results showed how different types of correction lines and speeds influence the robot accuracy and repeatability. These results proved that it is possible to apply the EGM module for robot motion control in laser welding. It can achieve high accuracy.

This thesis project provides basic knowledge about how to operate the EGM path correction. It not only can be applied for laser welding, but also for other applications such as tracking objects.

8 References

- [1] "RobIn – Robust in-process joint finding," Produktion2030, [Online]. Available: <http://www.produktion2030.se/en/projects/robin-robust-in-process-joint-finding/>. [Accessed 27 06 2016].
- [2] S. Tsukamoto, "Laser Welding," *Welding International*, pp. 767-774, 2003.
- [3] S. K. R. P. Boris Regaard, "Seam-tracking for high precision laser welding applications—Methods, restrictions and enhanced concepts," *Journal of Laser Applications*, vol. 21, nr 4, 2009.
- [4] Y. X. P. W. M. L. Yu Huang, "A seam-tracking laser welding platform with 3D and 2D visual information fusion vision sensor system," *The International Journal of Advanced Manufacturing Technology*, pp. 415-426, 2013.
- [5] X. C. Hong Luo, "Laser visual sensing for seam tracking in robotic arc welding of titanium alloys," *The International Journal of Advanced Manufacturing Technology*, pp. 1012-1017, 2005.
- [6] B. G. Blomdell Anders, "Extending an Industrial Robot Controller-Implementation and Applications of a Fast Open Sensor Interface," *IEEE Robotics & Automation Magazine*, pp. 85-94, 2005.
- [7] W. F. L. Diego Caberlon Santini, "AN OPEN CONTROL SYSTEM FOR MANIPULATOR ROBOTS," i *20th International Congress of Mechanical Engineering*, Gramado, RS, Brazil, 2009.
- [8] H. Y. J. Z. T. L. S. C. Yanling Xu, "Real-time seam tracking control technology during welding robot GTAW process based on passive vision sensor," *Journal of Materials Processing Technology*, pp. 1654-1662, 2012.
- [9] I. D. K. N. A. R. Anders Blomdell, "Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers," *IEEE International Conference on Robotics and Automation*, pp. 62-66, 2010.
- [10] R. A. B. J. J. M. Menno de Graaf, "Real-time seam tracking for robot laser welding using trajectory-based control," *Control Engineering Practice*, pp. 944-953, 2010.
- [11] M. T. X. Z. Z. T. De Xu, "Seam Tracking and Visual Control for Robotic Arc Welding Based on Structured Light Stereovision," *International Journal of Automation and Computing*, pp. 63-75, 2004.
- [12] S. M. H. E. M. J. Kah P, "Robotic arc welding sensors and programming in industrial applications," *International Journal of Mechanical and Materials Engineering*, vol. 10, pp. 1-16, 2015.
- [13] "Permanova WT04 Laser Welding Tool System for All Joining Applications," Permanova Lasersystem AB, [Online]. Available: <http://permanova.se/?id=469>. [Accessed 31 03 2016].
- [14] "Latest robot controller software from ABB combines flexibility and productivity for developers," ABB, 18 12 2014. [Online]. Available:

- <http://www.abb.com/cawp/seitp202/2a21e772c71016d1c1257daf004046bc.aspx>.
[Accessed 12 03 2016].
- [15] "WeldMaster Plattform," Precitec Group, [Online]. Available:
<http://www.precitec.de/en/products/joining-technology/process-monitoring/weldmaster-plattform/>. [Accessed 31 03 2016].
- [16] Application manual Controller software IRC5, Västerås Sweden: ABB AB Robotics Products, 2015.
- [17] "Protocol Buffers," Google developer, [Online]. Available:
<https://developers.google.com/protocol-buffers/>. [Använd 15 06 2016].
- [18] Technical reference manual System parameters, Västerås Sweden: ABB AB Robotics Products, 2016.
- [19] Technical reference manual RAPID Instructions, Functions and Data types, Västerås Sweden: ABB AB Robotics Products, 2016.
- [20] N. Xu, H. S. Huang and Y. K. Rong, "Automatic setup planning: current state-of-the-art and future perspective," *International Journal of Manufacturing Technology and Managment*, vol. 11, no. 2, pp. 193-208, 2007.
- [21] R. Stryczek, "A Hybrid Approach for Manufacturability Analysis," *Advances in Manufacturing Science and Technology*, vol. 35, nr 3, pp. 55-70, 2011.
- [22] International Standard, "IEC 61131 Programmable controller - Part 3: Programming Languages," *IEC*, nr 2nd ed., 2003.

A. RAPID code in IRB 4400 industrial robot for EGM test

```
MODULE TestEGM
VAR egmident egmID2;
VAR egmstate egmSt2;
!Satey position
CONST      robtarget      p10:=[[510.00,1245.00,235.00],[0.0248717,0.951187,-
0.306566,0.0253196],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST      robtarget      p21:=[[510.00,1245.00,255.00],[0.0248717,0.951187,-
0.306566,0.0253196],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
!Start position
CONST      robtarget      p20:=[[470.00,1245.00,235.00],[0.0248717,0.951187,-
0.306566,0.0253196],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
!destination position
CONST      robtarget      p60:=[[170.00,1245.00,235.00],[0.0248717,0.951187,-
0.306566,0.0253196],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST      robtarget      p70:=[[150.00,1245.00,235.00],[0.0248717,0.951187,-
0.306566,0.0253196],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST      robtarget      p80:=[[150.00,1245.00,280.00],[0.0248717,0.951187,-
0.306566,0.0253196],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
! Correction frame offset: none
CONST pose sensor_frame:=[[0,0,0],[1,0,0,0]];
! Used tool and workobject
!PERS tooldata tLaser:=[TRUE,[[0,0,900],[1,0,0,0]],[60,[170,0,220],[1,0,0,0],0,0,0]];
VAR intnum feeder_low:=0;
VAR iodev logfile;
VAR pos pos1:=[0,0,0];

PROC main()
! Move to start position. Fine point is demanded.
MoveJ p21, v50, fine, tLaser\WObj:=wobj0;
MoveL p10, v20, fine, tLaser\WObj:=wobj0;
EGM_UDPUC_Pth_1;
ENDPROC
PROC EGM_UDPUC_Pth_1()
EGMReset egmID2;
EGMGetId egmID2;
egmSt2:=EGMGetState(egmID2);
TPWrite "EGM state: "\Num:=egmSt2;
IF egmSt2 <= EGM_STATE_CONNECTED THEN
! Set up the EGM data source: UdpUc server using device "EGMsensor:"and con-
figuration "pathCorr"
EGMSetupUC ROB_1, egmID2, "pathCorr", "EgmSensor:"\PathCorr\APTR;
ENDIF
```

```
EGMActMove egmId2, sensor_frame\SampleRate:=24;
! Move to the start point. Fine point is demanded.
MoveL p20, v10, fine, tLaser\WObj:=wobj0;
!open a file and connect a trap routine to save robot TCP data every 0.1s
Open usbdisk1\File:="coordLog.txt", logfile\Write;
CONNECT feeder_low WITH Savedata;
ITimer 0.1, feeder_low;
! movements with path corrections.
EGMMoveL egmId2, p60, v10, fine, tLaser\WObj:=wobj0;
! close the file and Move to a safe position after path correction.
IDelete feeder_low;
Close logfile;
MoveL p70,v10,fine,tLaser\WObj:=wobj0;
MoveJ p80,v20,fine,tLaser\WObj:=wobj0;

! Release the EGM identity for reuse.
egmSt2:=EGMGetState(egmID2);
IF egmSt2 = EGM_STATE_CONNECTED THEN
    TPWrite "Reset EGM instance egmID2";
    EGMReset egmID2;
ENDIF
ENDPROC

TRAP Savedata
    pos1 := CPos(\Tool:=tLaser);
    Write logfile, ""\Pos:=pos1;
    !Write logfile, "\0A";
ENDTRAP

ENDMODULE
```

B. C++ code for creating the path correction message

```
#include "stdafx.h"
#include <WinSock2.h>
#include <iostream>
#include <fstream>
#include "egm.pb.h" // generated by Google protoc.exe
#pragma comment(lib, "Ws2_32.lib") // socket lib
#pragma comment(lib, "libprotobuf.lib") // protobuf lib
static int portNumber = 6510;
static unsigned int sequenceNumber = 0;
using namespace std;
using namespace abb::egm;
////////////////////////////////////
// Create a simple sensor path corr message
void CreateSensorPathCorrMessage(EgmSensorPathCorr* pSensorPathCorrMessage)
{
    EgmHeader* header = pSensorPathCorrMessage->mutable_header();
    header->set_mtype(EgmHeader_MessageType_MSGTYPE_PATH_COR-
RECTION);
    header->set_seqno(sequenceNumber++);
    header->set_tm(GetTickCount());
    double correction = 0.05*sequenceNumber;
    EgmPathCorr *pathcorr = pSensorPathCorrMessage->mutable_pathcorr();
    EgmCartesian *pc = pathcorr->mutable_pos();

    pc->set_x(0.0);
    pc->set_y(correction);
    pc->set_z(0.0);
    /*if (sequenceNumber < 20)
    {
        pc->set_z(0.0);
    }
    else
    {
        pc->set_z(0.0);
    }*/

    pathcorr->set_age(0);
}
// Display inbound robot message
void DisplayRobotMessage(EgmRobot *pRobotMessage)
{

```



```
        if (pRobotMessage->has_header() && pRobotMessage->header().has_seqno() && pRobotMessage->header().has_tm() && pRobotMessage->header().has_mtype())
        {
            printf("SeqNo=%d Tm=%u Type=%d\n", pRobotMessage->header().seqno(), pRobotMessage->header().tm(), pRobotMessage->header().mtype());

            if (pRobotMessage->header().seqno() == 0)
            {
                sequenceNumber = 0;
            }
        }
        else
        {
            printf("No header\n");
        }
    }

#define _TCHAR char
int main(int argc, _TCHAR* argv[])
{
    SOCKET sockfd;
    int n;
    struct sockaddr_in serverAddr, clientAddr;
    int len;
    char protoMessage[1400];

    /* Init winsock */
    WSADATA wsaData;
    if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0)
    {
        fprintf(stderr, "Could not open Windows connection.\n");
        exit(0);
    }
    // create socket to listen on
    sockfd = ::socket(AF_INET, SOCK_DGRAM, 0);
    memset(&serverAddr, sizeof(serverAddr), 0);
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serverAddr.sin_port = htons(portNumber);
    // listen on all interfaces
    bind(sockfd, (struct sockaddr *)&serverAddr, sizeof(serverAddr));
    string messageBuffer;
    while (true)
    {
        // receive and display message from robot
        len = sizeof(clientAddr);
        n = recvfrom(sockfd, protoMessage, 1400, 0, (struct sockaddr *)&clientAddr, &len);
        if (n < 0)
```

```
        {
            printf("Error receive message\n");
            continue;
        }
        // deserialize inbound message
        EgmRobot *pRobotMessage = new EgmRobot();
        pRobotMessage->ParseFromArray(protoMessage, n);
        DisplayRobotMessage(pRobotMessage);
        delete pRobotMessage;
        // create and send a sensor path corr message
        EgmSensorPathCorr *pSensorPathCorrMessage = new
EgmSensorPathCorr();
        CreateSensorPathCorrMessage(pSensorPathCorrMessage);
        pSensorPathCorrMessage->SerializeToString(&mes-
sageBuffer);

        // send a message to the robot
        n = sendto(sockfd, messageBuffer.c_str(), mes-
sageBuffer.length(), 0, (struct sockaddr *)&clientAddr, sizeof(clientAddr));
        if (n < 0)
        {
            printf("Error send message\n");
        }
        delete pSensorPathCorrMessage;
    }
    return 0;
}
```

C. MATLAB code example to analysis data

```
%Load the data
x=coordLog(:,1);
y=coordLog(:,2);
z=coordLog(:,3);
y1=y;
n=size(x,1);
%Get planed path correction value
for i=1:n
    y1(i)=foldlineequ(x(i));
end
plot(x,y,x,y1);
% compare and get the difference value
y2=abs(y-y1);
maxv=max(y2)% get max difference value
mse1=sum(y2.^2)/n %calculate mean square error
rmse1=sqrt(mse1) %calculate root mean square error(standard deviation)

%this function is for simulating planed path correction fold line
function y=foldlineequ(x)
if x>395
    y=1242.4+(x-395)*(2.6/75);
elseif x>320
    y=1245-(x-320)*(2.6/75);
elseif x>245
    y=1242.4+(x-245)*(2.6/75);
else
    y=1245-(x-170)*(2.6/75);
end
```