SENSOR-GUIDED ROBOTIC LASER WELDING

Menno de Graaf

This research was financially supported by the Dutch Technology Foundation STW (TWO.5927).

Sensor-guided Robotic Laser Welding

Printed by PrintPartners Ipskamp, Enschede

ISBN 978-90-365-2564-0

SENSOR-GUIDED ROBOTIC LASER WELDING

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. W.H.M. Zijm,
volgens het besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 6 december 2007 om 15.00 uur

door

Menno Wouter de Graaf

geboren op 22 februari 1978
te Bunschoten

Dit proefschrift is goedgekeurd door

prof. dr. ir. J. Meijer, promotor
prof. dr. ir. J.B. Jonker, promotor
dr. ir. R.G.K.M. Aarts, assistent-promotor

Het is de HEER die wijsheid schenkt,
Zijn woorden bieden kennis en inzicht.

(Spreuken 2:6, NBV)

Voorwoord

Na ruim vijf jaar is het mooi geweest! Het resultaat van al dit zwoegen (het boekje) ligt voor u. Ik heb een prachtige tijd achter de rug bij de vakgroep Werktuigbouwkundige Automatisering. Met enkele woorden probeer ik die voor u samen te vatten: onderzoeken, simuleren, experimenteren, papers schrijven, programmeren, koffie drinken, laserlassen, treinreizen, conferenties bezoeken (Aken, San Fransisco, Miami, Bolonga, Athene), afstudeerders begeleiden, haalbaarheid aantonen, etc. Dit promotiewerk was niet mogelijk geweest zonder de hulp van een groot aantal mensen.

Een achttal afstudeerders heeft een behoorlijke bijdrage geleverd en ik wil ze dan ook van harte bedanken: Tino, Jeroen vT, Thijs, André, Hiddo D, Frank vdV, Maarten en Arjan.

De goede sfeer in de vakgroep maakte dat ik met veel plezier naar mijn ’werk’ ging. Daarvoor wil ik alle collega’s bedanken: Ben, Benno, Bertus, Dannis, Dimitrios, Frank P, Frits, Gert, Hans-Martin, Hiddo S, Jan, Jeroen OB, Jeroen vT, Jilles, Johan, Johannes, Jonathan, Leo, Marco, Martina, Max, Pathiraj, Rob, Ronald, Sjoerd, Tjeerd, Toon, Tyrone en Wouter.

Mijn dank voor de dagelijkse begeleiding gaat uit naar Ben, Johan en Ronald. We hebben vele inhoudelijke sessies gehad waarin dit proefschrift werd besproken. Die waren soms zwaar en vermoeiend, maar het is de kwaliteit zeker ten goede gekomen.

Een promotieonderzoek is eigenlijk niet goed uit te voeren zonder je gedachten af en toe te verzetten. Het thuisfront (Jiska, Jefta en Imma) was hierbij dan ook onmisbaar, bedankt!

Menno de Graaf
Amersfoort, oktober 2007

Contents

Nomenclature

Latin symbols

Symbol	*Description*	*Unit*
a	Polynomial coefficient	[-]
$\mathbf{A}$	Transformation matrix, representing a given movement of the robot flange frame	
b	Polynomial coefficient	[-]
$\mathbf{B}$	Transformation matrix, representing a measured movement of the tool frame	
c	Polynomial coefficient	[-]
c_x	Scaling factor	[pixels/mm]
c_y	Scaling factor	[pixels/mm]
d	Polynomial coefficient	[-]
d	Focal distance between pilot frame and Nd:YAG frame	[mm]
f	Focal length	[mm]
i	Sensor measurement index	
j	Numerical iteration index	
$\mathbf{J}$	Intersection vector between sensor diode plane and workpiece plane	
$\mathbf{K}$	Cubic interpolation function for the seam trajectory	
K_y	Smoothing factor for the tractrix teaching algorithm	
K_z	Smoothing factor for the tractrix teaching algorithm	
K_γ	Smoothing factor for the polyfit teaching algorithm	
k	Robot measurement index	
l	Look-ahead distance between laser tool and sensor tool	
n	Segment index, polynomial order	
$\mathbf{n}$	Surface normal	
p	Teach step index	
$\mathbf{P}$	Position vector	

Latin symbols

Symbol	*Description*	*Unit*
$\mathbf{q}_d$	Desired robot joint position	
$\dot{\mathbf{q}}_d$	Desired robot joint velocity	
$\mathbf{q}_m$	Measured robot joint position	
$\mathbf{Q}$	Quaternion	
$\mathbf{R}$	Rotation matrix [3x3]	
$\mathbf{r}_x$	First column of a rotation matrix representing the x-axis	
$\mathbf{r}_y$	First column of a rotation matrix representing the y-axis	
$\mathbf{r}_z$	First column of a rotation matrix representing the z-axis	
$\mathbf{rot}()$	Rotation matrix corresponding to a pure rotation around one of the principal axes	
$\mathbf{s}$	Sensor measurements	
s_x	Sensor x-measurement	[mm]
s_y	Sensor y-measurement	[mm]
s_z	Sensor z-measurement	[mm]
s_{rx}	Sensor rx-measurement	[rad, deg]
S	Measured spot size	[pixels]
t	Time	[s]
T_s	Sample time of sensor	[s]
T_r	Sample time of robot	[s]
$\mathbf{T}$	Homogenous transformation matrix [4x4]	
$\mathbf{u}$	Normalised vector of rotation	
w	Angular velocity	[rad/s, deg/s]
$\mathbf{X}$	Transformation matrix, representing a transformation between the robot flange frame and the tool frame	

Greek symbols

Symbol	*Description*	*Unit*
α	Diode angle	[rad, deg]
β	Rotation angle around the y-axis	[rad, deg]
γ	Rotation angle around the x-axis	[rad, deg]
φ	Rotation angle around the z-axis	[rad, deg]
θ	Rotation angle	[rad, deg]
ϕ	Rotation angle	[rad, deg]
λ	Interpolation parameter	

Coordinate frames

Symbol	*Description*
B	Base frame
C	Coaxial camera frame
D	Sensor's laser diode frame
E	Sensor tool calibration object frame
F	Generic frame (**B**, **G**, **M** or **G**)
G	Seam frame
H	Product frame
L	Laser tool frame
M	Station frame
N	Flange frame
O	Object frame
P	Pilot laser frame
R	Laser tool calibration object frame
S	Sensor tool frame
T	Generic tool frame (**L** or **S**)

Abbreviations

Abbreviation	*Description*
API	Application Programming Interface
CAD	Computer Aided Design
CCD	Charge Coupled Device
CW	Continuous Wave
CMOS	Complementary Metal-Oxide-Semiconductor
CPU	Central Processing Unit
DOF	Degree-of-freedom
DGM	Direct Geometric Model
IGM	Inverse Geometric Model
ILC	Iterative Learning Control
I/O	Input/Output
GMAW	Gas Metal Arc Welding
GUI	Graphical User Interface
LLI	Low Level Interface
MSE	Mean Squared Error
RTRCAL	Real-time Robot Controller Abstraction Layer
SME	Small and Medium Enterprises
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 Background

Laser welding is a relatively new joining technique that has been adopted by the automotive industry for welding of large product series and also in small and medium enterprises (SME's) lasers are increasingly used. In the current car production, spot welds are more and more being replaced by laser welds due to faster welding and therefore a decreased cycle time.

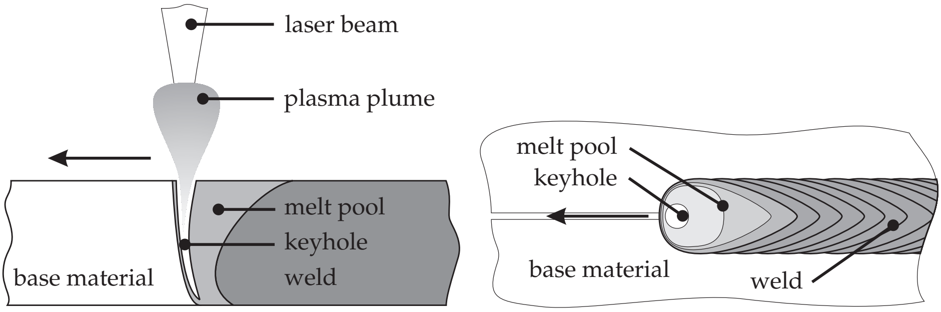
For the large product series in the automotive industry, the work preparation time between the series is rather short compared to the actual production time. In contrast, for small product series, it is a time-consuming task to accurately program the welding trajectory. This hampers a cost efficient application of the laser welding technique for such small product series or prototypes. To solve this problem, a high level of automation is needed in the work preparation.

This thesis considers the latter aspect for continuous seam welding of 3D seams. In particular, the requirements for the positioning accuracy of the laser beam on the seam and the automation of the process will be fulfilled using seam tracking sensors.

1.1.1 Laser welding

The principle of laser welding is to move a focussed laser beam over a seam with a prescribed velocity. The laser beam is focussed on the material using lenses and mirrors assembled in a laser welding head. The laser radiation is absorbed by the metal and due to the high energy content, a melt pool is formed. Due to the liquid flow in the melt pool both metals are mixed and after solidification the metals are joined.

In this work, continuous wave (CW) laser welding is considered. Two different laser welding modes are distinguished: conduction welding and keyhole welding. In conduction welding the metal is melted at the surface and the heat is transported into the material by heat conduction. In this way, shallow weld seams are formed. In figure 1.1, the principle of keyhole laser welding is illustrated. In keyhole laser welding, the intensity of the laser radiation is sufficiently high to evaporate the molten metal at the focal area of the laser beam. The recoil pressure of the metal vapour pushes the molten metal aside, creating a capillary known as the keyhole. The keyhole is filled with metal plasma and can extend over the full thickness of the material. Figure 1.1(a) shows that the plasma escapes from the keyhole, forming a plasma plume above the material. The weld seam is usually protected from oxidation by means of a shielding gas.



(a) Longitudinal section.

(b) Top view.

Figure 1.1: Illustration of the principle of keyhole laser welding.

The advantage of (keyhole) laser welding over conventional welding is the high welding speed that can be achieved while maintaining the full penetration depth. Another advantage is the ability to make overlap welds, where only one side of the material is accessible.

Process windows (figure 1.2), indicate the laser welding quality that is achieved when specific combinations of laser power and welding speed are applied. Too much laser power at low welding speed will result in excessive heat input and the molten metal may flow out of the seam, resulting in a cutting effect. On the other hand, a welding speed which is too high will result in insufficient heat input which leads to lack of fusion. The optimal conditions are found near the top-right part of the boundary between partial and full penetration.

Small differences in the process conditions will influence the boundaries of the process window. Variables that influence the laser welding process are:

- type of metal alloy.
- surface condition of the metal, e.g. roughness, oxidation, protective coatings.
- material thickness.
- edge preparation of the metal.

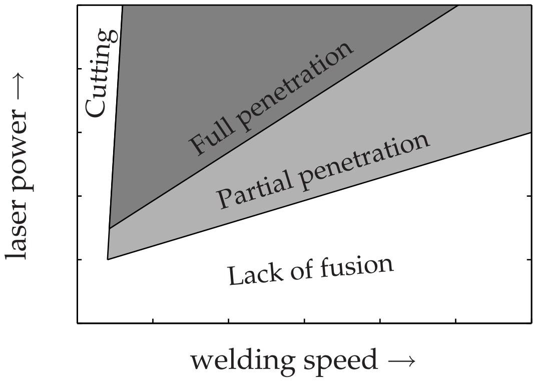


Figure 1.2: A typical process window for laser welding. The different areas indicate the quality of the weld at a certain combination of laser power and welding speed.

- weld type, e.g. butt weld, overlap weld.
- orientation of the weld.
- focal position.
- welding optics.
- shielding gas.

Changing these variables will result in a modified process window.

For linear welds in plate-material, one process window can be considered along the weld seam as the weld conditions are more or less constant. For complex three-dimensional products, the orientation of the weld changes along the weld seam. As the process window changes as a function of the orientation, the welding conditions change as well (Su, 2000). Hence, the optimal settings of the welding parameters will vary along the seam.

The required tracking accuracy is mainly determined by the size of the melt pool, which depends on the material and the spot size of the laser beam. The spot size of the focussed laser beam is a function of wavelength, the diameter of the unfocussed beam and the focal length of the lens. As there is a selection of different focal lengths that may be used, a single set of values for the path tracking accuracy in either lateral and focal direction does not exist. For laser welding of steel sheets using a laser spot diameter of 0.45 mm and a focal length of 150 mm, it was found that a lateral tolerance of ±0.2 mm has to be satisfied in order to avoid weld quality degradation (Duley, 1999; Römer, 2002; Olde Benneker, 2000). The requirements for the focal tolerance are in the same order. Hence, tolerances of ±0.2 mm in both focal and lateral direction will be taken as typical values.

1.1.2 Robotics

The handling of the laser welding head is done with robotic manipulators. For laser welding of two-dimensional seams in plate material, gantry manipulators are being used in industry. Such manipulators are accurate, but the accessibility of welds in three-dimensional assemblies is limited. Six-axis robotic manipulators on the other hand, are more flexible as they can reach more complicated seams, opening a wider range of applications. However, they are less accurate than gantry manipulators. In this work, serial six degree-of-freedom articulated arm robots are used. In figure 1.3, the used components in our laser welding system are shown.

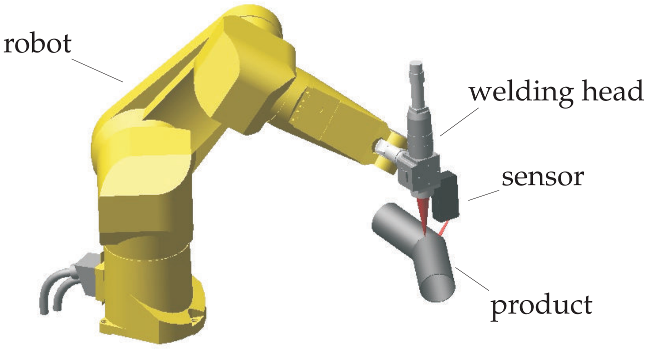


Figure 1.3: Laser welding system consisting of a robot, welding head, sensor and product.

Welding of three-dimensional seams is not a trivial extension of seam welding of two-dimensional seams. Accurate manual programming of these welding trajectories is time-consuming. CAD/CAM or Off-line programming software may be used to generate the welding trajectories, but it remains difficult to accurately match such a trajectory with the actual seam trajectory in the work cell. The important accuracy requirement is the accuracy of the laser focal point relative to the product. This accuracy is influenced by the robotic manipulator and the product, e.g. product tolerances and clamping errors. Sensors will be used to measure and correct both robot errors and product deviations.

The repeatability of six-axis robots is usually much better than their absolute accuracy and is often the only accuracy specification that is given by the robot manufacturer, as specified in the ISO standard 9283:1998. Typical values for the repeatability of six-axis robots are in the range of 0.02 mm to 0.1 mm. Yet, the absolute errors of such robots is commonly an order of magnitude larger and is thus not compatible with the tolerance demands for laser welding. To overcome the lacking absolute accuracy of six-axis robots, in this work a seam sensor is applied to measure the seam position at about 50 mm distance from the laser focal point.

1.1.3 Seam sensing

Sensors will be used to increase the accuracy and ease-of-use of six-axis robotic manipulators for laser welding. Many different sensing techniques for measuring three-dimensional information can be distinguished, e.g. inductive measurements, range measurement using different patterns, surface orientation measurement using different surfaces, optical triangulation.

Sensors based on optical triangulation with structured light are used most often as they provide accurate 3D information at a considerable distance from the product surface. Optical triangulation can be divided in two parts: beam scanning and pattern projecting. In beam scanning, a projector and detector are used that are simultaneously scanning the surface. Pattern projecting is done with different structures, e.g. one or multiple lines, circles, crosses, triangles. A two-dimensional camera observes the structures and features are detected using image processing. In our set-up, a single line pattern projecting sensor is used.

The sensor uses a laser diode to project a line of laser light on the surface at a certain angle. A camera observes the diffuse reflection of the laser diode on the surface at a different angle (section 3.5.2). Deviations on the surface will result in deviations on the camera image, which are detected. Several features are detected from the camera, e.g. the 3D position, an orientation angle. To use the sensor, the seam geometry should be detectable. Typical examples of detectable seam geometries are shown in figure 1.4.

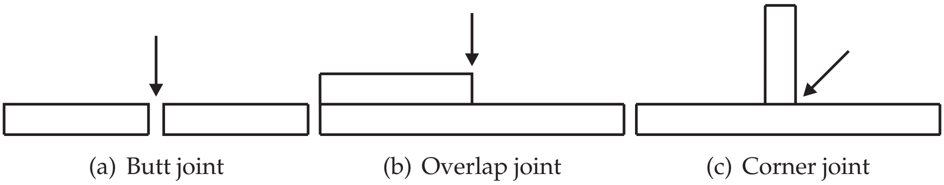

(a) Butt joint (b) Overlap joint (c) Corner joint

Figure 1.4: Typical seam geometries that are detectable by a seam tracking sensor.

1.2 Objective

The objective of this work is formulated as follows:

Development of a sensor-guided robotic laser welding system that is suitable for 3D seam welding in small product series.

A considerable amount of work-preparation time consists of programming the robot for a new welding job. This can be decreased by using a seam tracking sensor for automatic teaching of the seam trajectory. Once a seam has been taught this way, it can be welded in any number of products as long as no modifications of this trajectory occur, e.g. due to clamping errors, product tolerances, heat deviation errors, etc. In cases where modifications of this trajectory do occur, the sensor can be used to correct small errors from a pre-defined seam trajectory, using sensor measurements obtained some distance ahead of the laser focal point. Both the sensor and laser welding head are tools that are attached to the robot flange. The sensor measures relative to its own coordinate frame, but the corrections need to be applied in the laser coordinate frame. The transformations between these coordinate frames and the robot flange need to be accurately calibrated.

Integration of a seam tracking sensor in a robotic laser welding system results in the following tasks:

- Development of sensor-guided teaching procedures for obtaining the actual location of the seam trajectory.
- Development of procedures for real-time seam tracking during laser welding to correct small errors from a pre-defined seam trajectory.
- Development of automatic tool calibration methods for obtaining the transformations between the coordinate frames attached to laser, sensor and robot flange.

These tasks will be described in three chapters in this thesis. From a scientific point-of-view, a number of topics need special attention, which follows from the state-of-the-art.

1.3 State-of-the-art

This section presents the state-of-the art for sensor-guided robotic welding. Two major subjects are discussed: Tool calibration and seam teaching/tracking.

1.3.1 Tool calibration

Generic work on calibration of wrist-mounted robotic sensors has been performed in the late 80's and the beginning of the 90's. It is often referred to as hand-eye calibration. Homogenous transformation equations of the form $\mathbf{AX} = \mathbf{XB}$ are solved, where $\mathbf{X}$ represents the unknown tool transformation, $\mathbf{A}$ is a pre-defined movement relative to the robot flange and $\mathbf{B}$ is the measured change in tool coordinates (chapter 4, figure 4.2). An overview of the relevant literature on tool calibration is given next.

- Chou and Kamel (1988) are the first to write the calibration problem in the form $\mathbf{AX} = \mathbf{XB}$. They present a quaternion approach to mathematically solve the calibration problem.

- Shiu and Ahmad (1989) provide a closed-form solution to the calibration problem and the necessary conditions for uniqueness of the solution are stated.

- Tsai and Lenz (1989) describe a least-squares solution to the calibration problem.

- Chou and Kamel (1991) present a closed-form solution to the calibration problem using quaternions.

- Zhuang and Shiu (1993) present a least squares solution with and without orientation measurements to solve the calibration problem in the presence of noise.

- Nayak and Ray (1993) present a calibration method for an arc welding tool, where a point in 3D-space is approached from different directions

- Park and Martin (1994) describe two tool calibration methods for a wrist-mounted robotic sensor. One is a closed-form exact solution, the second one is a least squares solution to limit the influence of noise.

- Lee and Ro (1996) present a self calibrating model for hand-eye systems with motion estimation. Besides the hand-eye calibration problem, also the calibration of other unknown parameters, like camera parameters and kinematic parameters are addressed. Simulations are used to verify their method.

- Lu et al. (1996) present a least squares method for hand-eye calibration to limit the influence of noise. They compare their method with existing least squares methods using simulations.

- Thorne (1999) filed a US patent that describes a method for tool centre point calibration of a spot welding gun, by rotating around an estimated tool centre point above a flat surface. Only the position is calibrated, the orientation is not calibrated.

- Daniilidis (1999) presents a solution for hand-eye calibration, where the orientation is described using dual quaternions. Both simulations and experiments are performed as well as a comparison between his method and the one found in Tsai and Lenz (1989).

- Huissoon (2002) presents a solution to the calibration problem for sensor-guided robotic laser welding. A combined method using special calibration objects is presented, which calibrates the sensor frame relative to the robot flange frame and the laser frame relative to the sensor frame.

In most of the above-mentioned references, mathematical procedures are presented to solve the generic calibration problem of wrist-mounted sensors. In these publications, the procedures are verified using simulation data only. Daniilidis (1999) and Huissoon (2002) are the only ones who present experimental results. A major drawback in the presented simulations is that in all publications, except Lee and Ro (1996), only the influence of stochastic errors (measurement noise) is considered and not the influence of systematic errors originating from robot geometric non-linearities, optical deviations, sensor reflections, etc. In this thesis, the influence of such systematic errors on the tool calibration procedures is analysed.

Huissoon (2002) is the only author that directly applies the tool calibration methods to robotic laser welding. However, he does not give results of the used sensor tool calibration procedure. No publications were found that apply the calibration procedures to a laser welding head without a sensor. In this thesis, a calibration procedure for a laser welding head is developed, which uses the coaxial camera attached to the welding head and a calibration object.

1.3.2 Seam teaching and seam tracking

A lot of work on seam teaching and seam tracking has been performed in the 90's. An overview of the relevant literature on seam teaching and seam tracking is given next.

- Nayak and Ray (1993) give a thorough description on the subject of seam tracking for robotic welding. Various topics that play a role are dealt with: sensing and image-processing techniques, coordinate frames, high-level and low-level robot motion control, implementation issues and a description of several prototypes of seam tracking systems.

- Carozzi et al. (1995) present an open prototype robot controller that allows easy implementation of model-based and sensor-based control concepts.

- Kim et al. (1996) have developed a seam tracking system for robotic arc welding. The main purpose of the system is to achieve robustness to various process conditions, e.g. arc glare, welding spatters, fume and other unexpected brightness sources.

- Nunes et al. (1996) present a 3D surface-following control algorithm that uses sensor information from an ultrasound sensor.

- Yu and Na (1997) in part one of the paper, discuss the use of vision sensors for seam tracking in robotic arc welding. A mathematical model is proposed to estimate the occurrence of data deficiency in the vision sensor, which can be caused by the shadow effect or missing field-of-view.

- Yu and Na (1998) in part two of the paper, describe the applications.
- Kim et al. (1999) propose an automatic teaching method of the welding trajectory for robotic arc welding. A laser vision sensor was used to program both the position and orientation of the welding trajectory using geometric models of the weld joint, seam and robot trajectory.
- Andersen (2001) presents the practical application of using sensors for robotic laser welding in the ship building industry. His thesis shows little technical details on the used algorithms.
- Bauchspiess et al. (2001) propose a model predictive approach to trajectory tracking based on sensor information. Their method is implemented on a hydraulic robot guided by a CCD-camera.
- Bae et al. (2002) present a camera-based approach to simultaneous seam tracking and weld pool control in gas metal arc welding (GMAW) using fuzzy control.
- Fridenfalk and Bolmsjö (2003) present a 6 degree-of-freedom seam tracking algorithm is presented and implemented for use in the shipbuilding industry
- Luo and Chen (2005) use a laser vision sensor for seam tracking in robotic arc welding of titanium alloys.
- Bae and Park (2006) describe the development of an inductive sensor for seam tracking.

Seam teaching for arc welding has been regarded as being solved by many authors, e.g. Nayak and Ray (1993). They focus on ways to use sensor information for real-time seam tracking and job planning. Most publications apply seam tracking to robotic arc welding, where the accuracy requirements (±1 mm) are less stringent than for laser welding. Two publications (Andersen, 2001; Fridenfalk and Bolmsjö, 2003) apply seam tracking to the application of robotic laser welding. Unfortunately, Andersen (2001) gives no technical details of his algorithms. Fridenfalk and Bolmsjö (2003) describe their algorithm but do not pay much attention on robot integration aspects (e.g. robot-sensor synchronisation). The accuracy of his system (±2 mm) is much larger than the 0.2 mm that is aimed at in this thesis.

No publications were found that analyse the influence of various error sources (e.g. robot geometric errors) on seam teaching and seam tracking. The influence of these error sources on seam teaching and seam tracking will be analysed in this thesis.

1.4 Outline

Chapter 2 gives a description of the equipment that is used in this work. It describes the mechanical and electrical interfaces between the system components as well as the software framework, which plays a major role in this work.

Chapter 3 introduces the concept of homogenous coordinate transformations to mathematically describe transformations between coordinate frames. An overview of the used coordinate frames and transformations in a sensor-guided robotic laser welding system is given. A simulation environment is developed, where the seam trajectory, seam tracking sensor, robot arm and robot controller are modelled. The simulation environment is used to show the separate influence of various errors during tool calibration and seam-teaching.

Chapter 4 describes calibration procedures for accurately obtaining the geometrical transformations between the robot flange, the laser tool and the sensor tool. Special attention is paid to the influence of errors in the robot geometric model during the procedures.

Chapter 5 describes procedures for teaching the seam trajectory with a seam tracking sensor. In this chapter point-to-point movements are used to move the sensor along the seam trajectory. The actual ”blind” welding is done in a separate step after the seam trajectory has been taught.

Chapter 6 presents procedures for real-time seam tracking, where sensor information is used during the robot motion. A synchronisation procedure is presented, which allows to simultaneously use measurements from the sensor and the robot joint encoders. A ”trajectory-based” control architecture is developed, which embeds the synchronisation procedure. It contains a Real-time Setpoint generator, which generates the robot trajectory on-the-fly during the motion. This way the synchronised measurements from the sensor that measures ahead of the laser focal point are used optimally.

Chapter 2

System architecture

2.1 Introduction

A sensor-guided robotic laser welding system consists of different components. This chapter describes the equipment that is used in this work. The system components are described from a generic point of view making the results obtained in this work useful for different hardware as well.

The mechanical and electrical interfaces between the components are also described. A thorough description of the software architecture is given, it plays a major role in generalising the hardware and accessing it in a generic way. An Ethernet-based communication layer has been developed, which allows different robot platforms and sensor systems to be accessed from a computer located on the network. A graphical user interface called 24LASER (Twente-for-laser) has been developed, which can access the robot and sensor through this communication layer. 24LASER contains the tool calibration, seam teaching and seam tracking procedures that are developed in this work and allows them to be used on different robot platforms and sensor systems.

2.2 Equipment

The important hardware components in a sensor-guided robotic laser welding system are:

- Robot arm
- Robot controller
- Seam tracking sensor
- Welding head
- Laser source

These components will be described in the following subsections.

2.2.1 Robot arm

Industrial 6-axis robots from the Stäubli RX series are used in this work. Both a Stäubli RX90 and a Stäubli RX130 are applied. These robots are mostly used in accurate pick and place applications (e.g. semi-conductor industry). They are serial-link robots with six rotational degrees of freedom (figure 2.1), with drives located in the arm. This allows the robot tip to be positioned to any position and orientation within the robot workspace. The specifications of the used robot arms are given in table 2.1.

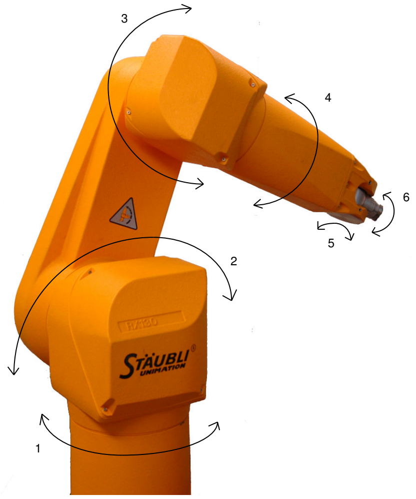


Figure 2.1: Stäubli RX130 robot arm

Robot specifications	RX90	RX130
Nominal payload capacity (kg)	6	12
Maximum payload capacity (kg)	11	20
Workspace (mm)	985	1360
Repeatability (mm)	±0.02	±0.03

Table 2.1: Specifications of the Stäubli RX90 and RX130

2.2.2 Robot controller

The robots are controlled by a Stäubli CS8 controller containing an industrial PC and drive controllers. The PC contains the robot control software and is equipped with a Sercos PCI-expansion card for communication with the drive controllers over a Sercos Bus system (Sercos, 1995). Each drive controller contains the power electronics to control a single axis of the robot.

The CS8 controller uses the VxWorks hard real-time operating system (Wind River, 2002). A standard CS8 controller can be programmed with the VAL3 robot programming language (Stäubli, 2004). In this work, a non-standard CS8 controller with Stäubli's Real-time Robot Controller Abstraction Layer (RTRCAL) is used (Pertin and Bonnet-des-Tuves, 2004). The RTRCAL provides a C++ API that allows the control of the robot on the low level that is required in this work.

A description of the software that runs on the CS8 controller is given in section 2.4.

2.2.3 Seam tracking sensor

A seam tracking sensor developed by Falldorf Sensor GmbH is used in this work. Its working principle is optical triangulation using structured light. A laser diode is mounted under an angle to project a line of laser light on the workpiece. The line is observed by a CMOS camera. The sensor can extract many features from the 2D CMOS image. The features that are used in this work are the 3D-position and an orientation angle. Other features that can be extracted are the height deviation, gap width for butt-welds, light intensity, amount of noise, etc. The specifications of the sensor can be found in table 2.2.

Falldorf specifications	
Stand-off (mm)	56
CMOS chip size (pixels)	512x256
Frame rate (Hz)	200
Length of scan line (mm)	14
Height range (mm)	10
Diode-angle (deg)	33.15

Table 2.2: Specifications of the Falldorf S5A PS 56/14/10 sensor

The Falldorf sensor system uses a dedicated software package called Inspector for processing the CMOS image and extracting the features. Detailed information on the software that runs on the sensor computer can be found in section 2.4.

2.2.4 Welding head

To focus the laser beam on a product different welding heads are used. The principle for these welding heads is similar. The laser light leaves the optical fiber as a diverging beam, which is made parallel with a collimator lens and then focussed on the product using a focussing lens. The focal length f depends on the used lens: 100, 150 or 200 mm lenses are available resulting in a laser spot diameter of 0.3 mm, 0.45 mmm or 0.6 mm respectively.

Water cooling for the collimator and focussing lens is available. Furthermore, a dichroic mirror is mounted in the parallel beam. It allows a camera to observe the process via a separate optical path.

2.2.5 Laser source

The laser light is generated using a 4kW lamp-pumped Nd:YAG laser (Trumph HL4006D) and transported to the welding head through an optical fiber with a diameter of 0.6 mm. Four fibers are connected to the laser allowing the laser to be switched between different working cells.

The laser is connected to a laser controller using digital and analog I/O. The laser controller manages the switching between the different workstations and is used for safety purposes, i.e. it makes sure the laser can not output laser light when the safety conditions (closed workstation door, emergency stop, etc) do not apply. The robot controllers at the different workstations are connected to the laser controller by means of a Devicenet industrial bus system.

2.3 Connection topology

The system components have been described in the previous sections. This section shows how these components are connected and how they interact. The connection topology of the system is shown in figure 2.2.

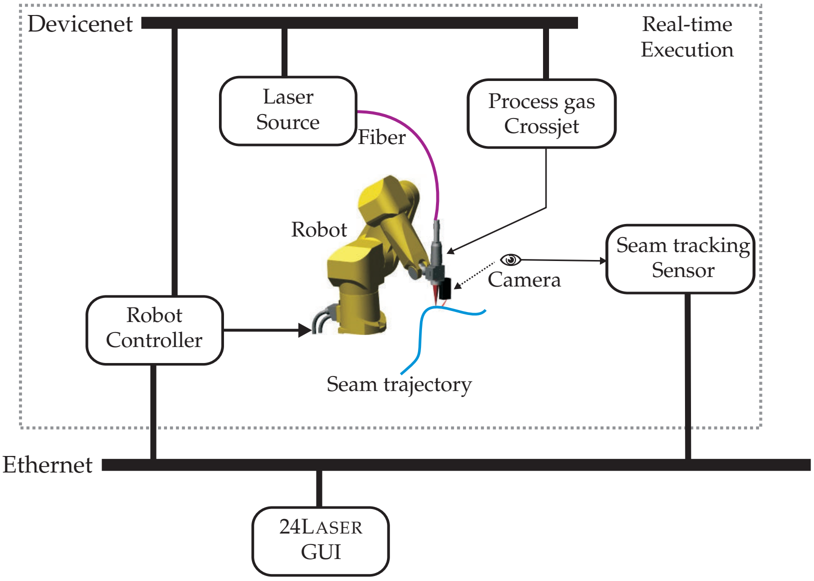


Figure 2.2: Connection topology

All the components inside the dotted box require real-time execution. The robot controller not only controls the robot motion but also the other equipment during laser welding. The laser, crossjet and process gas have to be switched on and off synchronously with the robot motion. The seam tracking sensor is also located within the real-time execution box, as its measurements must be synchronised with the measurements of the robot joint angles. The synchronisation procedure uses Ethernet UDP-communication and is explained in section 6.5.

The operator can control the system using a graphical user interface (GUI), which is called 24LASER . This does not require real-time guarantees. It is located on a computer that is connected to the robot controller using Ethernet. 24LASER is used to prepare and process a laser welding job. Furthermore it contains the implementations of the tool calibration procedures and the seam teaching and tracking algorithms that are described in this thesis.

2.4 Software architecture

The software architecture is schematically shown in figure 2.3. The software that was developed in this work is located on three different computer systems: The robot controller, the sensor system and the PC for the 24LASER GUI. The software is mainly written in C++.

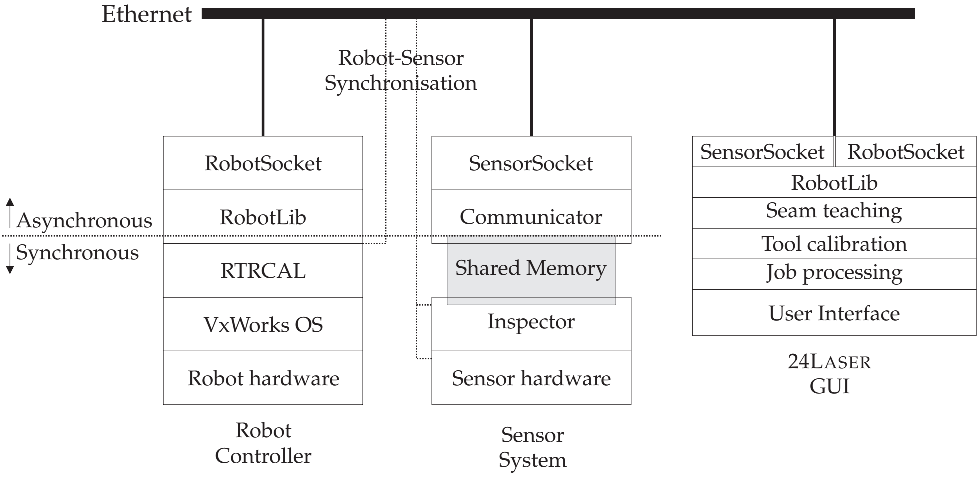


Figure 2.3: Software architecture

VxWorks is the real-time operating system that is running on the robot controller. Stäubli developed the Real-time Robot Controller Abstraction Layer (RTRCAL), which allows user-software to be written that can access and control the robot hardware (section 2.4.1). The RobotLib software is developed in this work. It contains the mathematics and tools needed to control the robot motion and obtain information about the robot status (section 2.4.2). The RobotSocket server is the communication layer that allows clients to connect to the robot and use it (section 2.4.3).

The image-processing of the seam tracking sensor is done by the Inspector software developed by Falldorf Sensor GmbH. After a CMOS image has been processed, Inspector writes the extracted features to a shared memory location. User software called Communicator (section 2.4.4) is developed to read the features from the shared memory, display them on the screen and communicate them to clients that connect to the SensorSocket server (section 2.4.5).

The 24LASER GUI contains a RobotSocket client and a SensorSocket client to connect to the robot and sensor respectively. This way, the communication between the robot, sensor and 24LASER is generic, different robots and sensors providing the correct API can be used as well.

The software parts that were used or developed in this work will be described in detail in the following sections.

2.4.1 Real-time Robot Controller Abstraction Layer

The RTRCAL (figure 2.4) or Low-Level-Interface (LLI) is a software package that is developed by Stäubli. It manages the robot joint motion control, motor power control, brake control and I/O through digital, analog or fieldbus interfaces and takes care of basic safety and calibration. The User Application can access the robot through a simple and general C/C++ software interface.

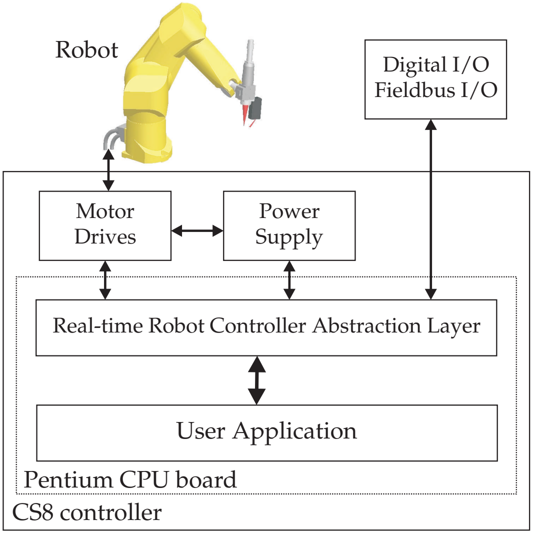


Figure 2.4: Real-time Robot Controller Abstraction Layer

The CS8 joint control structure is shown in figure 2.5. The user application must provide the RTRCAL with a joint command, consisting of desired joint position, joint velocity, torque and velocity feedforwards every 4 ms. In order to move it smoothly, the programmer must correctly calculate these setpoints. A joint feedback structure, containing the measured joint position, joint velocity, torque and position error is available at a rate of 4 ms. Internally, the CS8 controller interpolates the joint setpoints to a much higher frequency, where the joint control is carried out using a position control loop, a velocity control loop and current control loop, proprietary to Stäubli.

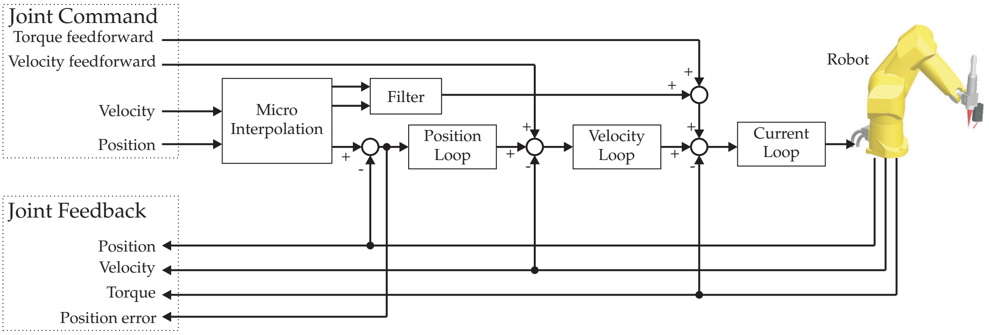


Figure 2.5: Control Structure of the CS8 joint motion controller

More information on the RTRCAL can be found in Pertin and Bonnet-des-Tuves (2004).

2.4.2 RobotLib

The developed RobotLib library contains the tools and mathematics to work with robots. It consists of a number of C++ classes for describing quaternions, transformations, tools, frames, locations, robots, trajectories, trajectory generation, etc. The RobotLib library depends on the Newmat library (Davies, 2002) for basic matrix and vector manipulation (add, subtract, multiply, etc). Detailed information on the RobotLib library can be found in appendix A.1.1.

2.4.3 RobotSocket

The RobotSocket communication layer is an Ethernet socket communication layer to let the robot communicate with its surroundings. It uses the client-server model: on the side of the robot controller, a RobotSocket server waits for RobotSocket clients that connect to it. The RobotSocket communication layer is programmed as a C++ class.

The RobotSocket server listens on a specific port (default 4000) for incoming TCP request packets. Request packets have a certain structure in which e.g. a Command ID is included to specify the requested robot command. After the request packet has been identified as being a correct request packet, the RobotSocket server performs the request based on the Command ID. If the request was performed correctly, the RobotSocket server responds with a reply packet. If an error occurred during processing of the request the RobotSocket server responds with an error packet.

This approach has several major advantages:

- Ethernet communication is fast (100 Mbit/s) and cheap. No additional I/O hardware is needed as the robot controller and the computers are standardly equipped with a network controller.
- Several error-checks are carried out, which guarantees correct packet delivery and prevents unexpected robot behaviour.
- The system is robot-independent. Robots from different manufacturers and with different controllers can be used as long as they follow the Application Programming Interface (API) as defined in the RobotSocket class.
- Robot emulators can be used. Instead of real robot hardware it is possible to connect to a robot emulator, which follows the RobotSocket API. This is ideal for testing and debugging purposes as no ”damage” can occur. Furthermore, these robot emulators can contain models of the robot and its surrounding. This way the influence of different factors (kinematics, dynamics, tool transformation errors, etc) on the path accuracy can be predicted and visualised.

The API of the RobotSocket communication layer is developed in a generic way, so different commands using a list of different parameters can be sent and received. It can be easily extended with different commands by adding additional Command ID’s and parameters.

2.4.4 Communicator

The Communicator software is written for the following tasks:

- To display current sensor data, sensor profile and intensity profile. The Falldorf seam tracking sensor can extract several features (appendix A.2) from a single camera image. These are displayed to provide feedback to the user.
- To communicate sensor features through the Ethernet to clients using the SensorSocket communication layer (section 2.4.5).
- To choose between real sensor data, or simulated data generated using models of the seam tracking sensor and seam trajectories. The modelling of sensor and seam trajectory is described in section 3.5.

2.4.5 Sensorsocket

The SensorSocket communication layer is an Ethernet socket communication layer, used to let the sensor communicate with its surroundings. It is similar to the RobotSocket communication layer described in section 2.4.3.

The SensorSocket server listens on a specific port (default 2000) for incoming TCP request packets. Request packets have a special structure. If the request packet is identified as being correct, the SensorSocket server carries out the request based on the Command ID within the request packet. If the request is carried out correctly, the SensorSocket server responds with a reply packet. If an error occurred during processing of the request the SensorSocket server responds with an error packet.

2.4.6 24Laser

The 24Laser GUI is the link between the robot operator and the hardware and software. 24Laser allows the robot operator to carry out a number of different tasks:

- Robot status and motion control. The status of the robot can be visualised, e.g. the status of Emergency Stops, Motion, Robot Power, Joint and Cartesian position. Furthermore, motion commands can be given to the robot.

- Seam teaching using point-to-point movements. Both manual teaching of locations and 3D seam teaching with the aid of a seam tracking sensor (described in chapter 5) can be carried out from the GUI.

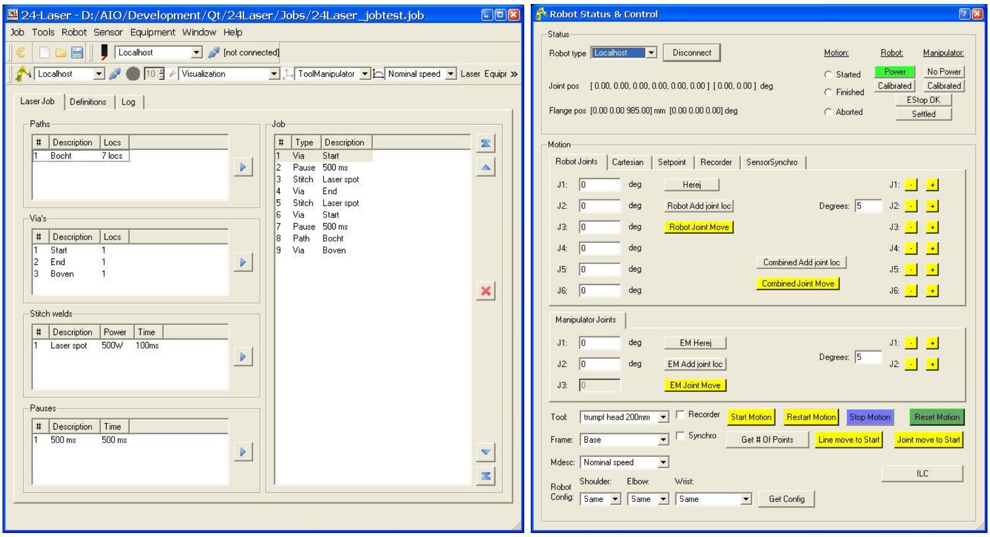
- Tool Calibration using point-to-point movements. Automatic tool calibration procedures for finding the transformations of the laser tool and sensor tool can be carried out from the GUI. These procedures are described in chapter 4.

- Job preparation. This task consists of setting the parameters on a welding trajectory, e.g. velocity, laser on/off, laser power, process gas on/off, crossjet on/off.

- Job processing. A welding job may consist of paths (welding trajectories), via locations (in-between locations), stitch welds (spots) and pauses. During job processing these are processed in a user-defined order.

Furthermore the GUI contains a database where different tool transformations and frame transformations can be stored and edited. Welding jobs can be loaded and saved in an XML file-format (XML, 2003).

The 24Laser GUI is developed using Trolltech's Qt software (Trolltech, 2005). Qt is a C++ software framework used for programming Graphical User Interfaces.



24-Laser - D:/AIO/Development/Qt/24Laser/Jobs/24Laser_jobtest.job

Localhost [not connected]

Localhost 10 Visualization ToolManipulator Nominal speed

Paths

#	Description	Locs
1	Bocht	7 locs

Via's

#	Description	Locs
1	Start	1
2	End	1
3	Boven	1

Stitch welds

#	Description	Power	Time
1	Laser spot	500W	100ms

Pauses

#	Description	Time
1	500 ms	500 ms

Job

#	Type	Description
1	Via	Start
2	Pause	500 ms
3	Stitch	Laser spot
4	Via	End
5	Stitch	Laser spot
6	Via	Start
7	Pause	500 ms
8	Path	Bocht
9	Via	Boven

(a) Main window

Robot Status & Control

Robot type: Localhost

Joint pos [0.00, 0.00, 0.00, 0.00, 0.00, 0.00] [0.00, 0.00] deg

Flange pos [0.00 0.00 985.00] mm [0.00 0.00 0.00] deg

Motion: ○ Started ○ Finished ○ Aborted

Robot: Power, Calibrated

Manipulator: No Power, Calibrated

Robot Joints: J1: 0 deg; J2: 0 deg; J3: 0 deg; J4: 0 deg; J5: 0 deg; J6: 0 deg; Degrees: 5

Manipulator Joints: J1: 0 deg; J2: 0 deg; J3: 0; Degrees: 5

Tool: trumpf head 200mm □ Recorder

Frame: Base □ Synchro

Mdesc: Nominal speed

Robot Config: Shoulder: Same Elbow: Same Wrist: Same

(b) Robot control window

*Figure 2.6: Screenshots of the 24*L<span style="font-variant:small-caps">ASER</span> *graphical user interface.*

It features cross-platform capabilities as the source can be compiled for MS Windows, Linux and Macintosh operating systems. 24LASER screenshots are shown in figure 2.6.

24LASER uses the RobotLib, RobotSocket and SensorSocket static libraries that were described in the previous sections. Furthermore a number of dynamic libraries have been developed for use in 24LASER . They were programmed as Dynamic Link Libraries (DLL's), allowing easier parallel software development:

- Seamtracking DLL. Used for real-time Seam tracking. Details of the used algorithms can be found in chapter 6.

- LaserCalib DLL. Used for laser tool calibration. The used algorithms and measurement method are described in section 4.2.

- SensorCalib DLL. Used for sensor tool calibration. The used algorithms and measurement method are described in section 4.3.

- Sensor2LaserCalib DLL. Used for a combined laser and sensor tool calibration. The used method is described in section 4.4.

- ILC DLL. Used for Iterative Learning Control (ILC), which is a control strategy used to increase the tracking accuracy of the joint motion controller, by repeating the trajectory and learning from errors made in previous runs. ILC is described in Hakvoort et al. (2006).

- PlugIn DLL. A generic DLL for integration within 24Laser . User functions contained in the DLL, making use of RobotSocket and SensorSocket, can be called from within 24Laser .

2.5 Discussion

In this chapter a description of the components that exist in a sensor-guided robotic laser welding system is given. A generic software framework has been described, which uses Ethernet sockets for communication between the system components. This makes the framework independent of the robot type or sensor type that is used. The software framework has been implemented on a Stäubli six-axis industrial robot and a commercially available seam tracking sensor made by Falldorf Sensor GmbH, but can be easily extended to other robots, lasers or sensors.

Chapter 3

Geometric modelling

3.1 Introduction

The goal of this chapter is to provide a realistic simulation of the measurements of the seam tracking sensor as it is being moved or positioned by a robot with respect to a seam in a workpiece. These simulations can be used in the sensor tool calibration procedure and seam teaching algorithms that will be described in the following chapters. To achieve this, the following components will be modelled:

- Seam trajectory
- Seam tracking sensor
- Robot arm and controller

In order to geometrically model a sensor-guided robotic laser welding system, the different coordinate frames that exist in such a systems will be described. The concept of homogenous transformation matrices is presented to describe the transformations between the coordinate frames.

3.2 Homogenous transformation matrix

The hardware components described in the previous chapter are physically connected. To be able to mathematically describe their position and orientation, a coordinate system or frame is attached to each component. The position and orientation of such a frame can be described relative to another frame. Frames are indicated by a bold capital. A transformation describes the location (position and orientation) of a frame with respect to a reference frame. Transformations are indicated by the symbol **T** and may have a leading superscript, that defines the reference frame they refer to. The leading subscript defines the frame they describe, e.g. transformation ${}^{A}_{B}\mathbf{T}$ describes frame **B** relative to frame **A**.

In literature, a 4x4 homogenous transformation matrix is often used to describe a transformation. A transformation ${}^{A}_{B}\mathbf{T}$ can be written as

$${}^{A}_{B}\mathbf{T} = \begin{bmatrix} {}^{A}_{B}\mathbf{R} & {}^{A}_{B}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.1}$$

where ${}^{A}_{B}\mathbf{R}$ is a 3x3 rotation matrix that describes the orientation of frame **B** with respect to frame **A** and ${}^{A}_{B}\mathbf{P}$ is a 3D position vector that describes the position of the origin of frame **B** with respect to frame **A**. In the remainder of this work the leading subscript and superscript of a transformation (or its components) may be omitted to make the equations better readable.

The position vector **P** contains the elements P_x, P_y and P_z, in x, y and z-direction respectively, as

$$\mathbf{P} = [P_x \quad P_y \quad P_z]^T. \tag{3.2}$$

The rotation matrix **R** consists of three orthonormal vectors $\mathbf{r_x} = [r_{xx} \quad r_{xy} \quad r_{xz}]^T$, $\mathbf{r_y} = [r_{yx} \quad r_{yy} \quad r_{yz}]^T$ and $\mathbf{r_z} = [r_{zx} \quad r_{zy} \quad r_{zz}]^T$, that describe the unit vectors along the three coordinate axes. It can be written as

$$\mathbf{R} = [\mathbf{r_x} \quad \mathbf{r_y} \quad \mathbf{r_z}] = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix}. \tag{3.3}$$

The homogenous transformation matrix can now be written in element-form as

$$\mathbf{T} = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} & P_x \\ r_{xy} & r_{yy} & r_{zy} & P_y \\ r_{xz} & r_{yz} & r_{zz} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.4}$$

Homogenous transformation matrices have several useful properties that will be used intensively in this thesis. A summary of these properties can be found in appendix B, more information can be found in textbooks on robotics, e.g. Paul (1982), Craig (1986), Khalil and Dombre (2002).

3.3 Representations of orientation

Different representations exist to describe the orientation between two coordinate frames, like yaw-pitch-roll, Euler angles, axis-angle, Euler parameters or quaternions and direction cosines (Khalil and Dombre, 2002). Khalil and Dombre show how these forms can be computed from one to another. In this work the quaternion and Euler angles representation are frequently used, therefore their definitions are given.

3.3.1 Quaternions

The quaternions are also called Euler parameters or Olinde-Rodrigues parameters. They were invented by Hamilton (1853). A quaternion $\mathbf{Q}$ has a scalar part q_s and a 3D vector part $\mathbf{q_v} = [q_{v_1} \;\; q_{v_2} \;\; q_{v_3}]^T$. For a rotation with angle θ around a unit 3D vector $\mathbf{u}$, the quaternion $\mathbf{Q}$ can be written as

$$\mathbf{Q} = \begin{bmatrix} q_s \\ \mathbf{q_v} \end{bmatrix} = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ \mathbf{u} \cdot \sin(\frac{\theta}{2}) \end{bmatrix}. \tag{3.5}$$

Appendix B describes the quaternion operations that are used in this thesis and the conversions between a quaternion and a rotation matrix and vice-versa.

3.3.2 Euler angles

The Euler angles specify three successive rotations around a principal coordinate axis. There are twelve meaningful choices for the order of rotation, the specific choice should always be explicitly mentioned (Craig, 1986). In this work the X-Y-Z Euler angles form is used. A rotation matrix $\mathbf{R}$ is described by successively rotating an angle γ around the X-axis, an angle β around the Y-axis and finally an angle φ around the Z-axis as

$$\mathbf{R} = \mathbf{rot}(\mathbf{x}, \gamma) \cdot \mathbf{rot}(\mathbf{y}, \beta) \cdot \mathbf{rot}(\mathbf{z}, \varphi), \tag{3.6}$$

where $\mathbf{rot}()$ is the rotation matrix that describes a pure rotation around one of the principal axes. Appendix B describes the conversions between Euler angles and the rotation matrix and vice-versa.

The Euler angles are commonly used if complete transformations are entered by the operator or if visual feedback concerning a transformation is given back to the operator, as they require only three parameters that are easy to visualise from a human point of view. In this work, a complete transformation is often written in the form $[P_x \;\; P_y \;\; P_z \;\; \text{mm}\gamma \;\; \beta \;\; \varphi\text{deg}]^T$, where P_x, P_y and P_z determine the position and γ, β and φ determine the orientation according to the X-Y-Z Euler representation.

3.4 Coordinate frames and transformations

An overview of the frames and the transformations that can be distinguished in a sensor-guided robotic welding system can be found in Nayak and Ray (1993). Their overview is generalised, by adding a Station frame $\mathbf{M}$ and a Product frame $\mathbf{H}$ (figure 3.1).

The following frames are defined:

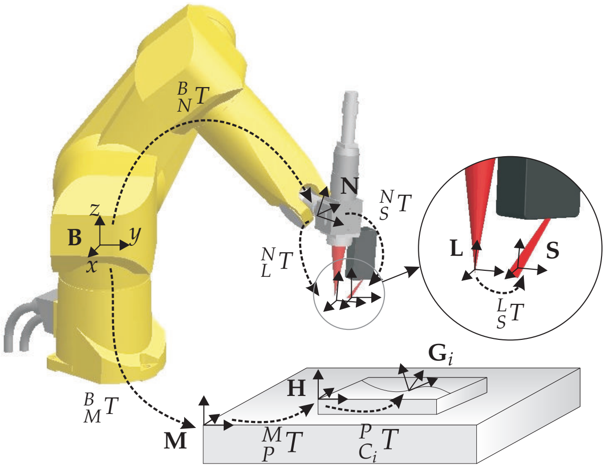


Figure 3.1: Frames and transformations for sensor-guided robotic laser welding

B: Base or world frame. This frame is attached to the robot base and does not move with respect to the environment.

N: Null frame. The Null frame is located at the robot end-effector or the end of the robot flange. The Null frame is described with respect to the Base frame by coordinate transformation ${}^B_N\mathbf{T}$, which is a function of the joint positions of the robot arm (section 3.5.3).

L: Laser tool frame. The Laser tool frame is located at the focal point of the laser beam. The z-axis of this frame coincides with the laser beam axis. Because the laser beam is axi-symmetric, the direction of the x-axis is arbitrary, unless the symmetry is broken due to e.g. the presence of peripheral equipment. Without of loss of generality it will be chosen in the direction of the Sensor tool frame in this work. The transformation ${}^N_L\mathbf{T}$ describes the laser tool frame with respect to the Null frame. This is a fixed transformation determined by the geometry of the welding head.

S: Sensor tool frame. The seam tracking sensor is fixed to the welding head and therefore indirectly to the robot flange. The transformation ${}^N_S\mathbf{T}$ describes the sensor tool frame with respect to the Null frame. The origin of **S** and the direction of the coordinate axes will be defined in section 3.5.2. Note that, because both transformations are fixed, this transformation can also be described with respect to the Laser tool frame instead of the Null frame by transformation ${}^L_S\mathbf{T}$.

M: Station frame. The station frame is the base of the station or work table a product is attached to. It is possible that the product is clamped on a manipulator which moves the product with respect to the base frame. In that case the transformation ${}^{B}_{M}\mathbf{T}$ describes the station frame with respect to the base frame and depends on the joint values of the manipulator.

H: Product frame. The product frame is located on a product. The transformation ${}^{M}_{H}\mathbf{T}$ describes the product frame with respect to the station frame. This frame is useful if a series of similar products are welded on different locations of a station.

G: Seam frame. Every discrete point on a seam can be described with a different coordinate frame (section 3.5.1). The transformation ${}^{H}_{G}\mathbf{T}$ describes a seam frame with respect to the product frame.

T: Robotic tool frame. A general robot movement is specified by the movement of a robotic tool frame **T**, which can be either the sensor tool **S** or laser tool **L**. Some equations that are derived in this work account for either **S** or **L**. In that case the symbol **T** will be used.

F: Robotic frame. A general robot movement is specified with respect to a frame **F**, which can be the base frame **B**, station frame **M**, product frame **H** or seam frame **F**. The symbol **F** will be used in general equations that account for all of these frames.

In many cases an external manipulator is not present and a series of products will only be welded at a fixed location in the work cell. Both ${}^{B}_{M}\mathbf{T}$ and ${}^{M}_{H}\mathbf{T}$ can then be chosen as unity. A seam frame **G** is then described with respect to the robot base with transformation ${}^{B}_{G}\mathbf{T}$.

3.5 Modelling

The goal of this section is to simulate the sensor measurements in a sensor-guided robotic laser welding system. To achieve this goal, the combination of robot, sensor and seam is modelled. With these models, different error sources that occur in a sensor-guided robotic laser welding system can be switched on and off to study the effect of these error sources independently. These models will be used for simulating the tool calibration and seam teaching procedures in the next chapters.

3.5.1 Seam trajectory model

A seam trajectory can be considered as a continuous curve in 3D space (figure 3.2). From a mathematical point of view, the position of a discrete point n on the seam trajectory can be described by a 3D vector $\mathbf{P}_n$. The modelled seam trajectory should also include a description of the orientation. The orientation of a discrete point on the seam trajectory is described by the surface normal $\mathbf{n}_n$ of the workpiece. Note that the surface normal in a single point on a seam trajectory does not provide information about the direction along the seam trajectory. This direction can only be derived if at least 2 points (and their order) are known.

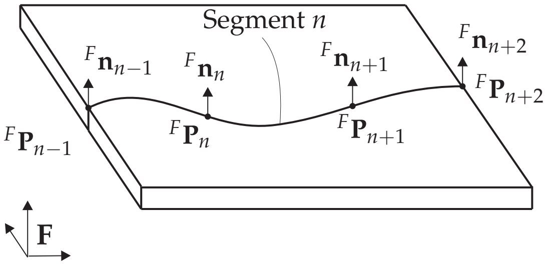


Figure 3.2: Seam trajectory model

The seam model shown in figure 3.2 is defined by choosing a number of discrete points on the seam trajectory that each consist of a position vector and a surface normal. In a product, the seam trajectory is continuous. Therefore the seam trajectory will be interpolated between the discrete base points. A segment-wise cubic parametric spline of the form

$$ {}^F\mathbf{K}_n(\lambda) = \begin{bmatrix} {}^F\mathbf{P}_n(\lambda) \\ {}^F\mathbf{n}_n(\lambda) \end{bmatrix} = \mathbf{a}_n\lambda^3 + \mathbf{b}_n\lambda^2 + \mathbf{c}_n\lambda + \mathbf{d}_n, \tag{3.7} $$

is chosen as the interpolation function, where ${}^F\mathbf{P}_n(\lambda)$ represents the position vector and ${}^F\mathbf{n}_n(\lambda)$ the surface normal with respect to frame $\mathbf{F}$. The coefficients on the segment are calculated with the Catmull-Rom (Barnhill and Riesenfled, 1974) expression as

$$ \begin{bmatrix} \mathbf{a}_n^T \\ \mathbf{b}_n^T \\ \mathbf{c}_n^T \\ \mathbf{d}_n^T \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} {}^F\mathbf{K}_{n-1}^T \\ {}^F\mathbf{K}_n^T \\ {}^F\mathbf{K}_{n+1}^T \\ {}^F\mathbf{K}_{n+2}^T \end{bmatrix}. \tag{3.8} $$

Given a spline segment n between ${}^F\mathbf{K}_n$ and ${}^F\mathbf{K}_{n+1}$ as shown in figure 3.2, four base points are needed to calculate the spline coefficients $\mathbf{a}_n$, $\mathbf{b}_n$, $\mathbf{c}_n$ and $\mathbf{d}_n$. The spline-functions of equation 3.7 exist between the start of segment n at $\lambda = 0$ (where ${}^F\mathbf{P}_n(0) = {}^F\mathbf{P}_n$ and ${}^F\mathbf{n}_n(0) = {}^F\mathbf{n}_n$) and the end of the segment at $\lambda = 1$ (where ${}^F\mathbf{P}_n(1) = {}^F\mathbf{P}_{n+1}$ and ${}^F\mathbf{n}_n(1) = {}^F\mathbf{n}_{n+1}$). An advantage of the cubic spline is that it can be used to interpolate complex trajectories with a small number of base points. Another useful property of the spline is its smoothness, because the first derivative of the interpolation function is continuous at the boundaries of the interval.

On the first and last segment of the seam trajectory, ${}^{F}\mathbf{K}_{n-1}$ and ${}^{F}\mathbf{K}_{n+2}$ respectably, are not available. In order to calculate the spline coefficients in these cases, virtual points ${}^{F}\mathbf{K}_{n-1}$ and ${}^{F}\mathbf{K}_{n+2}$ are used, which are computed by linearly extrapolating as

$${}^{F}\mathbf{K}_{n-1} = 2{}^{F}\mathbf{K}_{n} -{}^{F}\mathbf{K}_{n+1}, \tag{3.9}$$

for the first segment and

$${}^{F}\mathbf{K}_{n+2} = 2{}^{F}\mathbf{K}_{n+1} -{}^{F}\mathbf{K}_{n}, \tag{3.10}$$

for the last segment. If the seam trajectory consists of only of two base points, equation 3.7 reduces from cubic interpolation to linear interpolation ($\mathbf{a}_n = \mathbf{b}_n = 0$) this way.

3.5.2 Sensor model

The sensor is schematically shown in figure 3.3. It uses a laser diode to project a line of structured light onto a surface. A camera looks at this line on under an angle α. The sensor measures four degrees-of-freedom, namely three positions and one rotation angle, as will be outlined below. Furthermore a safety bit, which is called Onsheet is available. It becomes `true` if the seam trajectory is within the field-of-view of the sensor. These five features will be modelled in this section.

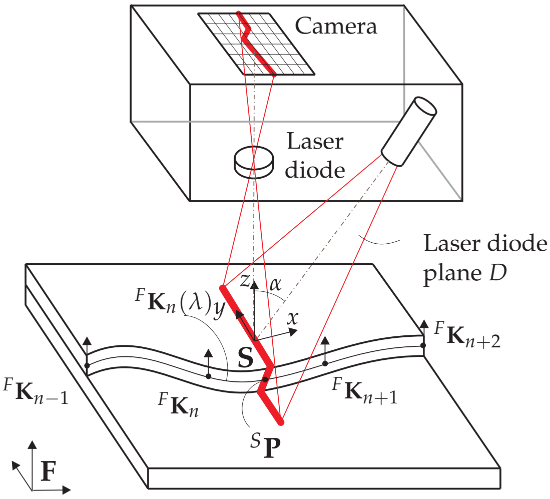


Figure 3.3: Principle of Optical Triangulation. Note that the sensor frame **S** *does not necessarily coincide with the top surface of the workpiece as in this illustration.*

The sensor uses a camera to look at the intersection ${}^{S}\mathbf{P} = [s_x \ \ s_y \ \ s_z]^T$ of the seam trajectory with the laser diode plane D. The 3D coordinates of the intersection are projected on the 2D camera chip. The position of the camera chip and the definition of the sensor frame **S** are such that s_y and s_x follow directly from the camera image. As the intersection has to be in the diode plane D, s_z and s_x are related using the known diode angle α as

$$s_x = s_z \cdot \tan(\alpha). \tag{3.11}$$

The origin of the sensor frame **S** is defined in the centre of the camera image on the laser diode plane. The z-axis of **S** is normal to the camera chip. The sensor measures the position of the intersection ${}^{S}\mathbf{P}$ relative to its coordinate frame **S**. A mathematical model of the sensor should therefore contain the computation of the intersection of the spline function given in equation 3.7 with the laser diode plane.

A location ${}^{S}_{G}\mathbf{T}$ on the seam trajectory with respect to its coordinate frame **S** is computed as

$${}^{S}_{G}\mathbf{T} = \mathbf{Trans}(s_x, s_y, s_z) \cdot \mathbf{Rot_x}(-s_{rx}), \tag{3.12}$$

where s_x, s_y and s_z are the sensor position measurements and s_{rx} is the sensor orientation measurement. $\mathbf{Trans}()$ is a homogenous transformation matrix denoting a pure translation and $\mathbf{Rot_x}()$ is a transformation denoting a pure rotation around the x-axis. The Falldorf GmbH Inspector software (appendix A) computes the real sensor measurements from a camera image. Simulated sensor measurements are derived in the following sections.

Computing the intersection

Suppose that both the sensor tool location ${}^{F}_{S}\mathbf{T}$ and the positional part of the interpolated seam model ${}^{F}\mathbf{P}_n(\lambda)$ are known with respect to some frame **F**. To simplify the computations, an extra frame **D** on the laser diode plane is introduced, which is obtained by rotating frame **S** around its y-axis with an angle of α. So frame **D** has the same origin as **S** and the transformation ${}^{S}_{D}\mathbf{T}$ is

$${}^{S}_{D}\mathbf{T} = \mathbf{Rot_y}(\alpha), \tag{3.13}$$

where $\mathbf{Rot_y}$ is the homogenous transformation matrix corresponding to a pure rotation with an angle α around the y-axis (appendix B). The x-axis of frame **D** is normal to the laser diode plane D. The positional part ${}^F\mathbf{P}_n(\lambda)$ of the interpolated seam model can be expressed with respect to frame $\mathbf{D}$ as

$$\begin{bmatrix} {}^D\mathbf{P}_n(\lambda) \\ 1 \end{bmatrix} = {}^S_D\mathbf{T}^{-1} \cdot {}^F_S\mathbf{T}^{-1} \cdot \begin{bmatrix} {}^F\mathbf{P}_n(\lambda) \\ 1 \end{bmatrix}. \tag{3.14}$$

Because the x-axis of frame $\mathbf{D}$ is normal to the laser diode plane, only ${}^DP_{x,n}(\lambda)$ is needed to find an intersection between the interpolated seam model and the laser diode plane. At such an intersection, the value of λ^{int} needs to be computed that satisfies the condition

$${}^DP_{x,n}(\lambda^{\text{int}}) = 0. \tag{3.15}$$

First of all, the segment n^{int} where an intersection takes place needs to be found. At such an intersection, the sign of ${}^DP_{x,n}$ at the start of segment n should be compared with the sign of ${}^DP_{x,n+1}$ at the end of the segment. An intersection of the seam trajectory with the laser diode plane does occur at segment n if

$${}^DP_{x,n} \cdot {}^DP_{x,n+1} < 0. \tag{3.16}$$

If no intersection exists on any of the segments of the seam trajectory, the seam trajectory is not within the field-of-view of the sensor. In that case, the Onsheet bit will be set to `false`. If a segment n with an intersection exists, equation 3.7 is differentiated with respect to λ to get

$$\frac{d}{d\lambda}{}^F\mathbf{K}_n(\lambda) = \frac{d}{d\lambda}\begin{bmatrix} {}^F\mathbf{P}_n(\lambda) \\ {}^F\mathbf{n}_n(\lambda) \end{bmatrix} = 3\mathbf{a}_n\lambda^2 + 2\mathbf{b}_n\lambda + \mathbf{c}_n. \tag{3.17}$$

And by pre-multiplying with ${}^S_DT^{-1}$ and ${}^F_ST^{-1}$ the derivative of ${}^D\mathbf{P}_n(\lambda)$ with respect to λ can be obtained as

$$\begin{bmatrix} \frac{d}{d\lambda}{}^D\mathbf{P}_n(\lambda) \\ 1 \end{bmatrix} = {}^S_DT^{-1} \cdot {}^F_ST^{-1} \cdot \begin{bmatrix} \frac{d}{d\lambda}{}^F\mathbf{P}_n(\lambda) \\ 1 \end{bmatrix}. \tag{3.18}$$

Using equations 3.14 and 3.18, ${}^DP_{x,n}(\lambda)$ and derivative $\frac{d}{d\lambda}{}^DP_{x,n}(\lambda)$ can be evaluated for any value of λ. By properly choosing the number of discrete points on the seam trajectory, the segments on this trajectory are only moderately curved. The higher order terms in equation 3.7 are therefore relatively small compared to the lower order terms. An iterative Newton-Raphson root-finding method

(Abramowitz and Stegun, 1972) is used to numerically compute the value of λ^{int} on the segment n that satisfies equation 3.15. Every iteration step j, a new value λ_{j+1} is computed as

$$\lambda_{j+1} = \lambda_j - \Delta\lambda_j, \tag{3.19}$$

where

$$\Delta\lambda_j = \frac{{}^{D}P_{x,n}(\lambda_j)}{\frac{d}{d\lambda}{}^{D}P_{x,n}(\lambda_j)}, \tag{3.20}$$

until the following convergence criterium is met

$$|{}^{D}P_{x,n}(\lambda_{j+1})| < \epsilon. \tag{3.21}$$

The convergence error denoted by ϵ is a small number. In order to improve the convergence speed and robustness of the algorithm, the value of $\Delta\lambda_j$ is divided by two if

- The computed λ_{j+1} is not on the segment between 0 and 1.
- The convergence error increased, so if $|{}^{D}P_{x,n}(\lambda_{j+1})| > |{}^{D}P_{x,n}(\lambda_j)|$.

As a starting value for λ_0, equation 3.21 is evaluated at the start ($\lambda = 0$) and at the end ($\lambda = 1$) of the segment. The value of λ is chosen, where $|{}^{D}P_{x,n}(\lambda)|$ is smallest.

Once the value λ^{int} at the intersection on segment n^{int} has been determined, the sensor 3D position measurements are calculated. Using the positional part of equation 3.7 and pre-multiplying with ${}^{F}_{S}\mathbf{T}^{-1}$, they are found as

$$\begin{bmatrix} s_x \\ s_y \\ s_z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{S}\mathbf{P}_{n^{\text{int}}}(\lambda^{\text{int}}) \\ 1 \end{bmatrix} =^{F}_{S} \mathbf{T}^{-1} \cdot \begin{bmatrix} {}^{F}\mathbf{P}_{n^{\text{int}}}(\lambda^{\text{int}}) \\ 1 \end{bmatrix}. \tag{3.22}$$

Orientation feature

Beside the three position features, the seam tracking sensor also calculates an orientation angle s_{rx}. This orientation angle is defined as the angle between the workpiece plane and the laser diode plane at the intersection, as it is seen in the sensor camera-plane. By substituting the value of λ^{int} from the previous section into equation 3.7, the surface normal ${}^{F}\mathbf{n}_{n^{\text{int}}}(\lambda^{\text{int}})$ of the seam trajectory at the intersection is computed with respect to frame **F**. It can be computed with respect to frame **S** as

$$\begin{bmatrix} {}^{S}\mathbf{n}_{n^{\text{int}}}(\lambda^{\text{int}}) \\ 1 \end{bmatrix} = {}^{F}_{S}T^{-1} \begin{bmatrix} {}^{F}\mathbf{n}_{n^{\text{int}}}(\lambda^{\text{int}}) \\ 1 \end{bmatrix}. \tag{3.23}$$

The vector ${}^{S}\mathbf{J}$ represents the intersection line between the workpiece plane and the laser diode plane with respect to the sensor frame. It is calculated as the cross-product of the two surface normals of these planes as

$${}^{S}\mathbf{J} = {}^{S}_{D}\mathbf{r}_{x} \times {}^{S}\mathbf{n}_{n^{\text{int}}}(\lambda^{\text{int}}), \tag{3.24}$$

where ${}^{S}_{D}\mathbf{r}_{x}$ is the surface normal of the laser diode plane and ${}^{S}\mathbf{n}_{n^{\text{int}}}(\lambda^{\text{int}})$ is the surface normal of the seam trajectory at the intersection, both with respect to the sensor frame. From equation 3.13, ${}^{S}_{D}\mathbf{r}_{x}$ is found as

$${}^{S}_{D}\mathbf{r}_{x} = \begin{bmatrix} \cos\alpha \\ 0 \\ -\sin\alpha \end{bmatrix}. \tag{3.25}$$

The camera-plane corresponds to the xy-plane of the sensor. The orientation angle s_{rx} is therefore computed as

$$s_{rx} = \arctan \frac{{}^{S}J_{x}}{{}^{S}J_{y}}. \tag{3.26}$$

Sensor noise and quantisation

The seam tracking sensor computes its features from a camera image, which consists of a limited number of pixels. Therefore the resolution of the features is also limited. Furthermore noise will be present due to external disturbances, like the amount of ambient light on the camera chip, fluctuations in the reflected structured light, sensor motion, etc.

Experiments have been performed by Van der Zee (2003) and Diphoorn (2004) to determine the amount of noise and quantisation. The noise and quantisation appeared to be different for the four degrees-of-freedom. Hence the sensor model includes distinct values for each degree-of-freedom, as summarised in table 3.1.

The noise and quantisation occur in a specific order. Let s^{nom} be the nominal sensor values that are computed in the previous section, m^{n} be the amount of

Feature	Noise	Quantisation
s_x	10 μm	5 μm
s_y	10 μm	12.5 μm
s_z	10 μm	5 μm
s_{rx}	0.07 deg	0.03 deg

Table 3.1: Sensor noise and quantisation amounts

noise and m^{q} be the amount of quantisation. The sensor values s^{nq} with noise and quantisation are computed as

$$s^{\mathrm{nq}} = \mathrm{round}(s^{\mathrm{nom}} + \mathrm{rand}(m^{\mathrm{n}}), m^{\mathrm{q}}), \tag{3.27}$$

where $\mathrm{rand}(m^{\mathrm{n}})$ is a random number between -m^{n} and m^{n} and $\mathrm{round}(s, m^{\mathrm{q}})$ rounds the value of s to the nearest integer multiple of m^{q}.

Sensor tool location

To compute the sensor measurements, the sensor model needs the location ${}^{F}_{S}\mathbf{T}$ of the sensor tool frame. From figure 3.1, it can be computed as

$${}^{F}_{S}\mathbf{T} = {}^{B}_{F}\mathbf{T}^{-1}\,{}^{B}_{N}\mathbf{T} \cdot {}^{N}_{S}\mathbf{T}. \tag{3.28}$$

Transformation ${}^{F}_{S}\mathbf{T}$ contains models of the robot geometry ${}^{B}_{N}\mathbf{T}$ (section 3.5.3) and the sensor tool transformation ${}^{N}_{S}\mathbf{T}$.

3.5.3 Geometric robot models

The Direct Geometric Model (**DGM**) or forward kinematics function of a robot or manipulator is the set of relations that defines the location of the flange or end-effector of the robot or manipulator as a function of its joint coordinates. In this work, the location of the flange is specified as a homogenous transformation matrix and transformation ${}^{B}_{N}\mathbf{T}$ can be expressed as

$${}^{B}_{N}\mathbf{T} = \mathbf{DGM}(\mathbf{q}), \tag{3.29}$$

where $\mathbf{q}$ is the vector of robot joint angles. Equation 3.29 has a single solution, i.e. every vector of joint angles corresponds to one position and orientation of the robot flange or end-effector. Symbolic nominal models for equation 3.29 are given in appendix C.1 for robots from the Stäubli RX series.

The Inverse Geometric Model (**IGM**) or inverse kinematics function of a robot or manipulator is the set of relations that gives the joint variables corresponding to a specified location of the end-effector. In this thesis, the Inverse Geometric Model for the robot arm is used frequently. It can be formulated as

$$\mathbf{q} = \mathbf{IGM}({}^{B}_{N}\mathbf{T}). \tag{3.30}$$

In general, equation 3.30 has multiple solutions and its complexity depends on the geometry of the robot. Equation 3.30 has no solution if ${}^{B}_{N}\mathbf{T}$ is outside the robot workspace. Typically, for six degree-of-freedom robots with only revolute joints, where three joint axes intersect at a point, there are eight solutions for each configuration of the robot shoulder (above, below), elbow (lefty, righty) and wrist (positive, negative) (Khalil and Dombre, 2002). Symbolic nominal expressions for equation 3.30 are given in appendix C.2 for robots from the Stäubli RX series.

In this work, two types of transformations will be distinguished:

- Nominal transformations, denoted by superscript n, e.g. ${}^{B}_{N}\mathbf{T}^{\mathrm{n}}$. These transformations are used by the robot control software, both in cases when real robot and sensor hardware is used, but also when models such as derived in this chapter are used.
- Actual transformations, denoted by superscript a, e.g. ${}^{B}_{N}\mathbf{T}^{\mathrm{a}}$. These transformations are only used for obtaining the simulated sensor measurements and may be (but do not have to be) different from the nominal transformations.

This way, the effect of errors between the nominal and actual robot geometry and sensor tool errors can be simulated. An advantage of this approach is the possibility to distinguish between different effects as robot geometric errors and tool transformation errors can be studied independently, which is not possible on the actual system.

To accurately predict the effect of geometric errors in a sensor-guided robotic laser welding system, the actual geometry of the robot arm should be modelled as well. Several parameters of a geometric model can be adjusted to improve the accuracy of such a robot model, e.g. arm lengths, encoder offset or link angles (Conrad et al., 2000). De Roo (2003) identified three kinds of geometric models for the Stäubli RX90 robot that is used in our laboratory using the Krypton Rodym 6D measurement system (Krypton, 2001). The robot model was described using the standard Denavit-Hartenberg notation (Denavit and Hartenberg, 1955), (Craig, 1986). Three identified models have been obtained, where the errors in the robot tip are attributed to the encoder offsets parameters, arm lengths parameters or link angles parameters respectively. The identified models increase the tip accuracy with about a factor of 2, compared to the tip accuracy of the nominal geometric robot model. These models are used to simulate sensor measurements and study the effect of geometric errors in the robot modelling on the sensor tool calibration (chapter 4) and seam teaching procedures (chapter 5).

3.5.4 Robot controller model

A geometric model of the robot and its controller is shown in figure 3.4.

During movements in task (Cartesian) space, the robot controller has to calculate the transformation ${}^{B}_{N}\mathbf{T}$ for a given tool transformation ${}^{N}_{T}\mathbf{T}$ (where $\mathbf{T}$ can be laser tool or sensor tool) and with respect to a given frame transformation ${}^{B}_{F}\mathbf{T}$ (where $\mathbf{F}$ can be base frame, station frame, product frame or seam frame) from a desired location ${}^{F}_{T}\mathbf{T}$. From the chain of transformations at the robot base frame in figure 3.1 follows that

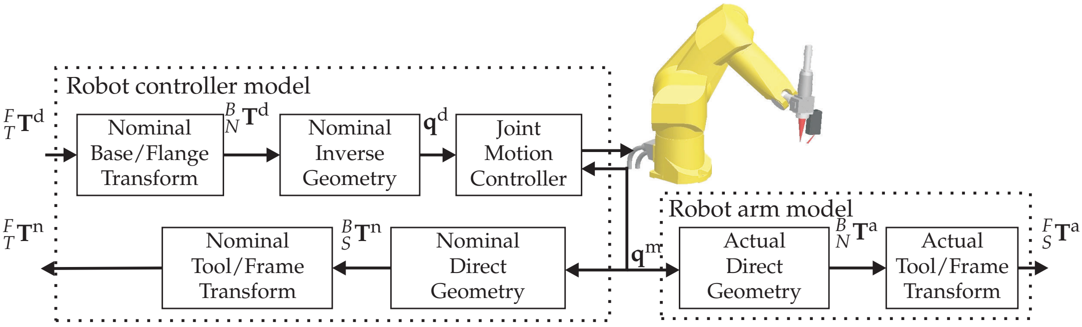
$$ {}^{B}_{N}\mathbf{T} \cdot {}^{N}_{T}\mathbf{T} = {}^{B}_{F}\mathbf{T} \cdot {}^{F}_{T}\mathbf{T}. \tag{3.31} $$

Rewriting equation 3.31 gives

$$ {}^{B}_{N}\mathbf{T} = {}^{B}_{F}\mathbf{T} \cdot {}^{F}_{T}\mathbf{T} \cdot {}^{N}_{T}\mathbf{T}^{-1}. \tag{3.32} $$

Using equation 3.32, transformation ${}^{F}_{T}\mathbf{T}^{\mathrm{d}}$ can be transformed to ${}^{B}_{N}\mathbf{T}^{\mathrm{d}}$, using nominal tool transformation ${}^{N}_{T}\mathbf{T}^{\mathrm{n}}$ and nominal frame transformation ${}^{B}_{F}\mathbf{T}^{\mathrm{n}}$. The nominal Inverse Geometric Model of the robot arm is used to calculate the desired joint angles $\mathbf{q}^{\mathrm{d}}$ of the end-point of the point-to-point movement using equation 3.30.

The obtained joint angles $\mathbf{q}^{\mathrm{d}}$ are the reference input for a joint motion controller, which tracks the measured robot joint angles $\mathbf{q}^{\mathrm{m}}$ equal to the desired joint angles $\mathbf{q}^{\mathrm{d}}$. As long as point-to-point movements are considered, it is not important how this is achieved as long as the measured joint angles $\mathbf{q}^{\mathrm{m}}$ at the end-point correspond to the desired joint angles $\mathbf{q}^{\mathrm{d}}$ at the end-point.



Robot controller model
${}^{F}_{T}\mathbf{T}^{\mathrm{d}}$
Nominal Base/Flange Transform
${}^{B}_{N}\mathbf{T}^{\mathrm{d}}$
Nominal Inverse Geometry
$\mathbf{q}^{\mathrm{d}}$
Joint Motion Controller
${}^{F}_{T}\mathbf{T}^{\mathrm{n}}$
Nominal Tool/Frame Transform
${}^{B}_{S}\mathbf{T}^{\mathrm{n}}$
Nominal Direct Geometry
$\mathbf{q}^{\mathrm{m}}$
Robot arm model
Actual Direct Geometry
${}^{B}_{N}\mathbf{T}^{\mathrm{a}}$
Actual Tool/Frame Transform
${}^{F}_{S}\mathbf{T}^{\mathrm{a}}$

Figure 3.4: Block diagram of the robot and its controller for point-to-point movements

The robot controller can be asked for the current tool location ${}^{F}_{T}\mathbf{T}^{\mathrm{n}}$. The measured joint angles $\mathbf{q}^{\mathrm{m}}$ are then used to predict the location of the robot flange ${}^{B}_{N}\mathbf{T}^{\mathrm{n}}$ with respect to the base frame, using equation 3.29 with the nominal Direct Geometric Model of the robot arm.

Equation 3.31 can be re-arranged to get

$$
{}^{F}_{T}\mathbf{T} = {}^{B}_{F}\mathbf{T}^{-1} \cdot {}^{B}_{N}\mathbf{T} \cdot {}^{N}_{T}\mathbf{T}. \tag{3.33}
$$

The nominal location of the robot flange ${}^{B}_{N}\mathbf{T}^{\mathrm{n}}$ and the known nominal transformations ${}^{N}_{T}\mathbf{T}^{\mathrm{n}}$ and ${}^{B}_{F}\mathbf{T}^{\mathrm{n}}$ are then used to predict ${}^{F}_{T}\mathbf{T}^{\mathrm{n}}$ using equation 3.33.

In the case when sensor measurements are simulated, the sensor model needs the actual location ${}^{F}_{S}\mathbf{T}^{\mathrm{a}}$ of the sensor. This location is calculated from the measured joint angles $\mathbf{q}^{\mathrm{m}}$ in a similar way, but instead of nominal robot models, actual robot models may be used. Furthermore instead of nominal tool and frame transformations, actual tool and frame transformations may be used.

3.6 Discussion

In this chapter an overview of the applied coordinate frames and transformations has been given. The mathematical definitions for describing transformations are given. Models have been derived for simulating the measurements of the seam tracking sensor. This allows the tool calibration and seam teaching procedures that are described in the following chapters to be simulated to investigate the influence of various errors that occur in a sensor-guided robotic laser welding system.

Chapter 4

Tool calibration procedures

4.1 Introduction

In the previous chapter the geometrical modelling of a seam tracking sensor was presented. This sensor measures relative to its coordinate system (**S**), but the measurements are applied in the laser coordinate frame (**L**). Figure 4.1 shows these frames, the robot flange (**N**), a location on the seam trajectory (**G**) and the transformations between them.

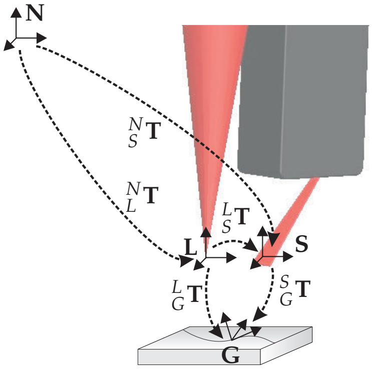


Figure 4.1: Coordinate frames for measuring a seam location with a seam tracking sensor.

According to figure 4.1, the transformation ${}^{L}_{G}\mathbf{T}$ that describes the transformation between laser frame **L** and seam frame **G**, representing the deviation of the laser focal point and the seam trajectory, can be computed as

$$ {}^{L}_{G}\mathbf{T} = {}^{L}_{S}\mathbf{T} \cdot {}^{S}_{G}\mathbf{T}, \tag{4.1} $$

where ${}^{L}_{S}\mathbf{T}$ is a fixed geometrical transformation from the laser frame to the sensor frame and ${}^{S}_{G}\mathbf{T}$ represents a measured location on the seam trajectory relative to the sensor frame **S**. To position the laser spot accurately on the seam, both of these transformations need to be accurately known (< 0.2 mm).

Robot movements can be defined relative to the flange frame **N** or relative to any tool frame **T**, which can be **S** or **L**. When robot movements are defined relative to the sensor or the laser tool frame, the transformations ${}^{N}_{L}\mathbf{T}$ (which represents the transformation from flange to laser frame) and ${}^{N}_{S}\mathbf{T}$ (which represents the transformation from flange to sensor frame) need to be known respectively. According to figure 4.1, one of the transformations ${}^{L}_{S}\mathbf{T}$, ${}^{N}_{L}\mathbf{T}$ and ${}^{N}_{S}\mathbf{T}$ can be computed if the other two are known using

$$ {}^{N}_{S}\mathbf{T} = {}^{N}_{L}\mathbf{T} \cdot {}^{L}_{S}\mathbf{T}. \tag{4.2} $$

This chapter describes three calibration procedures:

- Laser tool calibration to determine the transformation ${}^{N}_{L}\mathbf{T}$.
- Sensor tool calibration to determine the transformation ${}^{N}_{S}\mathbf{T}$.
- Combined calibration to determine the transformation ${}^{L}_{S}\mathbf{T}$.

In the first two calibration procedures a series of robot movements is performed to determine the transformations ${}^{N}_{L}\mathbf{T}$ and ${}^{N}_{S}\mathbf{T}$ respectively. The accuracy of these procedures is influenced by the accuracy of the robot movements. The third procedure (section 4.4) is accomplished by means of a direct measurement that does not involve robot movements.

In literature (section 1.3) the calibration of sensors attached to a robotic manipulator is generally known as the hand-eye calibration or sensor mount registration problem. It was first addressed by Chou and Kamel (1988) although several other approaches for obtaining a solution have been presented by Shiu and Ahmad (1989), Tsai and Lenz (1989), Zhuang and Shiu (1993) and Park and Martin (1994).

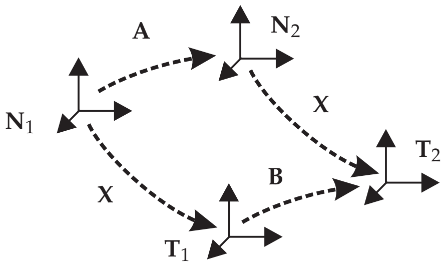


Figure 4.2: The hand-eye calibration problem.

The calibration problem (figure 4.2) is defined by a homogenous matrix equation of the form

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B}, \tag{4.3}$$

where **A** represents a given transformation (movement) of the robot flange frame from location $\mathbf{N}_1$ to location $\mathbf{N}_2$, **B** is a measured transformation in the tool frame from location $\mathbf{T}_1$ to location $\mathbf{T}_2$ and **X** is the unknown transformation from flange frame to tool frame.

In order to solve **X** uniquely, it is necessary to perform at least two arm movements whose axes of rotation are not parallel (Shiu and Ahmad, 1989), to form a system of equations of the form:

$$\mathbf{A}_1 \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B}_1 \tag{4.4a}$$

$$\mathbf{A}_2 \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B}_2. \tag{4.4b}$$

Like Park and Martin (1994) mentioned, there is always noise present in the measured values of **A** and **B**. Therefore a better approach involves making more measurements $\{(\mathbf{A}_1, \mathbf{B}_1), \cdots, (\mathbf{A}_n, \mathbf{B}_n)\}$ and to find an **X** that minimises some error criterion. For $n > 2$ additional equations are added to the system of equations 4.4 in order to obtain an over-determined set of equations. A ″best fit″ for the unknown transformation **X** is then found, e.g. by means of a least squares fit using a pseudo-inverse.

This chapter gives procedures for solving the calibration problem for a laser welding head and a seam tracking sensor that are attached to the robot flange. For sensor tool calibration, the algorithms that are found in literature are used to solve equation 4.4. To achieve a complete measurement of the **B** matrix, a special calibration object and measurement method are proposed. Furthermore, the influence of systematic errors in the **B** matrix due to robot geometric errors is investigated.

For Laser tool calibration, the algorithms that are found in literature can not be directly used. A measurement problem should be solved as the laser is destructive for standard cameras. Furthermore, the resulting measurement does not completely define the **B** matrix. Equation 4.4 will be solved in multiple calibration steps, where parts of this equation are solved in subsequent steps.

4.2 Laser tool calibration

In the case of laser tool calibration, equation 4.3 can be rewritten as

$${}^{N_i}_{N_{i+1}}\mathbf{T} \cdot {}^{N}_{L}\mathbf{T} = {}^{N}_{L}\mathbf{T} \cdot {}^{L_i}_{L_{i+1}}\mathbf{T}, \tag{4.5}$$

where the transformation of the robot flange from location $\mathbf{N}_i$ to location $\mathbf{N}_{i+1}$ equals

$$ {}^{N_i}_{N_{i+1}}\mathbf{T} = {}^{B}_{N_i}\mathbf{T}^{-1} \cdot {}^{B}_{N_{i+1}}\mathbf{T}. \tag{4.6}$$

Transformations ${}^{B}_{N_i}\mathbf{T}$ can be obtained from the robot joint measurements using equation 3.29.

Assuming that the transformation ${}^{L_i}_{R}\mathbf{T}$, which represents some possibly unknown frame $\mathbf{R}$ relative to the laser tool frame $\mathbf{L}$ can be measured, the movement of the laser tool ${}^{L_i}_{L_{i+1}}\mathbf{T}$ from location $\mathbf{L}_i$ to location $\mathbf{L}_{i+1}$ can be expressed as

$$ {}^{L_i}_{L_{i+1}}\mathbf{T} = {}^{L_i}_{R}\mathbf{T} \cdot {}^{L_{i+1}}_{R}\mathbf{T}^{-1}. \tag{4.7}$$

However, it is not straightforward to measure transformation ${}^{L_i}_{R}\mathbf{T}$, as the high power laser focal point is not visible for the human eye and can destroy standard cameras.

In the calibration procedure outlined in the next section, a coaxial camera that is fixed to the welding head (and therefore fixed to frame $\mathbf{L}$) will be used to measure movements of a small spot of light. A problem that needs to be dealt with is that only the 2D position of the light spot on the camera chip can be measured. Consequently the transformation ${}^{L_i}_{R}\mathbf{T}$ can not be determined uniquely from such a measurement. Therefore a different approach will be used, which will be explained in the following sections.

4.2.1 Optical system and frames

For the laser tool calibration procedure, several additional frames will be introduced. These frames are shown in figure 4.3. This figure also schematically shows the mounting of the coaxial CCD camera, which is attached to the welding head.

The following additional frames can be defined:

C: Camera frame. The surface normal of the camera chip coincides with the z-axis of frame **C** and with the optical axis of the laser beam.

R: Reference frame. A calibration object with a mirror, a pinhole with a light emitting diode (LED) and a plate of black anodised aluminum is placed in the robot work space (section 4.2.2, figure 4.4). The origin of frame **R** is located at the pinhole providing a fixed point in robot space that can be measured by the camera.

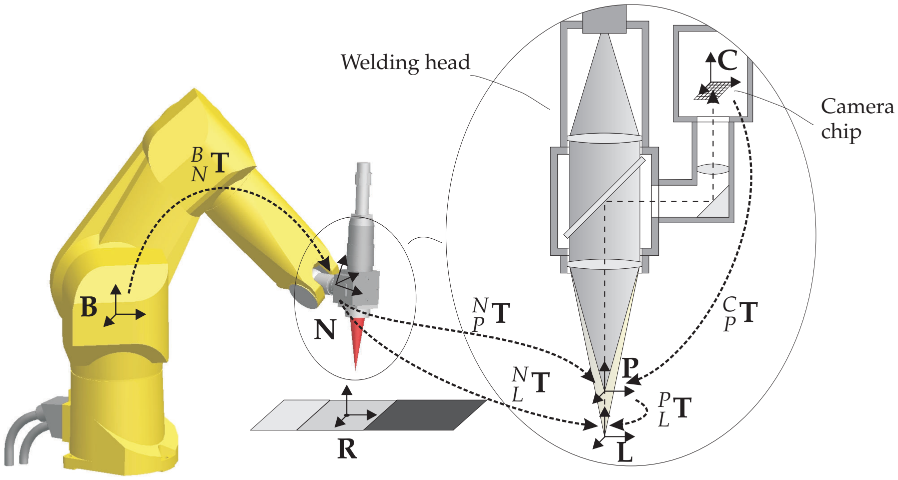


Figure 4.3: Coordinate frames used in the laser tool calibration procedure

P: Pilot laser frame. A low power pilot laser will be applied, using the same optical fiber as the high power Nd:YAG laser. The projection of the laser light spot from this pilot laser on the camera chip will be used for focal measurements.

In the calibration procedure, the pilot laser will be used. The transformation ${}^{N}_{L}\mathbf{T}$ can be split according to

$${}^{N}_{L}\mathbf{T} =^{N}_{P} \mathbf{T} \cdot^{P}_{L} \mathbf{T}. \tag{4.8}$$

The pilot laser and Nd:YAG laser share the same optical axis. The rotation of these frames around the optical axis is assumed to be identical. Therefore, the transformation from pilot frame to laser frame equals

$${}^{P}_{L}\mathbf{T} = \mathbf{Trans}(0, 0, d), \tag{4.9}$$

where $\mathbf{Trans}(0, 0, d)$ is a transformation that represents a pure translation with a distance d along the z-axis. The distance d only depends on the optical system of the welding head. If the optical system does not change, it needs to be determined once. In the calibration procedure the transformation ${}^{N}_{P}\mathbf{T}$ from flange frame to pilot frame will be determined and finally a procedure is outlined to determine distance d (section 4.2.3). Using equations 4.8 and 4.9, the transformation ${}^{N}_{L}\mathbf{T}$ from flange frame to laser frame is easily computed.

The camera may be rotated freely around the optical axis. The orientation between the laser frame or the pilot frame and the camera frame is therefore related as

$$ {}^{L}_{C}\mathbf{R} = {}^{P}_{C}\mathbf{R} = \mathbf{rot}(\mathbf{z}, \varphi), \tag{4.10}$$

where $\mathbf{rot}(\mathbf{z}, \varphi)$ is a rotation matrix that represents a pure rotation around the z-axis with an angle φ.

4.2.2 Calibration object and measurement method

A special calibration object has been developed for this application (figure 4.4). It uses three different surfaces all in the same plane:

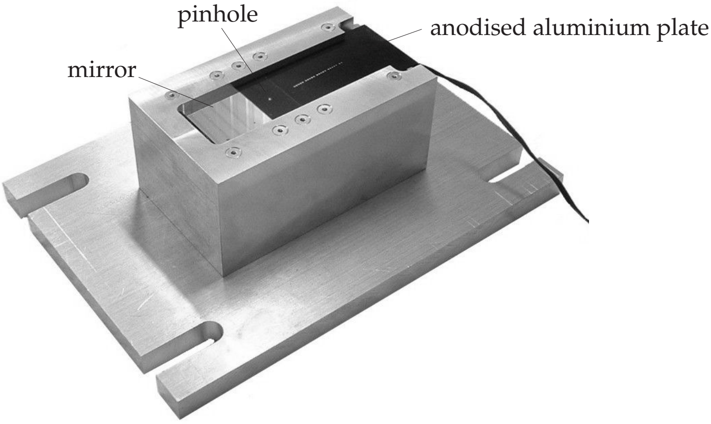


Figure 4.4: Laser tool calibration object

- Mirror surface. The pilot laser will be projected on the mirror to find the focal position of the pilot laser on the camera. Furthermore the surface normal of the mirror will be aligned to the optical axis of the welding head.
- Black surface with a pinhole and a LED light source behind it. The pinhole is a fixed position in the robot workspace, which will be related to the position of the pilot laser on the camera. Different movements of the welding head will be performed to determine the position and orientation of the pilot laser frame relative to the robot flange.
- Piece of anodised aluminium. By making a number of spots at different focal positions, the distance d between the pilot frame and laser frame is determined.

The camera is used to observe the pinhole in the calibration object or a reflection of the pilot laser beam in the mirror. In either case, the camera image shows a light spot in a dark background (figure 4.5). Using basic image processing, the centre of gravity of the light spot and the number of pixels of the light spot are determined. Details of the image processing algorithms that are used can be found in Van Tienhoven (2004).

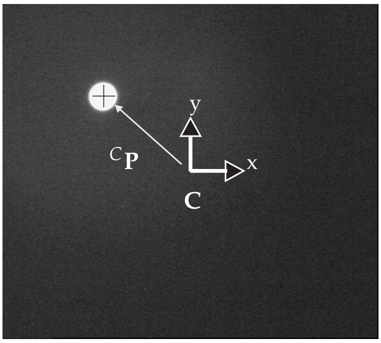


Figure 4.5: Typical camera measurement during the laser tool calibration procedure

The position $^C\mathbf{P}$ in pixels of the centre of this light spot relative to the camera frame **C** is denoted by

$$^C\mathbf{P} = [^C P_x \;\; ^C P_y \;\; 0]^T. \tag{4.11}$$

The camera measures 2 degrees-of-freedom. The orientation of the camera frame and the pilot frame are related through equation 4.10. If frame **P** moves from location $\mathbf{P_i}$ to location $\mathbf{P_{i+1}}$, the camera frame moves from location $\mathbf{C_i}$ to location $\mathbf{C_{i+1}}$. The positions between the frame movements are then related as

$$^{P_i}_{P_{i+1}}\mathbf{P} = \mathbf{rot}(\mathbf{z}, \varphi) \cdot {}^{C_i}_{C_{i+1}}\mathbf{P} \cdot \frac{1}{c}, \tag{4.12}$$

where $\frac{1}{c}$ is a scaling factor to convert from camera pixels to millimetres. The camera pixels may not be square, therefore different ratios c_x and c_y may be used for the camera x-direction and y-direction.

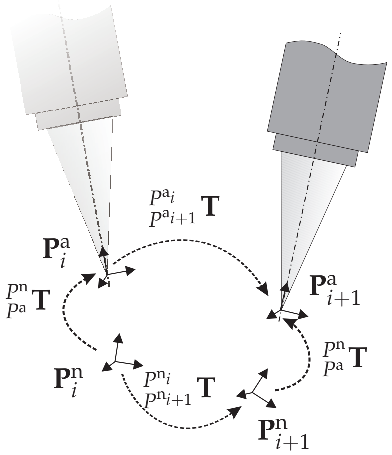
4.2.3 Calibration procedure

To make sure that the observed object stays within the range of the camera, movements are defined relative to a nominal pilot frame $\mathbf{P}^{\mathrm{n}}$, which is an estimation of the actual pilot laser frame $\mathbf{P}^{\mathrm{a}}$. The maximum allowable difference between the nominal frame and the actual frame depends on the field-of-view of the camera. In practice this is large enough to start the calibration procedure with a user-defined nominal tool transformation ${}^{N}_{P^{\mathrm{n}}}\mathbf{T}$ that is known from the dimensions of the welding head. The actual transformation ${}^{N}_{P^{\mathrm{a}}}\mathbf{T}$ that will be determined in the calibration procedure can now be described as

$$ {}^{N}_{P^{\mathrm{a}}}\mathbf{T} = {}^{N}_{P^{\mathrm{n}}}\mathbf{T} \cdot {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{T}, \tag{4.13} $$

where ${}^{N}_{P^{\mathrm{n}}}\mathbf{T}$ is the transformation from robot flange to the nominal pilot frame and ${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{T}$ is the transformation correction between the nominal pilot frame and the actual pilot frame.

Figure 4.6 shows a single robot movement, where frame $\mathbf{P}^{\mathrm{n}}$ is moved from location $\mathbf{P}^{\mathrm{n}}_{i}$ to location $\mathbf{P}^{\mathrm{n}}_{i+1}$. This results in a movement of frame $\mathbf{P}^{\mathrm{a}}$, from location $\mathbf{P}^{\mathrm{a}}_{i}$ to location $\mathbf{P}^{\mathrm{a}}_{i+1}$.



${}^{P^{\mathrm{a}}{}_{i}}_{P^{\mathrm{a}}{}_{i+1}}\mathbf{T}$
$\mathbf{P}^{\mathrm{a}}_{i}$
${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{T}$
$\mathbf{P}^{\mathrm{a}}_{i+1}$
${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{T}$
$\mathbf{P}^{\mathrm{n}}_{i}$
${}^{P^{\mathrm{n}}{}_{i}}_{P^{\mathrm{n}}{}_{i+1}}\mathbf{T}$
$\mathbf{P}^{\mathrm{n}}_{i+1}$

Figure 4.6: A robot movement with the corresponding frame displacements

Using the known transformation ${}^{N}_{P^{\mathrm{n}}}\mathbf{T}$, equation 4.5 can now be written relative to the nominal pilot frame $\mathbf{P}^{\mathrm{n}}$, instead of the flange frame $\mathbf{N}$, as

$$ {}^{P^{\mathrm{n}}_{i}}_{P^{\mathrm{n}}_{i+1}}\mathbf{T} \cdot {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{T} = {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{T} \cdot {}^{P^{\mathrm{a}}_{i}}_{P^{\mathrm{a}}_{i+1}}\mathbf{T}. \tag{4.14} $$

Transformation ${}^{P^{\mathrm{n}}_{i}}_{P^{\mathrm{n}}_{i+1}}\mathbf{T}$ is a prescribed transformation (movement) in the nominal pilot frame and ${}^{P^{\mathrm{a}}_{i}}_{P^{\mathrm{a}}_{i+1}}\mathbf{T}$ is a measured transformation in the actual pilot frame. Transformation ${}^{P^{\mathrm{a}}_{i}}_{P^{\mathrm{a}}_{i+1}}\mathbf{T}$ can not be completely measured because a 2D camera is used. By choosing the prescribed movement ${}^{P^{\mathrm{n}}_{i}}_{P^{\mathrm{n}}_{i+1}}\mathbf{T}$ in a suitable way, parts of equation 4.14 can be solved in subsequent steps. During the calibration procedure, the translation and rotation components of ${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{T}$ are solved separately using the following procedure:

1. *Determination of the pilot frame orientation.* A number of robot movements are made that are defined as pure translations in the nominal pilot frame, while observing the pinhole in the camera image to find the rotation matrix ${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R}$. During this calibration step, the ratio c to convert from camera pixels to millimetres is computed as well as the camera rotation angle φ.
2. *Determination of the surface normal and focal position.* The calibration object is placed arbitrarily in the robot workspace. In this step the optical axis of the welding head and the surface normal of the calibration object are aligned by projecting the pilot laser on the camera image using the mirror and moving the pilot frame along its optical axis. Furthermore, the focal position of the pilot laser on the camera will be determined.
3. *Determination of pilot frame position.* A number of robot movements are made that are defined as pure rotations in the nominal pilot frame, while observing the pinhole in the camera image to find the position vector ${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P}$.

1. Determination of the pilot frame orientation

In the first step of the calibration procedure, the orientation of the actual pilot frame relative to the nominal pilot frame, represented by ${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R}$, is determined. A robot movement that consists of a pure translation in the nominal pilot frame is made. Since no rotation is considered, ${}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{R}$ and ${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{R}$ in equation 4.14 are identity matrices. Substitution of equation 4.12 into equation 4.14 gives

$$\begin{bmatrix} \mathbf{I} & {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R} & {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R} & {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & \mathbf{rot}(\mathbf{z},\varphi) \cdot {}^{C_i}_{C_{i+1}}\mathbf{P} \cdot \frac{1}{c} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.15}$$

Performing the multiplications yields

$$\begin{bmatrix} {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R} & {}^{P^{\mathrm{n}}{}_i}_{P^{\mathrm{n}}{}_{i+1}}\mathbf{P} + {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R} & {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R} \cdot \mathbf{rot}(\mathbf{z},\varphi) \cdot {}^{C_i}_{C_{i+1}}\mathbf{P} \cdot \frac{1}{c} + {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.16}$$

Hence,

$${}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{P} = {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{R} \cdot \mathbf{rot}(\mathbf{z},\varphi) \cdot {}^{C_i}_{C_{i+1}}\mathbf{P} \cdot \frac{1}{c}. \tag{4.17}$$

Using equation 4.10, we can write ${}^{\mathrm{Pn}}_{\mathrm{Pa}}\mathbf{R} \cdot \mathbf{rot}(\mathbf{z}, \varphi)$ as ${}^{\mathrm{Pn}}_{C}\mathbf{R}$ to obtain

$$ {}^{C_i}_{C_{i+1}}\mathbf{P} \cdot \frac{1}{c} = {}^{C}_{\mathrm{Pn}}\mathbf{R} \cdot {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}\mathbf{P}, \tag{4.18} $$

where ${}^{C}_{\mathrm{Pn}}\mathbf{R}$ is the orientation between the nominal pilot frame and the camera frame. ${}^{C_i}_{C_{i+1}}\mathbf{P}$ is measured and ${}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}\mathbf{P}$ is given, ${}^{\mathrm{Pa}}_{\mathrm{Pn}}\mathbf{R}$, c and φ remain to be computed. By making a series of these movements, a linear system can be built for the computation of ${}^{C}_{\mathrm{Pn}}\mathbf{R}$. Rewriting equation 4.18 in its component-form gives

$$ \begin{bmatrix} \frac{1}{c_x} \cdot {}^{C_i}_{C_{i+1}}P_x \\ \frac{1}{c_y} \cdot {}^{C_i}_{C_{i+1}}P_y \\ 0 \end{bmatrix} = \begin{bmatrix} {}^{C}_{\mathrm{Pn}}r_{xx} & {}^{C}_{\mathrm{Pn}}r_{yx} & {}^{C}_{\mathrm{Pn}}r_{zx} \\ {}^{C}_{\mathrm{Pn}}r_{xy} & {}^{C}_{\mathrm{Pn}}r_{yy} & {}^{C}_{\mathrm{Pn}}r_{zy} \\ {}^{C}_{\mathrm{Pn}}r_{xz} & {}^{C}_{\mathrm{Pn}}r_{yz} & {}^{C}_{\mathrm{Pn}}r_{zz} \end{bmatrix} \begin{bmatrix} {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_x \\ {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_y \\ {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_z \end{bmatrix}. \tag{4.19} $$

The first two rows of equation 4.19 give

$$ \left[c_x {}^{C}_{\mathrm{Pn}}r_{xx} \cdot {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_x + c_x {}^{C}_{\mathrm{Pn}}r_{yx} \cdot {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_y + c_x {}^{C}_{\mathrm{Pn}}r_{zx} \cdot {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_z\right] = {}^{C_i}_{C_{i+1}}P_x \tag{4.20a} $$

$$ \left[c_y {}^{C}_{\mathrm{Pn}}r_{xy} \cdot {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_x + c_y {}^{C}_{\mathrm{Pn}}r_{yy} \cdot {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_y + c_y {}^{C}_{\mathrm{Pn}}r_{zy} \cdot {}^{\mathrm{Pn}_i}_{\mathrm{Pn}_{i+1}}P_z\right] = {}^{C_i}_{C_{i+1}}P_y. \tag{4.20b} $$

If n movements are made, a linear system of the form $\mathbf{Ax} = \mathbf{b}$ can be built for the computation of

$$ \mathbf{x} = [c_x {}^{C}_{\mathrm{Pn}}r_{xx} \;\; c_x {}^{C}_{\mathrm{Pn}}r_{yx} \;\; c_x {}^{C}_{\mathrm{Pn}}r_{zx} \;\; c_y {}^{C}_{\mathrm{Pn}}r_{xy} \;\; c_y {}^{C}_{\mathrm{Pn}}r_{yy} \;\; c_y {}^{C}_{\mathrm{Pn}}r_{zy}]^T. \tag{4.21} $$

The system of equations is described as

$$ \begin{bmatrix} {}^{\mathrm{Pn}_0}_{\mathrm{Pn}_1}P_x & {}^{\mathrm{Pn}_0}_{\mathrm{Pn}_1}P_y & {}^{\mathrm{Pn}_0}_{\mathrm{Pn}_1}P_z & 0 & 0 & 0 \\ 0 & 0 & 0 & {}^{\mathrm{Pn}_0}_{\mathrm{Pn}_1}P_x & {}^{\mathrm{Pn}_0}_{\mathrm{Pn}_1}P_y & {}^{\mathrm{Pn}_0}_{\mathrm{Pn}_1}P_z \\ {}^{\mathrm{Pn}_1}_{\mathrm{Pn}_2}P_x & {}^{\mathrm{Pn}_1}_{\mathrm{Pn}_2}P_y & {}^{\mathrm{Pn}_1}_{\mathrm{Pn}_2}P_z & 0 & 0 & 0 \\ 0 & 0 & 0 & {}^{\mathrm{Pn}_1}_{\mathrm{Pn}_2}P_x & {}^{\mathrm{Pn}_1}_{\mathrm{Pn}_2}P_y & {}^{\mathrm{Pn}_1}_{\mathrm{Pn}_2}P_z \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^{\mathrm{Pn}_{n-1}}_{\mathrm{Pn}_n}P_x & {}^{\mathrm{Pn}_{n-1}}_{\mathrm{Pn}_n}P_y & {}^{\mathrm{Pn}_{n-1}}_{\mathrm{Pn}_n}P_z & 0 & 0 & 0 \\ 0 & 0 & 0 & {}^{\mathrm{Pn}_{n-1}}_{\mathrm{Pn}_n}P_x & {}^{\mathrm{Pn}_{n-1}}_{\mathrm{Pn}_n}P_y & {}^{\mathrm{Pn}_{n-1}}_{\mathrm{Pn}_n}P_z \end{bmatrix} \begin{bmatrix} c_x {}^{C}_{\mathrm{Pn}}r_{xx} \\ c_x {}^{C}_{\mathrm{Pn}}r_{yx} \\ c_x {}^{C}_{\mathrm{Pn}}r_{zx} \\ c_y {}^{C}_{\mathrm{Pn}}r_{xy} \\ c_y {}^{C}_{\mathrm{Pn}}r_{yy} \\ c_y {}^{C}_{\mathrm{Pn}}r_{zy} \end{bmatrix} = \begin{bmatrix} {}^{C_0}_{C_1}P_x \\ {}^{C_0}_{C_1}P_y \\ {}^{C_1}_{C_2}P_x \\ {}^{C_1}_{C_2}P_y \\ \vdots \\ {}^{C_{n-1}}_{C_n}P_x \\ {}^{C_{n-1}}_{C_n}P_y \end{bmatrix}. \tag{4.22} $$

A set of predefined movements is used, which makes the system over-determined. A least-squares solution $\mathbf{x}$ of this set of equations can be found from

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b}, \tag{4.23}$$

where $\mathbf{A}^{\dagger}$ is the pseudo-inverse of matrix $\mathbf{A}$ that is defined as

$$\mathbf{A}^{\dagger} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}. \tag{4.24}$$

The vectors $\mathbf{r}_1 = [{}^{C}_{P\mathrm{n}}r_{xx} \;\; {}^{C}_{P\mathrm{n}}r_{yx} \;\; {}^{C}_{P\mathrm{n}}r_{zx}]$ and $\mathbf{r}_2 = [{}^{C}_{P\mathrm{n}}R_{xy} \;\; {}^{C}_{P\mathrm{n}}r_{yy} \;\; {}^{C}_{P\mathrm{n}}r_{zy}]$ are the first and second row of the rotation matrix ${}^{C}_{P\mathrm{n}}\mathbf{R}$ and therefore should be unit vectors. The scaling factors c_x and c_y can be computed from the first and second half of $\mathbf{x}$ as:

$$c_x = ||c_x\mathbf{r}_1|| \tag{4.25a}$$
$$c_y = ||c_y\mathbf{r}_2||. \tag{4.25b}$$

As c_x and c_y are now known, the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are computed as

$$\mathbf{r}_1 = \frac{1}{c_x} \cdot c_x\mathbf{r}_1 \tag{4.26a}$$
$$\mathbf{r}_2 = \frac{1}{c_y} \cdot c_y\mathbf{r}_2 \tag{4.26b}$$

The third row $\mathbf{r}_3 = [{}^{C}_{P\mathrm{n}}r_{xz} \;\; {}^{C}_{P\mathrm{n}}r_{yz} \;\; {}^{C}_{P\mathrm{n}}r_{zz}]$ of rotation matrix ${}^{C}_{P\mathrm{n}}\mathbf{R}$ still has to be computed. Because of the orthonormality condition of the rotation matrix, $\mathbf{r}_3$ is determined by the cross product of

$$\mathbf{r}_3 = \frac{\mathbf{r}_1 \times \mathbf{r}_2}{|\mathbf{r}_1 \times \mathbf{r}_2|}. \tag{4.27}$$

As a result of the least-squares procedure, the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ may not be completely perpendicular. Therefore the cross product of $\mathbf{r}_1$ and $\mathbf{r}_2$ is divided by its length and to achieve orthonormality, $\mathbf{r}_1$ is corrected as

$$\mathbf{r}_1^{c} = \mathbf{r}_2 \times \mathbf{r}_3. \tag{4.28}$$

The orientation of the camera frame relative to the nominal pilot frame is obtained from the vectors $\mathbf{r}_1^{c}$, $\mathbf{r}_2$ and $\mathbf{r}_3$ as

$${}^{P\mathrm{n}}_{C}\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^{c} \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix}. \tag{4.29}$$

The orientation ${}^{N}_{Pa}\mathbf{R}$ between the flange frame and the actual pilot frame will be computed from ${}^{Pn}_{C}\mathbf{R}$. Because the camera frame and the actual pilot frame share the same z-axis (optical axis), ${}^{N}_{Pa}\mathbf{r}_z$ is computed as

$${}^{N}_{Pa}\mathbf{r}_z = {}^{N}_{Pn}\ \mathbf{R} \cdot {}^{Pn}_{C}\ \mathbf{r}_z. \tag{4.30}$$

As the laser beam is rotation-symmetric, the rotation around the z-axis may be chosen arbitrarily if no sensor is attached to the welding head. Section 3.4 defines the orientation of the laser frame if a sensor is attached to the welding head. Assuming the x-axis of the nominal pilot frame is defined in line with this definition, the y-axis of the actual pilot frame will be computed from the x-axis of the nominal pilot frame as

$${}^{N}_{Pa}\mathbf{r}_y = {}^{N}_{Pa}\ \mathbf{r}_z \times {}^{N}_{Pn}\ \mathbf{r}_x. \tag{4.31}$$

To maintain orthonormality, ${}^{N}_{Pa}\mathbf{r}_x$ is computed as

$${}^{N}_{Pa}\mathbf{r}_x = {}^{N}_{Pa}\ \mathbf{r}_y \times {}^{N}_{Pa}\ \mathbf{r}_z. \tag{4.32}$$

The complete orientation of the actual pilot frame relative to the flange frame is now found. Furthermore the scaling factors c_x and c_y for conversion from camera pixels to millimetres are obtained.

2. Determination of the surface normal and focal position

This second step of the calibration procedure can be considered as a preparation for the third step, where the position vector ${}^{Pn}_{Pa}\mathbf{P}$ will be determined. To achieve this, movements of the pinhole in the pilot frame $\mathbf{P}$ as a result of robot movements will be measured. To find the correct position vector, the position of the pinhole on the surface must be related to the focal position of the pilot laser on the camera, which is achieved in this second calibration step.

In the previous step the pinhole was observed by the camera. In this step, the pilot laser will be observed for which purpose the mirror in the calibration object (figure 4.4) is used. To find the focal position of the pilot laser, the welding head is positioned above the mirror. The pilot laser is imaged on the camera via the mirror. The height of the welding head is varied in small steps along the optical axis. At every position, the number of pixels in the pilot spot size is measured in the camera image (figure 4.7).

From the geometry of the laser beam, it is expected that the spot area S can be written as a second order polynomial function of the height variation Δz as

$$S = a \cdot \Delta z^2 + b \cdot \Delta z + c. \tag{4.33}$$

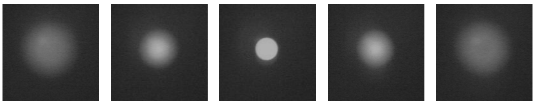


Figure 4.7: Sequence of images with the laser head moving from 3 mm below to 3 mm above the pilot focus

Using a polyfit procedure, the coefficients a, b and c are obtained. The value of Δz^{min} where the polynomial has a minimum (focal position) is found as

$$\Delta z^{\text{min}} = \frac{-b}{2a}. \tag{4.34}$$

The welding head will be moved a distance Δz^{min} along the optical axis to the focal position. In the third step, the pinhole will be used again. Hence the laser tool has to be moved from the pinhole to the mirror and back. It is essential that these movements are parallel to the unknown surface of the calibration object. Therefore the optical axis of the laser tool and the surface normal are aligned as well in this step.

During the height variation, the variations ${}^{C}P_x$ and ${}^{C}P_y$ in x and y-direction in the camera image are also measured. If the optical axis was parallel to the surface normal of the mirror no variation in these directions would be expected. The variations ${}^{C}P_x$ and ${}^{C}P_y$ are expected to show a linear behaviour. They can be expressed as

$${}^{C}P_x = a_x \cdot \Delta z + b_x, \tag{4.35}$$

and

$${}^{C}P_y = a_y \cdot \Delta z + b_y. \tag{4.36}$$

The coefficients are obtained from a polyfit procedure. To position the optical axis of the welding head parallel to the surface normal of the mirror, it needs to rotate an angle γ around its tool x-axis and an angle β around its tool y-axis. These angles are computed as

$$2\gamma = \arctan(a_y), \tag{4.37}$$

and

$$2\beta = \arctan(a_x). \tag{4.38}$$

The factor 2 is caused by the fact that a mirror is used.

3. Determination of the pilot frame position

The orientation of the pilot frame relative to the robot flange is known and is aligned with the surface normal of the calibration object. The position of the focal point relative to the surface of the calibration object has been determined, but this position of the focal point is not known relative to the robot flange yet. This pilot frame position calibration is the subject of this third step of the calibration procedure. A set of given pure rotations relative to the current nominal pilot frame $\mathbf{P}^{\mathrm{n}}$ is made. As a result of the first calibration step, the orientation of the nominal pilot frame frame $\mathbf{P}^{\mathrm{n}}$ has become equal to the orientation of the actual pilot frame $\mathbf{P}^{\mathrm{a}}$. Hence, for the pure rotation relative to $\mathbf{P}^{\mathrm{n}}$, equation 4.14 becomes

$$\begin{bmatrix} {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{R} & {}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{4.39}$$

where ${}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{R}$ is a predefined rotation, ${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P}$ is a measured tool movement and ${}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P}$ is the position vector from nominal to actual pilot frame that needs to be computed. Equation 4.39 can be simplified to

$$\begin{bmatrix} {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{R} & {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{R} \cdot {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{R} & {}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P} + {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.40}$$

Hence

$${}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{R} \cdot {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} = {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} + {}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P}. \tag{4.41}$$

Solving ${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P}$ from equation 4.41 yields:

$${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P} = {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}\mathbf{R} \cdot {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P} - {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}\mathbf{P}. \tag{4.42}$$

To express the measured tool movement ${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P}$, it is measured in camera coordinates and then transformed to the pilot frame using equation 4.12. Only the x and y-coordinates of ${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}\mathbf{P}$ can be measured in the camera image so two equations can be derived from equation 4.42:

$${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}P_x = ({}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}r_{xx} - 1)\ {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}P_x + {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}r_{yx}\ {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}P_y + {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}r_{zx}\ {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}P_z, \tag{4.43a}$$

$${}^{P^{\mathrm{a}}_i}_{P^{\mathrm{a}}_{i+1}}P_y = ({}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}r_{xy}\ {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}P_x + ({}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}r_{yy} - 1)\ {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}P_y + {}^{P^{\mathrm{n}}_i}_{P^{\mathrm{n}}_{i+1}}r_{zy}\ {}^{P^{\mathrm{n}}}_{P^{\mathrm{a}}}P_z. \tag{4.43b}$$

With n rotations a linear system of the form $\mathbf{Ax} = \mathbf{b}$ can be built for the computation of $\mathbf{x} ={}^{P^\mathrm{n}}_{P^\mathrm{a}} \mathbf{P}$:

$$\begin{bmatrix} {}^{P^\mathrm{n}_0}_{P^\mathrm{n}_1}r_{xx} - 1 & {}^{P^\mathrm{n}_0}_{P^\mathrm{n}_1}r_{yx} & {}^{P^\mathrm{n}_0}_{P^\mathrm{n}_1}r_{zx} \\ {}^{P^\mathrm{n}_0}_{P^\mathrm{n}_1}r_{xy} & {}^{P^\mathrm{n}_0}_{P^\mathrm{n}_1}r_{yy} - 1 & {}^{P^\mathrm{n}_0}_{P^\mathrm{n}_1}r_{zy} \\ \vdots & \vdots & \vdots \\ {}^{P^\mathrm{n}_{n-1}}_{P^\mathrm{n}_n}r_{xx} - 1 & {}^{P^\mathrm{n}_{n-1}}_{P^\mathrm{n}_n}r_{yx} & {}^{P^\mathrm{n}_{n-1}}_{P^\mathrm{n}_n}r_{zx} \\ {}^{P^\mathrm{n}_{n-1}}_{P^\mathrm{n}_n}r_{xy} & {}^{P^\mathrm{n}_{n-1}}_{P^\mathrm{n}_n}r_{yy} - 1 & {}^{P^\mathrm{n}_{n-1}}_{P^\mathrm{n}_n}r_{zy} \end{bmatrix} \cdot \begin{bmatrix} {}^{P^\mathrm{n}}_{P^\mathrm{a}}P_x \\ {}^{P^\mathrm{n}}_{P^\mathrm{a}}P_y \\ {}^{P^\mathrm{n}}_{P^\mathrm{a}}P_z \end{bmatrix} = \begin{bmatrix} {}^{P^\mathrm{a}_0}_{P^\mathrm{a}_1}P_x \\ {}^{P^\mathrm{a}_0}_{P^\mathrm{a}_1}P_y \\ \vdots \\ {}^{P^\mathrm{a}_{n-1}}_{P^\mathrm{a}_n}P_x \\ {}^{P^\mathrm{a}_{n-1}}_{P^\mathrm{a}_n}P_x \end{bmatrix}. \tag{4.44}$$

Using $n > 3$ makes the linear system in equation 4.44 over-determined. The least squares best solution is found in a similar way as in the orientation calibration step. The transformation from robot flange to actual pilot frame can now be computed as

$${}^{N}_{P^\mathrm{a}}\mathbf{T} ={}^{N}_{P^\mathrm{n}} \mathbf{T} \cdot{}^{P^\mathrm{n}}_{P^\mathrm{a}} \mathbf{T}, \tag{4.45}$$

with

$${}^{P^\mathrm{n}}_{P^\mathrm{a}}\mathbf{T} = \begin{bmatrix} \mathbf{I} & {}^{P^\mathrm{n}}_{P^\mathrm{a}}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.46}$$

In the experiments, eight pre-defined rotations are used. These rotations are equally spaced in eight directions. To get a considerable change of the pinhole in the camera image, large rotations are desired. The optical fiber limits these rotations to about 20 degrees.

4. Measuring Pilot-YAG focal distance

With the calibration steps one to three, the transformation ${}^{N}_{P^\mathrm{a}}\mathbf{T}$ from the robot flange to the actual pilot frame can be determined. The final step is to measure the difference d in focal height between the pilot frame as it is focussed on the camera chip and the Nd:YAG frame as it is focussed on the surface. This difference is fixed in the tool geometry and is independent of the way the welding head is attached to the robot. It is therefore sufficient to measure it once as long as the optical path of the welding head and coaxial camera does not change. This is done by making spot welds on a plate of anodised aluminium attached to the calibration object (in the same plane as the mirror) and determining the height corresponding to the smallest spot relative to the pilot focal position. This is the only step that involves operating the Nd:YAG laser. Once the distance d is known, the transformation ${}^{N}_{L}\mathbf{T}$ of the laser tool frame relative to the flange frame can be computed using equations 4.8 and 4.9 and the calibrated transformation ${}^{N}_{P\mathrm{a}}\mathbf{T}$, which completes the calibration procedure.

4.2.4 Experimental results

In this section, the experimental results that are obtained with the laser tool calibration procedure are presented. A welding head with a 200 mm focussing lens is attached to the robot flange. From the dimensions of this welding head a nominal tool transformation ${}^{N}_{L\mathrm{n}}\mathbf{T}$ is expressed in Euler RxRyRz notation as $[262\ \ 0\ \ 75\ \ \ 0\ \ -90\ \ 90]$.

Orientation

In the first calibration step, the orientational part of ${}^{N}_{P}\mathbf{T}$ is determined. During this step, the camera scaling factors c_x and c_y and the camera orientation φ are computed as well. Using the theory that is described in section 4.2.3, an orientation correction between the nominal tool orientation and the actual tool orientation is calculated. Using this correction, the nominal tool orientation is updated and the orientation step will be repeated to show its convergence. These orientation corrections are shown in figure 4.8.

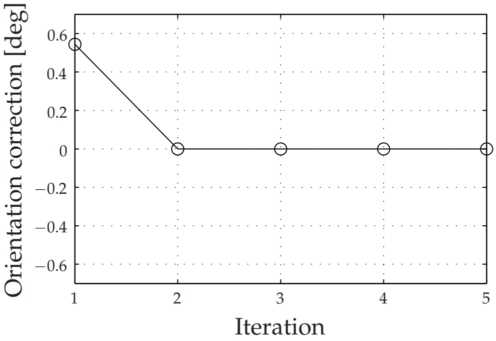


Figure 4.8: Orientation correction of the laser tool calibration procedure

The orientation step of the calibration procedure converges very well. After one iteration, the rotation ${}^{N}_{P}\mathbf{R}$ does not change noticeably anymore. The orientation correction between the nominal tool orientation and the actual tool orientation is around 0.5 degrees, which shows that the orientation of the welding head was manufactured quite well. The resulting values are shown in table 4.1.

The camera scaling factors c_x and c_y are almost equal, which shows that the pixels from the CCD camera are almost completely square. The camera is rotated about 90 degrees relative to the pilot and Nd:YAG laser tool frame.

Variable	Description	Value
c_x	Camera X-Scaling	59.8 [pixels/mm]
c_y	Camera Y-Scaling	59.6 [pixels/mm]
φ	Camera orientation	-90.5 [deg]

Table 4.1: Obtained experimental results from the orientation step of the laser tool calibration procedure

Focal position and surface normal

During the second calibration step, the focal position of the pilot laser on the camera and the surface normal of the mirror are determined. First of all, the spot sizes are measured over a considerable height range (10 mm) of the welding head, to make sure that the minimum is within the measurement range. The measured spot sizes as a result of this height variation are plotted in figure 4.9.

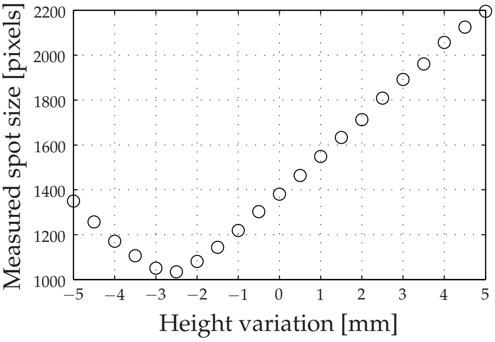


Figure 4.9: First step: Measured spot size in pixels as a function of the height variation, which is used to globally find the focal position of the pilot laser on the camera

It can be observed that further away from the minimum spot size, the spot sizes behave linearly instead of the expected quadratic behavior. It is expected that this is caused by the image processing algorithm in combination with the pilot laser being out of focus. Instead of using the polyfit procedure, the global minimum has been found by simply choosing the smallest value (-2.5 mm in this case).

The welding head will be moved to this position and the measurement procedure is repeated with a smaller height range (2 mm) around the focal position. The measured spot sizes are shown in figure 4.10.

Because the measurements are obtained closely around the focal position, a polyfit procedure will be used to accurately find the minimum. To include the effect of possible non-symmetries, a third order polynomial is used. The welding head is then positioned such that the pilot laser is in focus above the mirror. This procedure is repeated five times, the corrections are plotted in figure 4.11.

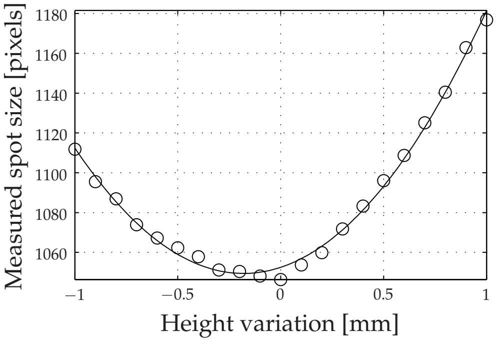


Figure 4.10: Remaining steps: Measured spot size in pixels as a function of a smaller height variation, which is used to accurately find the focal position of the pilot laser on the camera using a polyfit procedure

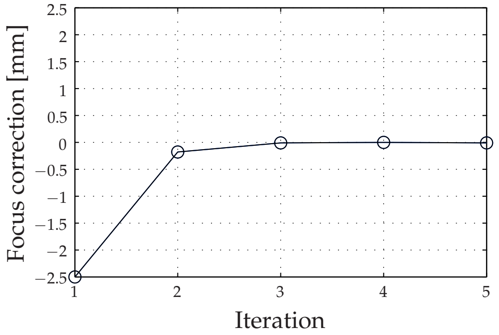


Figure 4.11: Focus correction after each iteration.

The procedure converges quickly. After the first polyfit step (iteration 2), no significant corrections are found anymore.

Besides the spot sizes, the x and y deviations of the pilot laser on the camera are measured as well to be able to find the surface normal of the mirror. These deviations are shown in figure 4.12.

The welding head needs to rotate in such a way that the optical axis becomes parallel to the surface normal of the mirror. These rotations are found to be -0.34 degrees around the x-axis and -0.18 degrees around the y-axis.

The convergence of the surface normal corrections is shown in figure 4.13. After the second iteration, the corrections become insignificant ($<$ 0.01 deg), which corresponds to a focal deviation of 5 μm, if the welding head is moved a distance of 30 mm perpendicular to the surface normal.

Because the optical axis of the welding head is now perpendicular to the mirror surface (and also to the other surfaces), the welding head can be moved from the

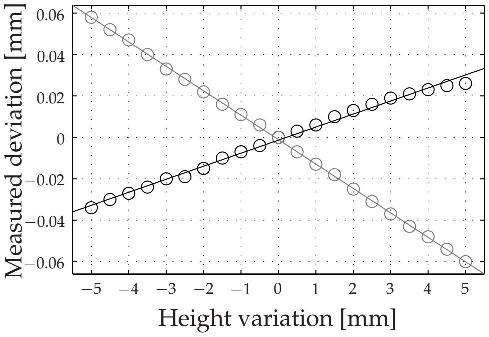


Figure 4.12: Measured camera deviation as a function of the height variation, which is used to find the surface normal of the mirror (— X, — Y)

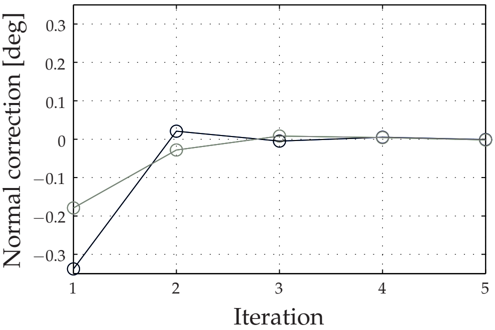


Figure 4.13: Focus normal correction after each iteration (— rx, — ry)

mirror back to the pinhole while the focal position of the pilot laser relative to the surface does not change.

Position

The third step in the calibration procedure is to find the position vector ${}^{N}_{P}\mathbf{P}$. An iterative procedure is used to update the position vector to see if it converges. The position corrections are shown in figure 4.14.

Two iteration steps are needed for the procedure to converge. In the first step, a large deviation of more than 2 mm is found in the tool y-direction. In the second step, the remaining 0.5 mm deviation is determined.

The laser tool calibration is now complete. The resulting tool transformation in Euler RxRyRz notation is: [262.34 -0.60 71.89 mm -20.69 -89.52 69.31 deg]. The actual orientations are quite close. The large deviation of the Euler rotation values is caused by the fact that the rotation around the y-axis is close to -90 degrees.

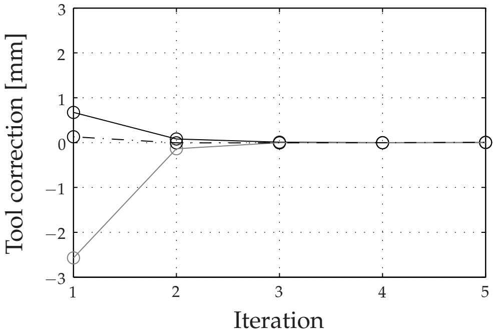


Figure 4.14: Position correction of the laser tool calibration procedure (— X, — Y, -·- Z)

Repeatability

The accuracy of the laser tool calibration procedure will be judged by comparing the results from a series of experiments with each other. A repeatability experiment has been performed, where ten laser tool calibrations have been performed at one location in the robot workspace.

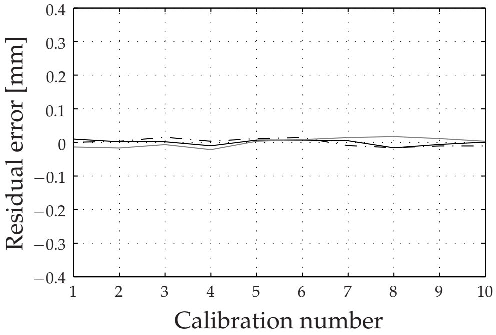


Figure 4.15: Repeatability experiment of the laser tool calibration procedure (— X, — Y, -·- Z)

Figure 4.15 shows that the laser tool calibration procedure reproduces very well as the deviations between the calibrations is about 30 μm, which is in the same order as the robot repeatability.

Because the welding head is geometrically fixed to the robot flange, the same transformation should be found if the calibration procedure is performed at a different location in the robot workspace. Two repeatability experiments are performed, where the calibration object is rotated about 45 degrees on the work table in every experiment to make sure the robot joint positions have significantly changed between the experiments.

The result of the repeatability of the procedure was similar as in figure 4.15. The average tool transformations at the three calibration locations are given in table 4.2. A considerable deviation of more than 1 mm can be observed between the positions. Although the repeatability of the procedure was good, the average tool transformations that are found show a remarkable difference. By observing equation 4.47, these differences can be caused by errors in the robot positioning (errors in A-matrix of equation 4.3) or measurement errors (errors in B-matrix of equation 4.3). It is unlikely that the differences are caused by measurement errors, as the positions and orientations of the welding head relative to the calibration object are similar in all three calibration runs. Therefore the differences must be caused by robot geometric errors.

Experiment	Average tool transformation (Euler RxRyRz)
1	[262.121 -0.723 72.295 mm -47.2 -89.6 42.8 deg]
2	[262.351 0.331 73.456 mm -17.0 -89.8 73.0 deg]
3	[262.341 -0.337 73.854 mm -19.0 -89.7 71.0 deg]

Table 4.2: Average tool transformations of three series of tool transformations where the calibration object was located differently in the robot workspace

In section 4.3.5, simulation results are presented for the sensor tool calibration procedure. The sensor tool calibration procedure is similar to the laser tool calibration procedure in the sense that also the equation 4.3 will be solved. In these simulations, the effect of robot geometric errors on the calibration procedure is taken into account. These simulations give an explanation for the inaccuracy of the sensor tool calibration procedure, which is also valid for the laser tool calibration procedure.

4.3 Sensor tool calibration

4.3.1 Introduction

The optical seam tracking sensor that is used in this work can measure four degrees-of-freedom for typical seam trajectories, namely three translations and one rotation angle (section 3.5.2). The matrix **B** in equation 4.3 contains the sensor measurements and is therefore not completely defined. For the laser tool calibration procedure in section 4.2, a similar difficulty occurred, which has been overcome by combining several calibration steps. For sensor calibration, a different approach is possible as a complete six degree-of-freedom transformation (3 translations and 3 rotations) can be measured, where the sensor is positioned above a special calibration object (section 4.3.2).

During the calibration procedure, the sensor will be positioned in different poses (position and orientation) relative to this calibration object. An initially unknown coordinate frame **E** is attached to the calibration object. Each sensor pose is denoted by a coordinate frame $\mathbf{S}_i$, where i is the pose number. The robot flange pose is denoted by frame $\mathbf{N}_i$.

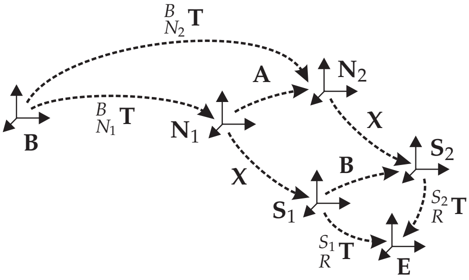


Figure 4.16: Frames used in the sensor calibration procedure

For the case of sensor tool calibration, equation 4.3 can be written as

$${}^{N_i}_{N_{i+1}}\mathbf{T} \cdot {}^{N}_{S}\mathbf{T} = {}^{N}_{S}\mathbf{T} \cdot {}^{S_i}_{S_{i+1}}\mathbf{T}, \tag{4.47}$$

where

$${}^{N_i}_{N_{i+1}}\mathbf{T} = {}^{B}_{N_i}\mathbf{T}^{-1} \cdot {}^{B}_{N_{i+1}}\mathbf{T}, \tag{4.48}$$

and

$${}^{S_i}_{S_{i+1}}\mathbf{T} = {}^{S_i}_{E}\mathbf{T} \cdot {}^{S_{i+1}}_{E}\mathbf{T}^{-1}. \tag{4.49}$$

Transformations ${}^{B}_{N_i}\mathbf{T}$ can be obtained from the robot joint measurements. As long as the calibration object is within the measurement range of the sensor, the transformations ${}^{S_i}_{E}\mathbf{T}$ and ${}^{S_{i+1}}_{E}\mathbf{T}$ can be measured using a measurement method, which is explained in section 4.3.2. The calibration object may be placed arbitrarily in the robot workspace and therefore the absolute location of the calibration object is unknown. By positioning the sensor in different poses above this calibration object, the relative movement of the sensor can be measured according to equation 4.49.

4.3.2 Calibration object and measurement method

To be able to compute the sensor pose relative to a calibration object from a single sensor measurement, Huissoon (2002) proposes to add edges to such a calibration object. He describes a calibration object with three edges and a mathematical approach for computing the sensor pose relative to the calibration object. In this thesis a calibration object with two edges (figure 4.17) is used, as this is sufficient (and even better) to determine the complete sensor pose. It will be shown that this calibration object offers improved accuracy and noise suppression over Huissoon's object . A more complex non-linear matrix equation has to be solved however, which Huissoon avoids by using an additional edge feature.

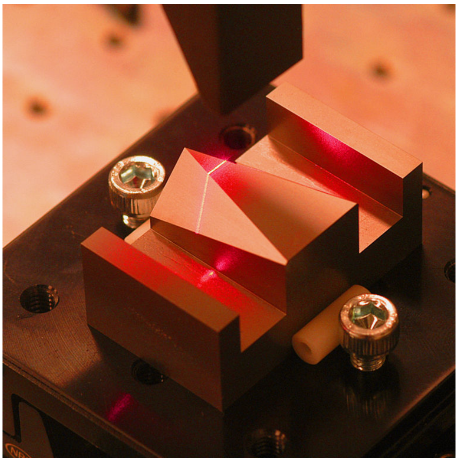


Figure 4.17: Sensor positioned above the calibration object

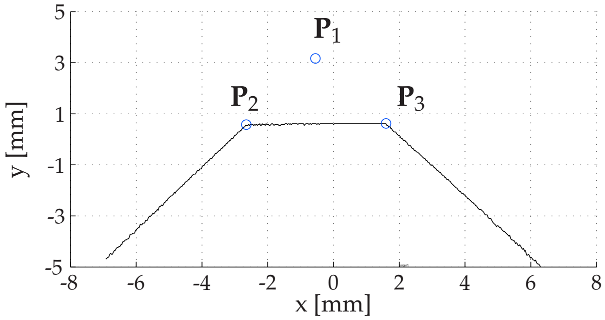


Figure 4.18: Typical sensor profile on the calibration object

The two upper edges of the calibration object give three lines in a sensor image (figure 4.18). From the lines in the sensor image, the intersections of the edges of the calibration object with the laser plane can be computed. This is done by fitting lines through the sensor data and computing their intersections $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ as indicated in figure 4.18. Note that $\mathbf{P}_1$ is a virtual intersection that is constructed by extrapolating the outer line parts. These intersections are 3D vectors in the sensor frame, where two coordinates are measured in the CMOS image and the third coordinate is computed with equation 3.11 using the known diode angle of the sensor. Hence, there are a total number of nine parameters, where six of them are independent. Every sensor pose yields a unique combination of these parameters. It will be shown how to compute the sensor pose from the intersections.

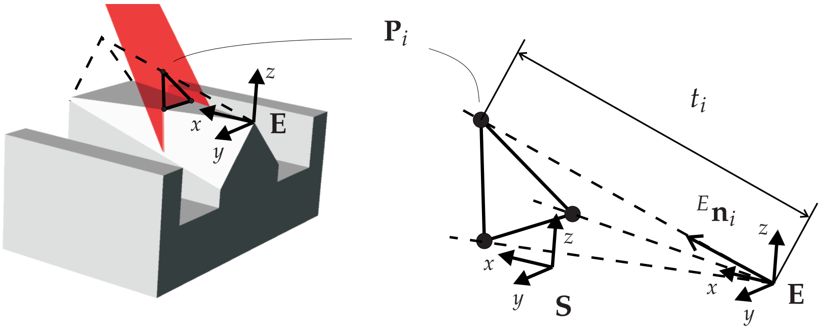


Figure 4.19: Calibration object with the description of an edge relative to frame **E***, which has its origin at the intersection of the three edges.*

The intersections are measured by the sensor relative to its coordinate frame **S** as ${}^S\mathbf{P}_i$, where i is the intersection number. Figure 4.19 shows the diode plane of the sensor relative to the calibration object and the location of frame **E**, which is fixed to the calibration object. The intersections ${}^S\mathbf{P}_i$ relative to the sensor frame and ${}^E\mathbf{P}_i$ relative to the calibration object frame are related as

$$\begin{bmatrix} {}^E\mathbf{P}_i \\ 1 \end{bmatrix} =^E_S \mathbf{T} \cdot \begin{bmatrix} {}^S\mathbf{P}_i \\ 1 \end{bmatrix}, \tag{4.50}$$

where ${}^E_S\mathbf{T}$ is the transformation that determines the location of the sensor relative to the calibration object. The intersections ${}^S\mathbf{P}_i$ are measured by the sensor and ${}^E\mathbf{P}_i$ can be written as

$${}^E\mathbf{P}_i = t_i \cdot {}^E\mathbf{n}_i, \tag{4.51}$$

where ${}^E\mathbf{n}_i$ is a unit vector along edge i of the calibration object that is known from its geometry and t_i indicates the distance between intersection ${}^E\mathbf{P}_i$ and the origin of frame **E**.

Substitution of equation 4.51 in equation 4.50 and expressing the transformation ${}^E_S\mathbf{T}$ in its rotational (${}^E_S\mathbf{R}$) and translational (${}^E_S\mathbf{P}$) parts yields

$$t_i \cdot^E \mathbf{n}_i =^E_S \mathbf{R} \cdot^S \mathbf{P}_i +^E_S \mathbf{P}. \tag{4.52}$$

The unknown lengths t_i in equation 4.52 will now be determined. Once these lengths are known, the unknown components ${}^E_S\mathbf{R}$ and ${}^E_S\mathbf{P}$ of transformation ${}^E_S\mathbf{T}$

can be solved. As shown in figure 4.19 the three edge features form a pyramid and the intersections are the corners of a triangle that fits into this pyramid.

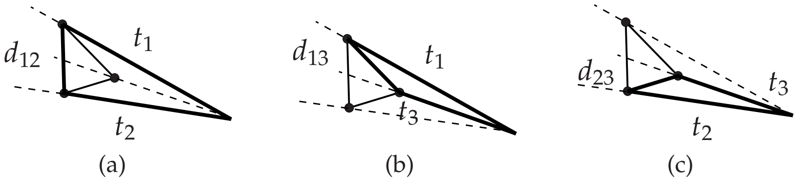


Figure 4.20: triangles for computation of t_i

In order to compute the lengths t_i, a quadratic system of three cosine formulas can be written to relate the scalars t_i to the length of the sides of the triangles shown in figure 4.20:

$$t_i^2 + t_j^2 - 2t_i t_j \mathbf{n}_i \cdot \mathbf{n}_j - d_{ij} = 0, \qquad (i,j) = \{(1,2),(1,3),(2,3)\} \tag{4.53}$$

where terms d_{ij} are the distances between the intersections $\mathbf{P}_i$. Since the distances d_{ij} are independent of the chosen frame, they can be calculated directly from the measurements $\mathbf{P}_i$ in the sensor frame. In this way a set of three quadratic equations has been found for the computation of t_i. The solution of the set of equations is found with a standard Newton-Raphson iteration scheme. Because the set of equations is quadratic, the Jacobian matrix necessary for the Newton-Raphson iterations is determined analytically by differentiating equation 4.53 to t_i. Using the geometry of the reference shape and an approximate estimation of the sensor pose a good starting estimate for t_i is found for the numerical iterations. This iteration scheme quickly converges to an accurate solution for t_i, so the remaining unknowns in equation 4.52 are the components ${}^E_S\mathbf{R}$ and ${}^E_S\mathbf{P}$ of transformation ${}^E_S\mathbf{T}$.

To find ${}^E_S\mathbf{R}$ and ${}^E_S\mathbf{P}$, another frame $\mathbf{Z}$ is introduced, which has its origin in the top intersection $\mathbf{P}_1$, its y-axis is parallel to the sensor frame $\mathbf{S}$ and its x-axis is normal to the laser diode plane of the sensor (figure 4.21).

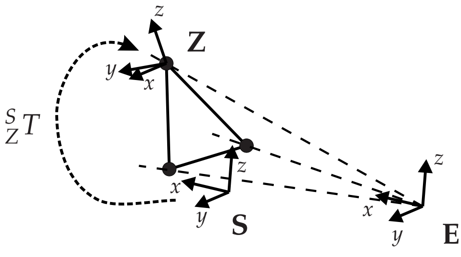


Figure 4.21: New frame $\mathbf{Z}$

This new frame **Z** can be expressed relative to the sensor frame **S** as

$$ {}^S_Z\mathbf{T} = \begin{bmatrix} \mathbf{rot}(\mathbf{y},\alpha) & \mathbf{P}_1 \\ \mathbf{0} & 1 \end{bmatrix}, \tag{4.54}$$

where $\mathbf{rot}(\mathbf{y},\alpha)$ is the rotation matrix that corresponds to a pure rotation around the sensor y-axis with the diode angle α. Equation 4.52 can be rewritten using frame **Z** instead of frame **S**, which gives

$$ t_i \cdot {}^E\mathbf{n}_i = {}^E_Z\mathbf{R} \cdot {}^Z\mathbf{P}_i + {}^E_Z\mathbf{P}. \tag{4.55}$$

The intersections ${}^S\mathbf{P}_i$ relative to the sensor frame and ${}^Z\mathbf{P}_i$ are related as

$$ \begin{bmatrix} {}^Z\mathbf{P}_i \\ 1 \end{bmatrix} = {}^Z_S\mathbf{T} \cdot \begin{bmatrix} {}^S\mathbf{P}_i \\ 1 \end{bmatrix}, \tag{4.56}$$

where ${}^Z\mathbf{P}_i$ are the intersections relative to frame **Z**. This results in ${}^Z\mathbf{P}_1 = [0\ \ 0\ \ 0]^T$, ${}^Z\mathbf{P}_2 = [0\ \ {}^ZP_{y,2}\ \ {}^ZP_{z,2}]^T$ and ${}^Z\mathbf{P}_3 = [0\ \ {}^ZP_{y,3}\ \ {}^ZP_{z,3}]^T$. Substituting ${}^Z\mathbf{P}_1$ into equation 4.55 gives

$$ t_1 \cdot {}^E\mathbf{n}_1 = {}^E_Z\mathbf{P}, \tag{4.57}$$

so that ${}^E_Z\mathbf{P}$ is known. Substituting ${}^Z\mathbf{P}_2$ and ${}^Z\mathbf{P}_3$ into equation 4.55 and writing the components of ${}^E_Z\mathbf{R}$ gives

$$ t_i \cdot {}^E\mathbf{n}_i - {}^E_Z\mathbf{P} = \begin{bmatrix} {}^E_Zr_{yx} \cdot {}^ZP_{y,i} + {}^E_Zr_{zx} \cdot {}^ZP_{z,i} \\ {}^E_Zr_{yy} \cdot {}^ZP_{y,i} + {}^E_Zr_{zy} \cdot {}^ZP_{z,i} \\ {}^E_Zr_{yz} \cdot {}^ZP_{y,i} + {}^E_Zr_{zz} \cdot {}^ZP_{z,i} \end{bmatrix}. \tag{4.58}$$

Equation 4.58 is valid for $i = 2, 3$, therefore a linear system in the form $\mathbf{Ax} = \mathbf{b}$ can be constructed for the computation of two of the three columns of the rotation matrix ${}^E_Z\mathbf{R}$ describing the orientation of frame **Z** relative to frame **E**:

$$ \begin{bmatrix} {}^ZP_{y,1} & {}^ZP_{z,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & {}^ZP_{y,1} & {}^ZP_{z,1} & 0 & 0 \\ 0 & 0 & 0 & 0 & {}^ZP_{y,1} & {}^ZP_{z,1} \\ {}^ZP_{y,2} & {}^ZP_{z,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & {}^ZP_{y,2} & {}^ZP_{z,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & {}^ZP_{y,3} & {}^ZP_{z,3} \end{bmatrix} \cdot \begin{bmatrix} {}^E_Zr_{yx} \\ {}^E_Zr_{zx} \\ {}^E_Zr_{yy} \\ {}^E_Zr_{zy} \\ {}^E_Zr_{yz} \\ {}^E_Zr_{zz} \end{bmatrix} = \begin{bmatrix} t_2 \cdot {}^En_{x,2} - {}^E_ZP_x \\ t_2 \cdot {}^En_{y,2} - {}^E_ZP_y \\ t_2 \cdot {}^En_{z,2} - {}^E_ZP_z \\ t_3 \cdot {}^En_{x,3} - {}^E_ZP_x \\ t_3 \cdot {}^En_{y,3} - {}^E_ZP_y \\ t_3 \cdot {}^En_{z,3} - {}^E_ZP_z \end{bmatrix}. \tag{4.59}$$

Because the vectors ${}^{E}_{Z}\mathbf{r}_y$ and ${}^{E}_{Z}\mathbf{r}_z$ represent the second and third column of the rotation matrix ${}^{E}_{Z}\mathbf{R}$, they are normalised. The first column of ${}^{E}_{Z}\mathbf{R}$ is equal to the cross product of the other two:

$$
{}^{E}_{Z}\mathbf{r}_x = \frac{{}^{E}_{Z}\mathbf{r}_y \times {}^{E}_{Z}\mathbf{r}_z}{|{}^{E}_{Z}\mathbf{r}_y \times {}^{E}_{Z}\mathbf{r}_z|}. \tag{4.60}
$$

The vector ${}^{E}_{Z}\mathbf{r}_z$ is corrected to maintain orthonormality as

$$
{}^{E}_{Z}\mathbf{r}_z ={}^{E}_{Z} \mathbf{r}_x \times{}^{E}_{Z} \mathbf{r}_y. \tag{4.61}
$$

The transformation ${}^{E}_{Z}\mathbf{T}$ is now known, the transformation ${}^{E}_{S}\mathbf{T}$ can be computed with

$$
{}^{E}_{S}\mathbf{T} ={}^{E}_{Z} \mathbf{T} \cdot{}^{Z}_{S} \mathbf{T}. \tag{4.62}
$$

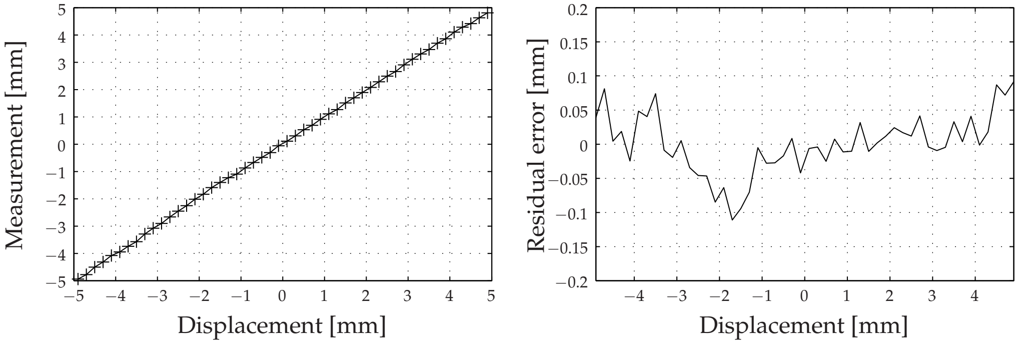
The sensor pose is now known relative to frame **E** of the calibration object.

Range and accuracy measurements

In order to investigate the accuracy of the measurement method, a number of different range measurements have been performed. The sensor has been positioned a distance of 6 mm from the origin of frame **E** along the x-axis, with the same orientation as frame **E**. From this position, the sensor is translated along the three coordinate axes of the sensor tool frame. The relative displacements of the sensor are measured with the described measurement method. Furthermore, the residual error, which is the difference between the predefined translation and the measured displacement is computed. The measurements have been performed with steps of 0.2 mm within the measurement range of the calibration object. The results are shown in figures 4.22, 4.23 and 4.24.

The measurement range in the x-direction (10 mm) is limited by the size of the calibration object. It is somewhat bigger than the measurement range in the y-direction (6 mm) and z-direction (6 mm), which are limited by the camera chip size and diode angle of the sensor. Almost no linear trend can be observed in the residual errors, which shows that the predefined robot movement corresponds to the sensor measurement. This is a very important conclusion as it shows that the robot length unit corresponds to the measured sensor length unit.

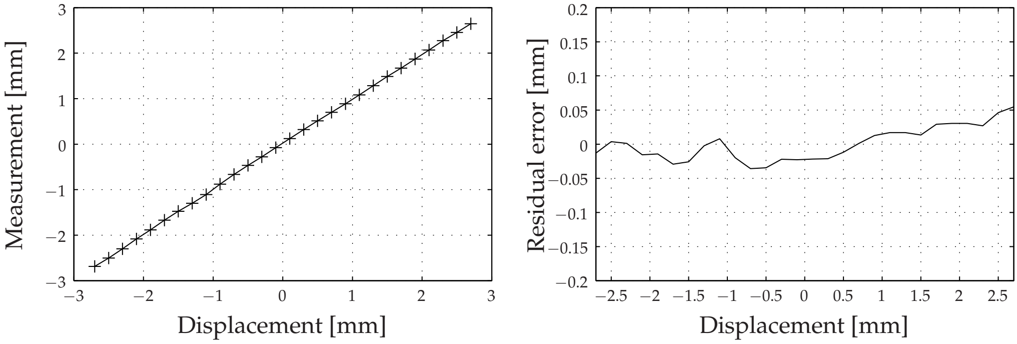
The residual errors for the x-direction are within 0.1 mm, the standard deviation in this direction is 0.044 mm. The residual errors in the y-direction are within 0.05 mm with a standard deviation of 0.025 mm. The residual errors in the z-direction are mainly within 0.05 mm with a standard deviation of 0.038 mm. The



(a) Displacement

(b) Residual error

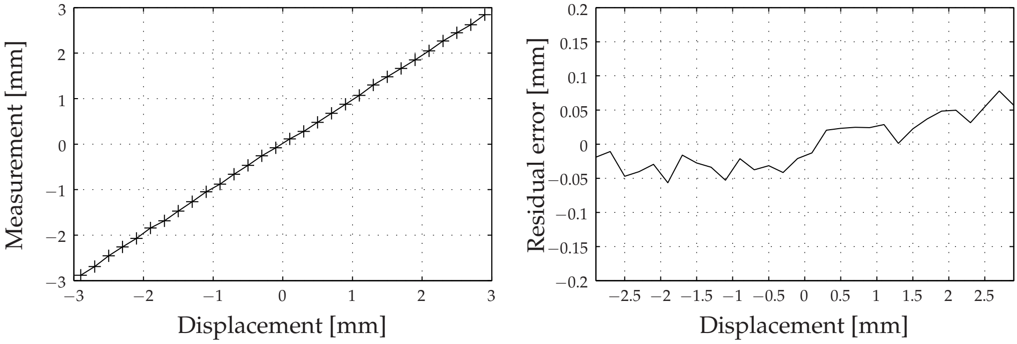
Figure 4.22: Range measurement in sensor x-direction



(a) Displacement

(b) Residual error

Figure 4.23: Range measurement in sensor y-direction



(a) Displacement

(b) Residual error

Figure 4.24: Range measurement in sensor z-direction

residual error in y-direction corresponds to an error perpendicular to the welding direction and an error in z-direction corresponds to a focal error. These directions need to be measured more accurately than the x-direction, which is realised with the proposed measurement procedure.

Beside position range measurements, also orientation range measurements have been performed. The sensor is positioned in the middle of the calibration object. From this position, the sensor is rotated around the three coordinate axes of the sensor tool frame. The relative orientation displacements of the sensor are measured and the residual error is calculated. The measurements have been performed with steps of 0.5 deg within the measurement range of the calibration object. The results are shown in figures 4.25, 4.26 and 4.27.

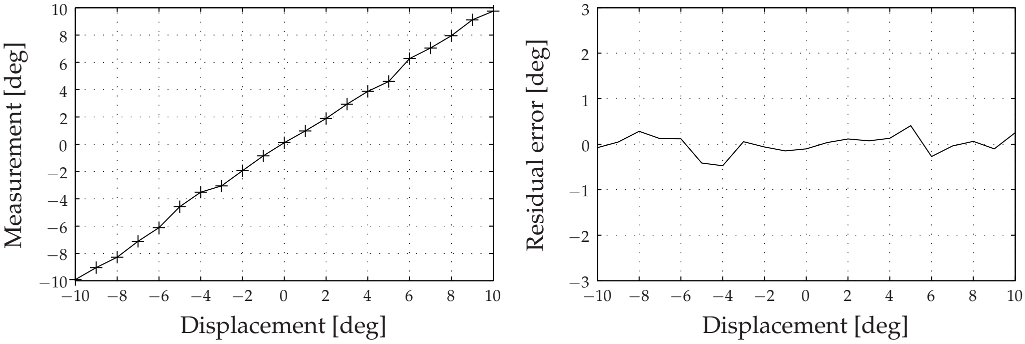
The residual errors for rotations around the x-axis are mainly within 0.5 deg, the standard deviation in this direction is 0.21 deg. The residual errors for rotations around the y-axis are within 1.0 deg with a standard deviation of 0.49 deg. In this direction, a linear trend can be observed in the residual errors, which is expected to be caused by a small error in the diode angle of the sensor. The residual errors for rotations around the z-axis are within 3.0 deg with a standard deviation of 1.29 deg. Also in this direction, a linear trend can be observed in the residual errors. The residual errors of rotations around the y-axis and x-axis correspond to orientation errors of the laser beam, which need to be measured more accurately than the rotation around the z-axis, which is achieved with the proposed measurement procedure.

The range measurements that were performed show that the measurement method produces reliable results. The position measurements are better than the orientation measurements. The residual errors that are obtained are a factor two to three better than the errors presented by Huissoon. In the following section the measurement method will be used in an automatic sensor tool calibration method, to obtain the transformation ${}^{N}_{S}\mathbf{T}$ from robot flange to sensor tool.

4.3.3 Sensor tool calibration procedure

This section presents a least-squares solution to solve equation 4.47 in the presence of measurement noise. The used procedure was obtained from Park and Martin (1994). A complete description of the procedure can be found in their paper, a summary is given below. Equation 4.47 can be written in a rotational and a translational part as

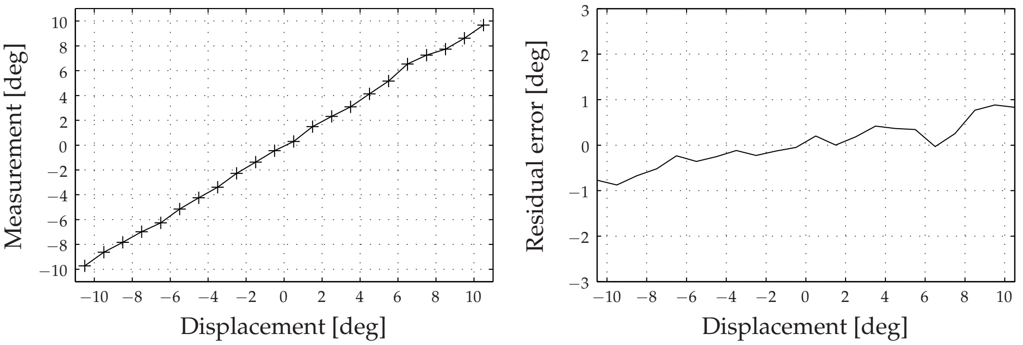
$$\begin{bmatrix} {}^{N_i}_{N_{i+1}}\mathbf{R} & {}^{N_i}_{N_{i+1}}\mathbf{P} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{N}_{S}\mathbf{R} & {}^{N}_{S}\mathbf{P} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{N}_{S}\mathbf{R} & {}^{N}_{S}\mathbf{P} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{S_i}_{S_{i+1}}\mathbf{R} & {}^{S_i}_{S_{i+1}}\mathbf{P} \\ 0 & 1 \end{bmatrix}, \tag{4.63}$$



(a) Displacement

(b) Residual error

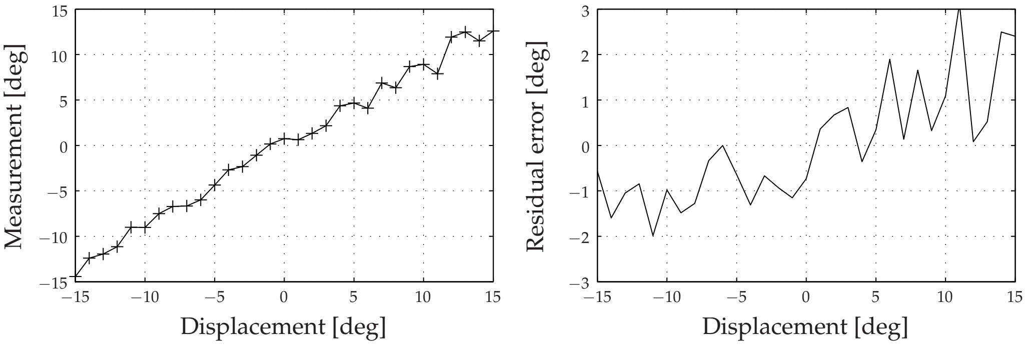
Figure 4.25: Range measurement for rotating around the tool x-axis



(a) Displacement

(b) Residual error

Figure 4.26: Range measurement for rotating around the tool y-axis



(a) Displacement

(b) Residual error

Figure 4.27: Range measurement for rotating around the tool z-axis

implying that,

$$ {}^{N_i}_{N_{i+1}}\mathbf{R} \cdot {}^{N}_{S}\mathbf{P} + {}^{N_i}_{N_{i+1}}\mathbf{P} = {}^{N}_{S}\mathbf{R} \cdot {}^{S_i}_{S_{i+1}}\mathbf{P} + {}^{N}_{S}\mathbf{P}, \tag{4.64} $$

and

$$ {}^{N_i}_{N_{i+1}}\mathbf{R} \cdot {}^{N}_{S}\mathbf{R} = {}^{N}_{S}\mathbf{R} \cdot {}^{S_i}_{S_{i+1}}\mathbf{R}. \tag{4.65} $$

Solving rotational part

Although the calibration procedure presented by Park and Martin (1994) is able to solve the rotational part in equation 4.65, the expected accuracy is low due to the fact that the orientation accuracy of the measurement method is rather low. In practical cases, the orientation does not need to be more accurately known than the mechanical interface is manufactured. Furthermore, the rotation matrix ${}^{N}_{S}\mathbf{R}$ is also used in the translational part (equation 4.64), so errors and fluctuations in the orientation also influence the accuracy of ${}^{N}_{S}\mathbf{P}$. Therefore the orientation of the nominal tool transformation is used and only the translational part ${}^{N}_{S}\mathbf{P}$ will be calibrated.

Solving translational part

The translational part in equation 4.64 is solved by finding a translation vector ${}^{N}_{S}\mathbf{P}$ that minimises

$$ \eta = \sum_{i=1}^{n} \| ({}^{N_i}_{N_{i+1}}\mathbf{R} - \mathbf{I}){}^{N}_{S}\mathbf{P} - {}^{N}_{S}\mathbf{R} \cdot {}^{S_i}_{S_{i+1}}\mathbf{P} + {}^{N_i}_{N_{i+1}}\mathbf{P} \|^2. \tag{4.66} $$

The value of ${}^{N}_{S}\mathbf{P}$ that minimises η (with the ${}^{N}_{S}\mathbf{R}$ from the nominal tool transformation) is then the standard least-squares solution of equation 4.64:

$$ {}^{N}_{S}\mathbf{P} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T d, \tag{4.67} $$

where

$$ \mathbf{C} = \begin{bmatrix} {}^{N_1}_{N_2}\mathbf{R} - \mathbf{I} \\ {}^{N_2}_{N_3}\mathbf{R} - \mathbf{I} \\ \vdots \\ {}^{N_{n-1}}_{N_n}\mathbf{R} - \mathbf{I} \end{bmatrix}, \tag{4.68} $$

and

$$d = \begin{bmatrix} {}^{N}_{S}\mathbf{R} \cdot {}^{S_1}_{S_2}\mathbf{P} - {}^{N_1}_{N_2}\mathbf{P} \\ {}^{N}_{S}\mathbf{R} \cdot {}^{S_2}_{S_3}\mathbf{P} - {}^{N_2}_{N_3}\mathbf{P} \\ \vdots \\ {}^{N}_{S}\mathbf{R} \cdot {}^{S_{n-1}}_{S_n}\mathbf{P} - {}^{N_{n-1}}_{S_n}\mathbf{P} \end{bmatrix} \tag{4.69}$$

Considerations

To carry out the sensor tool calibration procedure, the robot flange has to be positioned in different poses, in such a way that the calibration object stays within the measurement range of the sensor. To achieve this, a nominal sensor tool transformation ${}^{N}_{S_n}\mathbf{T}$ is used, which should closely correspond to the real sensor tool transformation. This nominal sensor tool transformation can for example be obtained from the geometry of the welding head and sensor. A number of predefined movements are performed relative to the nominal tool frame.

4.3.4 Experimental results

The resulting accuracy of a calibrated tool transformation is very difficult to obtain, because it is not possible to exactly know or measure the actual tool transformation and therefore it is not possible to compare the result of the calibration procedure with the actual tool transformation. Instead, the accuracy of the calibration procedure will be judged by comparing the results from a series of experiments with each other.

First of all a repeatability experiment has been performed, where a number of sensor tool calibrations has been carried out at one location in the robot workspace. Every time a sensor measurement is needed, ten sensor measurements are averaged to decrease the effect of stochastic noise on a sensor measurement. The sensor is positioned in 13 different poses above the calibration object.

Figure 4.28 shows that the procedure repeats quite well as the residual errors from the average tool transformation are below 0.1 mm.

Because the sensor is geometrically fixed to the robot flange, the same transformation should be found if the calibration procedure is performed at a different location in the robot workspace. Two repeatability experiments have been performed, where the calibration object is rotated about 45 degrees on the work table in every experiment to make sure the robot joint positions have significantly changed between the experiments.

At the other locations, the procedure shows a similar repeatability as in figure 4.28. The average tool transformation of the calibrations are shown in table 4.3.

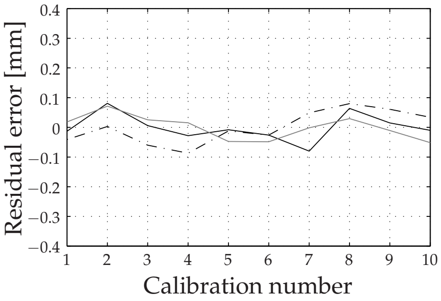


Figure 4.28: Repeatability experiment of the sensor tool calibration procedure (— X, — Y, -·- Z)

Experiment	Average tool transformation (Euler RxRyRz)
1	[-257.970 -41.850 163.075 mm 0 90 -90 deg]
2	[-257.886 -42.378 162.380 mm 0 90 -90 deg]
3	[-257.410 -43.027 162.825 mm 0 90 -90 deg]

Table 4.3: Average tool transformations of three series of tool transformations where the calibration object was located differently in the robot workspace

Although the repeatability of the procedure is good, the average tool transformations that are found show a remarkable difference. The largest difference is about 1.2 mm. By observing equation 4.47, these differences can be caused by errors in the robot positioning (errors in A-matrix of equation 4.3) or sensor measurement errors (errors in B-matrix of equation 4.3). It is very unlikely that the differences are caused by sensor measurement errors, as the positions and orientations of the sensor relative to the calibration object are similar in all three calibration runs. Therefore the differences must be caused by robot geometric errors. In section 4.3.5, the results of these experiments will be simulated. In these simulations, the effect of robot geometric errors on the calibration procedure is taken into account.

4.3.5 Simulations

The models that were derived in section 3.5 can be used to calculate the sensor measurements if a single seam trajectory is defined. These models can also be used for the sensor tool calibration procedure, although instead of a single seam trajectory, a calibration object with two edges and a virtual edge is positioned in the sensors field-of-view. Each of the three edges can be modelled with the seam trajectory model as a linear segment between two points. The sensor model can now be used three times to calculate the intersections of each of the three edges. Once these intersections are known relative to the sensor tool frame, the theory that is described in section 4.3.2 can be used to compute the transformation ${}^{E}_{S}\mathbf{T}$

that is used in the sensor tool calibration procedure. Different sources of errors (sensor noise, quantisation, geometric robot errors) can be used during the simulations.

Sensor noise and quantisation

First of all a simulation is carried out, which shows that the sensor tool calibration procedure is working correctly. A sensor tool calibration has been simulated, where only sensor and quantisation are used as error sources. A total of ten simulation runs were carried out. The average tool transformation that was found is [-257.000 -43.001 163.002 mm 0 90 -90 deg], which only differs a few micrometres from the actual tool transformation that was used to model the sensor measurements. The residual errors of the single calibration runs from the average tool transformation are plotted in figure 4.29.

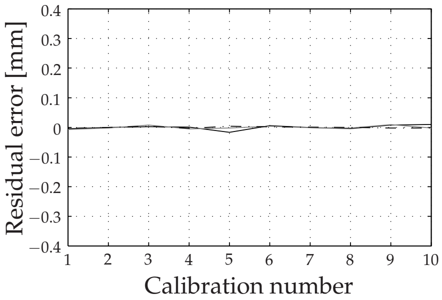


Figure 4.29: Repeatability simulation of the sensor tool calibration procedure for the ideal case with only sensor noise and quantisation (— X, — Y, -·- Z)

The calibration procedure repeats very well in this simulation. The sensor noise and quantisation errors affect the outcome only very little (less than 20 μm). Therefore there is no need to increase the number of robot movements that is used in a single calibration run, nor to increase the number of measurements that is averaged at every robot pose.

Geometric armlengths model

The previous simulation has been repeated, but in this case geometric errors in the robot model are taken into account. In this subsection the geometric model (section 3.5.3) where tip errors are attributed to errors in the robot arm lengths is used to simulate the sensor measurements. Furthermore in these simulations, the calibration object is positioned in three different locations in the robot workspace to see if this influences the outcome of the simulations. In all three simulation runs, the repeatability of the calibration procedure is better than 20 μm and therefore no residual error is plotted. The average tool transformation that was calculated is plotted in table 4.4.

Experiment	Average tool transformation (Euler RxRyRz)
Actual	[-257.000 -43.000 163.000 mm 0 90 -90 deg]
1	[-256.704 -42.750 163.056 mm 0 90 -90 deg]
2	[-256.756 -42.686 162.972 mm 0 90 -90 deg]
3	[-256.767 -42.776 162.726 mm 0 90 -90 deg]

Table 4.4: Average tool transformations of three simulated tool calibration runs, where the calibration object was located at different positions in the robot workspace. Simulated sensor measurements were obtained using the geometric armlengths model in the robot direct geometric model.

First of all it can be observed that the correct tool transformations can not be obtained anymore. The average tool transformations differ up to 0.3 mm in different directions. The geometric errors that are present in the system due to the fact that the robot geometric models do not correctly represent the actual robot geometry influence the tool calibration procedure in such a way, that an accurate tool transformation can not be found.

Secondly, the average tool transformations that are obtained also differ between different locations of the calibration object in the robot workspace. The differences in these simulations are however not as large as the ones that are found in the experiments. The simulations in this section are performed using a geometric robot model, where only the armlenghts parameters are different. This results in a different position of the robot tip, but the orientation does not change. During the sensor tool calibration procedure the robot movements that are performed consist mainly of orientation changes. Therefore the simulations from this section are repeated in the next section, but in that case the encoder offsets geometric robot model is used as it also results in orientation errors of the robot tip.

Geometric encoder offsets model

In this section the sensor measurements are simulated using a geometric model where tip errors are attributed to errors in the encoder offsets. The average tool transformations that were calculated are given in table 4.5.

Experiment	Average tool transformation (Euler RxRyRz)
Actual	[-257.000 -43.000 163.000 mm 0 90 -90 deg]
1	[-256.965 -42.886 162.180 mm 0 90 -90 deg]
2	[-256.855 -43.275 162.475 mm 0 90 -90 deg]
3	[-257.068 -43.491 163.028 mm 0 90 -90 deg]

Table 4.5: Average tool transformations of three simulated tool calibration runs, where the calibration object was located at different locations in the robot workspace. Simulated sensor measurements were obtained using the encoder offsets geometric robot model.

Again, it can be observed that the actual tool transformation can not be found correctly. The maximum differences in the tool positions increase to 0.9 mm. The differences between the calibration runs where the calibration object is located in different positions in the robot workspace has increased as well to 0.9 mm. It can be concluded that in simulations with a more realistic geometric robot model in which also orientation errors occur at the robot tip, the accuracy of the calibration procedure is unsatisfactory.

4.3.6 Discussion

In this section an automatic sensor tool calibration procedure has been developed and implemented that computes the transformation ${}^{N}_{S}\mathbf{T}$ between the robot flange and the sensor tool frame. Experiments and simulations have been carried out, which shows that the calibration procedure is correctly implemented. Both experiments and simulations show that the resulting accuracy of the calibration procedure is influenced by robot geometric errors and that the accuracy is therefore limited to about 1 mm.

A straightforward solution to increase the accuracy of tool calibration procedures is obviously to increase the accuracy of the robot geometric models that are used. However, this is not straightforward due to the following problems:

1. It is not straightforward to make a good geometric model structure and find the correct parameters. Many parameters that influence the robot geometry may have to be taken into account, e.g. arm lengths, encoder offsets, link angles, elasticity due to tip mass, etc.

2. Symbolic expressions for calculating the Inverse Geometric Model can usually not be derived for models consisting of these parameters. Instead, iterative solutions need to be calculated, which are more computer intensive.

Increasing the accuracy of the robot geometric models is out of the scope of this work, although in chapter 7 some recommendations are given for using more accurate robot geometric models.

4.4 Combined laser and sensor tool calibration

In the previous sections, a laser tool calibration and a sensor tool calibration procedure have been described to find the transformations ${}^{N}_{L}\mathbf{T}$ and ${}^{N}_{S}\mathbf{T}$. In both of these calibration procedures, a number of movements have been defined, relative to a known coordinate frame. These movements have been measured in the unknown tool coordinate frame. Equations of the form $\mathbf{A} \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B}$ are solved. Due to the robot movements, both procedures have an error in the matrix $\mathbf{A}$, because the geometric robot model does not correspond to the actual robot geometry. The resulting errors in the transformations ${}^{N}_{L}\mathbf{T}$ and ${}^{N}_{S}\mathbf{T}$ can be up to 1 mm and therefore the error in the transformation ${}^{L}_{S}\mathbf{T}$ may be even larger. This results in an offset during laser welding, which is far too large for laser welding.

At the start of this chapter it was mentioned that the most important transformation that needs to be accurately known (better than 0.2 mm) is the transformation ${}^{L}_{S}\mathbf{T}$, which can be computed using equation 4.2 from the result of the laser tool calibration procedure and the sensor tool calibration procedure. Because both ${}^{N}_{L}\mathbf{T}$ and ${}^{N}_{S}\mathbf{T}$ can not be accurately determined due to geometric robot errors, this section describes a combined calibration procedure, where the transformation ${}^{L}_{S}\mathbf{T}$ is directly measured, without making a robot movement. Therefore, errors in the robot geometric model do not influence the measurement, which makes it much more accurate.

4.4.1 Calibration procedure

In order to measure the transformation ${}^{L}_{S}\mathbf{T}$, the two calibration objects that are used in the sensor and laser tool calibration procedure are combined in a single calibration object (figure 4.30).

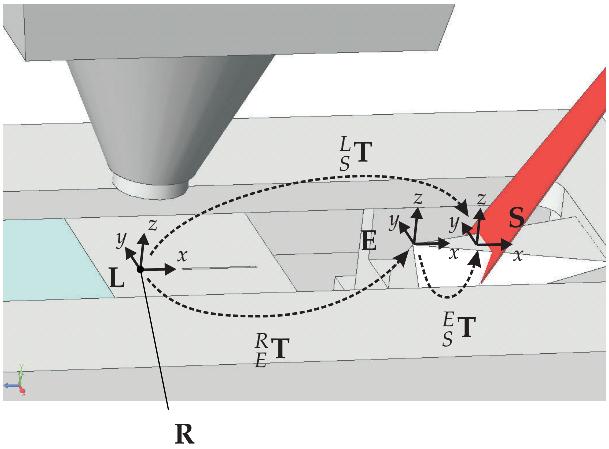


Figure 4.30: Laser and sensor positioned above the combined calibration object. Note that in this figure, frame **L** *(attached to the welding head) overlaps frame* **R** *(attached to a pinhole in the calibration object), which is not generally the case.*

From this figure, it can be derived that the transformation ${}^{L}_{S}\mathbf{T}$ is computed as

$${}^{L}_{S}\mathbf{T} = {}^{L}_{R}\mathbf{T} \cdot {}^{R}_{E}\mathbf{T} \cdot {}^{E}_{S}\mathbf{T}. \tag{4.70}$$

The sensor calibration object has its own frame **E** and is mounted in such a way that the transformation ${}^{R}_{E}\mathbf{T}$ is known from the geometry of the object. The transformation ${}^{E}_{S}\mathbf{T}$ can be measured using the measurement method described in section 4.3.2. The only unknown transformation is ${}^{L}_{R}\mathbf{T}$, which will be derived next. After the laser tool calibration procedure has been performed, a number of important conditions are determined:

1. The welding head is positioned normal to the calibration object.
2. The origin of the laser frame **L** has been determined in the camera image, so the position of the light spot can be measured relative to frame **L**.
3. The focal position of the welding head is positioned at the surface of the calibration object.
4. The scaling factors c_x and c_y from camera pixels to millimetres are determined.

These conditions will be used to determine transformation ${}^{L}_{R}\mathbf{T}$. As the optical axis of the welding head is normal to the calibration object (condition 1), the vector ${}^{L}_{R}\mathbf{r}_z$ equals $[0\ \ 0\ \ 1]^T$. Using condition 2, 3 and 4, it is known that ${}^{L}_{R}P_x$ and ${}^{L}_{R}P_y$ can be measured from the camera image, using the scaling factor c_x and c_y, and ${}^{L}_{R}P_z = 0$. The transformation ${}^{L}_{R}\mathbf{T}$ is now partly determined as:

$$
{}^{L}_{R}\mathbf{T} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & {}^{L}_{R}P_x \\ \sin(\phi) & \cos(\phi) & 0 & {}^{L}_{R}P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.71}
$$

where the only unknown is the rotation angle ϕ around the optical axis of the welding head. This rotation can not be determined from a single light spot in the camera image, therefore the laser tool calibration surface, which contains the light spot has been extended with a line source to determine the orientation. The line source points in the direction of the x-axis of frame **R**.

4.4.2 Image processing

A typical camera image, containing both the light spot and the line source, is shown in figure 4.31. The vector ${}^{L}_{R}\mathbf{P}$ and angle ϕ will be determined from the camera image using image processing.

Thresholding

The original image is a grayscale image. The first operation that will be performed on this image is thresholding. The goal of thresholding is to distinguish

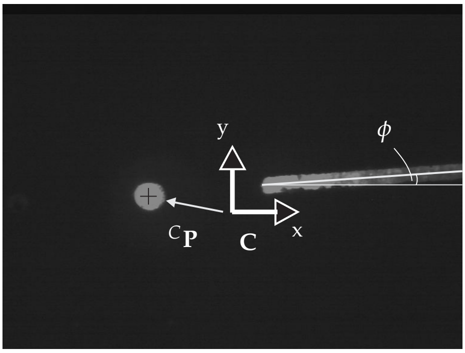


Figure 4.31: Typical camera image with a light spot and a line source for combined tool calibration

the pixels belonging to the spot and the line and the pixels belonging to the background. This is achieved by making the pixels that have a lower value than the threshold black and pixels that have a value that is equal or larger than the threshold white. The resulting image after the threshold operation is given in figure 4.32.

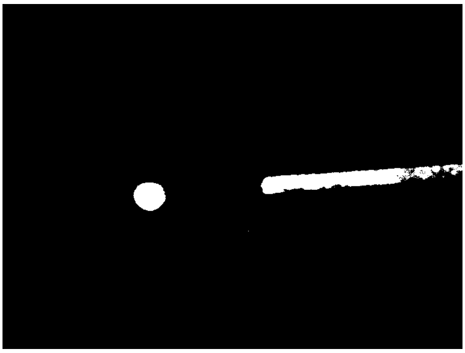


Figure 4.32: Thresholded camera image for combined tool calibration. Threshold value of 60

Determining the spot position

To determine the spot position from the thresholded image, pixels that belong to the spot need to be distinguished from pixels that belong to the line. For this purpose, the diameter of the spot has been purposely made somewhat larger than the thickness of the line. An erosion operation (Van der Heijden, 1994) will be used to remove the line from the thresholded camera image. By choosing the erosion mask as a circle with a certain diameter, only the pixels where a logical AND operation with the mask equals the mask are made white, the rest is made black. The resulting image after the erosion operation is given in figure 4.33.

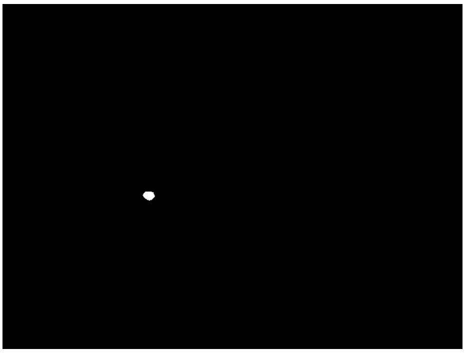


Figure 4.33: Eroded camera image for combined tool calibration. Mask diameter of 15 pixels

The mask diameter is chosen in such a way that it is larger than the line width, but smaller than the spot diameter. The center of the spot ${}^{C}\mathbf{P}$ in pixels is easily found by computing the center of gravity (IPL98, 1998) of the resulting image. The vectors ${}^{L}_{R}P_x$ and ${}^{L}_{R}P_y$ can be computed from the center of gravity using equation 4.12.

Determining the line angle

To determine the rotation angle ϕ, a Hough transform (Van der Heijden, 1994) is performed on the thresholded image. Each straight line in an image can be described in the following parametric form:

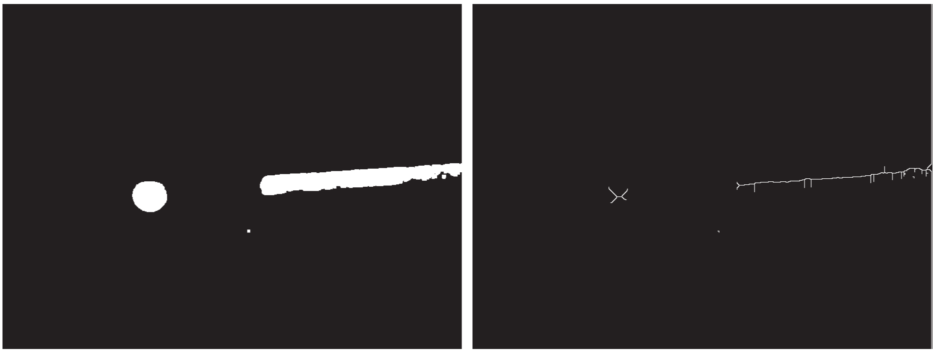
$$x\sin(\phi) - y\cos(\phi) + r = 0 \qquad \phi \in [0, 2\pi], r \geq 0, \tag{4.72}$$

where ϕ is the direction of the line relative to the x-axis and r is the shortest distance to the origin. It is well known that an infinite number of lines can go through a single point in the image plane. If that point has coordinates (x_0, y_0) in the image plane, all the lines that go through it obey the following equation:

$$r(\phi) = x_0\sin(\phi) - y_0\cos(\phi). \tag{4.73}$$

This corresponds to a sinusoidal curve in the (r, ϕ) plane, which is unique to that point. If the curves corresponding to two points are superimposed, the location (in the Hough space) where they cross correspond to lines (in the original image space) that pass through both points. More generally, a set of points that form a straight line in the image space, will produce sinusoids which cross at the parameters for that line in the Hough space. Thus, the problem of detecting colinear points can be converted to the problem of finding concurrent curves.

The line in the thresholded camera image is more than one pixel thick, so in practice, a number of lines at different angles fit within the camera image. To find a single line with the correct angle and to reduce the number of computations, two basic image pre-processing steps are performed before the Hough transform is applied (figure 4.34). First of all, the thresholded image is dilated, which means that the neighbouring nine pixels for each white pixel in the original are made white. The dilation procedure thickens the line and fills single black pixels that belong to the line. Next, the dilated image is skeletonised, which reduces the thick parts to single pixel parts.



(a) Dilation

(b) Skeletonising

Figure 4.34: Pre-processing steps for the Hough transform.

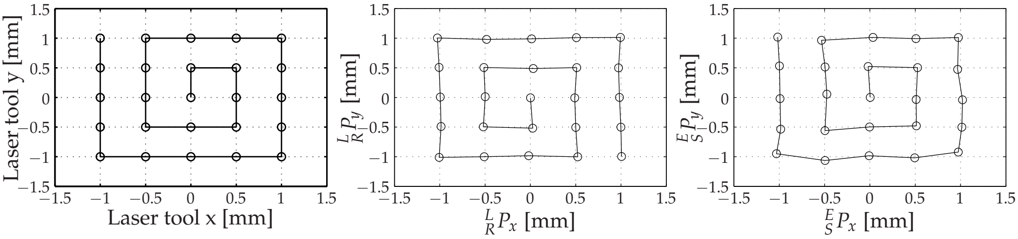
The white pixels in the skeletonised image are drawn as sinusoids in the Hough space using equation 4.73. The result is given in figure 4.35. Because the original image contains one line, the Hough space contains one spot, where the different sinusoids intersect. The angle ϕ and distance r to the origin are easily found from the intersection in the Hough space.

Figure 4.35: Hough transform for $r(\phi)$. Resolution of 0.1 degrees from -10 degrees to 10 degrees.

4.4.3 Results

Before starting the combined calibration procedure, a laser tool calibration procedure has been performed to satisfy the four conditions stated before. Once the laser tool calibration procedure has finished, the transformation ${}^{L}_{S}\mathbf{T}$ can be obtained using the combined calibration procedure. The transformation ${}^{R}_{E}\mathbf{T}$ is accurately determined from the geometry of the calibration object in Euler RxRyRz notation as [45.0 0.0 3.0 mm 0.0 0.0 0.0 deg].

The result of the combined calibration procedure is independent of the exact position of the welding head relative to the calibration object, as long as the four conditions are satisfied and the spot, line and calibration object are within the measurement field-of-view. Therefore the combined calibration procedure is performed at different positions in the XY-plane of the laser tool frame. The relative movements of the laser tool frame, the measured displacement of the spot and the measured displacement of the calibration object are shown in figure 4.36.



(a) Movement (b) Spot measurement (c) Calibration object measurement

Figure 4.36: Pre-defined laser tool movement with the measured displacement of the spot and the measured displacement of the calibration object.

The position of the spot is measured with a maximum residual error of less than 0.05 mm. The position of the calibration object is measured with a maximum residual error of 0.1 mm. The residual error of the transformation between laser and sensor tool frame is shown in figure 4.37. The maximum residual error of this transformation is 0.1 mm, which shows that the combined calibration procedure reproduces very well. As shown before, the total accuracy of the combined calibration procedure is determined by the measurement accuracy of the spot measurement and the calibration object measurement.

Stochastic errors in the measurements can easily be removed by averaging the result of the calibration procedure at the different measurement positions. The average transformation ${}^{L}_{S}\mathbf{T}$ in Euler RxRyRz notation is [48.10 -2.89 -1.16 mm 0 0 0 deg].

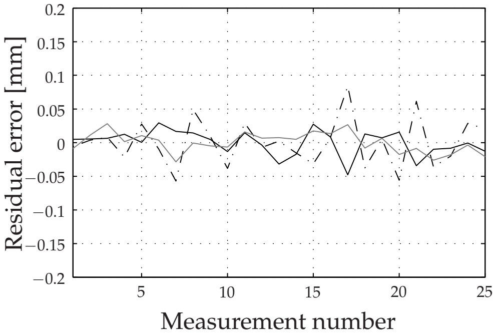


Figure 4.37: Residual error of the transformation between laser and sensor tool frame at the different measurement locations (— X, -·- Y, — Z).

4.5 Discussion

From the tool calibration experiments and simulations in this chapter, the following conclusions can be drawn:

- An automatic laser tool calibration procedure has been developed and implemented that computes the transformation ${}^{N}_{L}\mathbf{T}$ between the robot flange and the laser tool frame using a coaxial camera that is attached to the laser welding head. Experiments have been performed, which show that the resulting accuracy of the calibration procedure is influenced by robot geometric errors and that because of these errors, the accuracy is limited to about 1 mm.

- An automatic sensor tool calibration procedure has been developed and implemented that computes the transformation ${}^{N}_{S}\mathbf{T}$ between the robot flange and the sensor tool frame. Both experiments and simulations show that the resulting accuracy of the calibration procedure is influenced by robot geometric errors and that therefore the accuracy is limited to about 1 mm.

- A combined tool calibration procedure has been developed and implemented that computes the transformation ${}^{L}_{S}\mathbf{T}$ between the laser and sensor tool frame without moving the robot. This method is more accurate (< 0.1 mm), because robot geometric errors do not influence the accuracy of this method.

Chapter 5

Seam teaching using point-to-point movements

5.1 Introduction

Seam teaching for arc welding has been described by many authors, for an overview see e.g. Nayak and Ray (1993). They focus on ways to use sensor information for real-time seam tracking. The accuracy requirements for laser welding however give a different perspective to seam teaching, because the resulting accuracy should be better than 0.2 mm. This is close to the limits that can be obtained by commonly used 6-axes industrial robots. Many effects (robot geometric model, errors in tool transformations) that are acceptable for arc welding can not be accepted for laser welding. This chapter considers seam teaching using point-to-point movements where dynamic effects (robot dynamics, robot-sensor synchronisation) do not play a role. The use of sensor information during the robot motion for real-time seam tracking will be considered in chapter 6.

The following factors influence the accuracy that can be achieved with a seam-teaching system:

- Sensor errors (measurement noise, pixel resolution, etc)
- Errors in the sensor and laser tool transformations
- Errors in the used nominal geometric robot model (in the robot controller)

A clear overview of the influence of these errors on the accuracy during welding does not exist up to now, so a simulation environment has been developed, where the influence of errors can be investigated individually. The previous chapter has showed how the robot with its controller, the seam tracking sensor and a seam trajectory could be modelled. These models will be used now in the simulation environment to show that the seam teaching process can be investigated successfully.

The outline of this chapter is as follows. The seam teaching process is described in section 5.2. Two seam teaching algorithms for teaching of a seam trajectory with an unknown geometry are described in section 5.3. Experiments (section 5.5) and simulations (section 5.6) have been performed, which show how different errors influence the accuracy that can be obtained with a seam teaching system.

5.2 The seam teaching process

Because point-to-point movements are considered in this chapter, the sensor only measures when the robot is at rest. Index p denotes the measurement location along the seam trajectory. At location p on the seam trajectory, the seam tracking sensor measures the data $s_{x,p}$, $s_{y,p}$, $s_{z,p}$ and $s_{rx,p}$ with respect to its sensor coordinate frame **S**. These measurements are computed to transformation ${}^{S}_{G}\mathbf{T}_p$ using equation 3.12. According to figure 3.1, a location ${}^{F}_{G}\mathbf{T}_p$ on the seam trajectory can also be expressed with respect to a generic frame **F** (which can be **B**, **M** or **P**) as

$$ {}^{F}_{G}\mathbf{T}_p = {}^{F}_{S}\mathbf{T}_p \cdot {}^{S}_{G}\mathbf{T}_p, \tag{5.1} $$

where ${}^{F}_{S}\mathbf{T}_p$ is the location of the sensor tool with respect to frame **F**.

In this section an overview of the seam teaching process is given. Two phases are distinguished: the teaching phase and the re-teaching phase. In the case of teaching, no future information of the seam trajectory is available, so an estimation (extrapolation) for the next $(p+1)$ seam location will be computed by a teaching algorithm from historic seam locations that are stored in buffers. In the case of re-teaching, no estimation for the next seam location needs to be computed as it is already known from a previous teaching procedure or from off-line programming software. To correct for clamping and position errors, the re-teaching phase is used to obtain the actual seam trajectory for each individual product of a series.

5.2.1 Teaching phase

A schematic representation for teaching an unknown seam trajectory is shown in figure 5.1.

Every teach step p, the seam teaching algorithm does the following:

- Measure and store the current sensor tool location ${}^{F}_{S}\mathbf{T}_p$ in the Sensor Trajectory Buffer and the sensor measurement ${}^{S}_{G}\mathbf{T}_p$ in the Sensor Measurement Buffer.

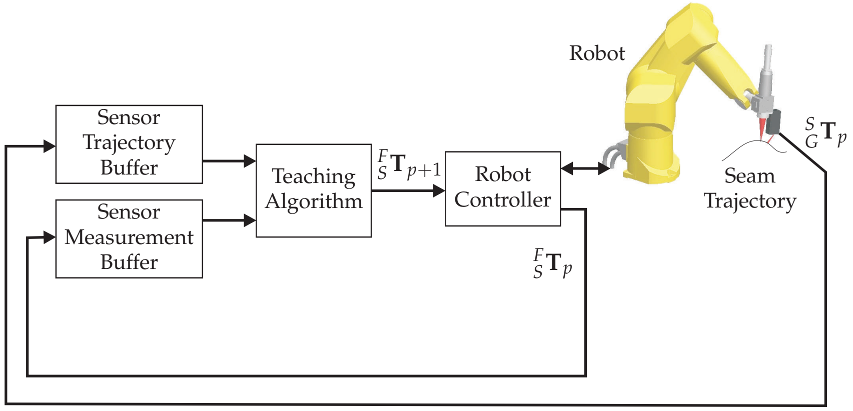


Figure 5.1: Schematic representation of the teaching phase

- Calculate a prediction $^F_S\mathbf{T}_{p+1}$ of the next location on the seam trajectory and let the sensor tool move to this location.
- Perform basic safety checks, e.g. the seam trajectory should always be in the field-of-view of the sensor during teaching.

At teach step p, the Seam Trajectory Buffer contains the sensor tool locations

$$[^F_S\mathbf{T}_1 \;\; \ldots \;\; ^F_S\mathbf{T}_{p-1} \;\; ^F_S\mathbf{T}_p]. \tag{5.2}$$

Furthermore, the sensor measurements at teach step p are stored in the Sensor Measurements Buffer. This buffer contains

$$[^S_G\mathbf{T}_1 \;\; \ldots \;\; ^S_G\mathbf{T}_{p-1} \;\; ^S_G\mathbf{T}_p]. \tag{5.3}$$

A seam location $^F_G\mathbf{T}_p$ can be computed using equation 5.1 from the sensor measurements $^S_G\mathbf{T}_p$ and the transformation $^F_S\mathbf{T}_p$. During seam teaching experiments, these values are computed from real measurements of the sensor and robot. Simulations can be carried out, where these values are computed using the models described in section 3.5. The prediction $^F_S\mathbf{T}_{p+1}$ of the next location on the seam trajectory is computed by the chosen seam teaching algorithm and will be discussed in section 5.3.

5.2.2 Re-teaching phase

A schematic representation of re-teaching is given in figure 5.2. The locations of a pre-defined seam trajectory are already stored in the Nominal Seam Trajectory Buffer. Instead of calculating a prediction $^F_S\mathbf{T}_{p+1}$ of the next seam location during every teach step, the sensor tool is moved to the next location on this nominal seam trajectory, from start to end. At every step p, the sensor measurement ${}^{S}_{G}\mathbf{T}_p$ is stored in the Sensor Measurement Buffer. Furthermore, the robot is asked for the current sensor tool location ${}^{F}_{S}\mathbf{T}_p$, which is stored in the Sensor Trajectory Buffer. A seam location ${}^{F}_{G}\mathbf{T}_p$ on the actual seam trajectory can be computed using equation 5.1.

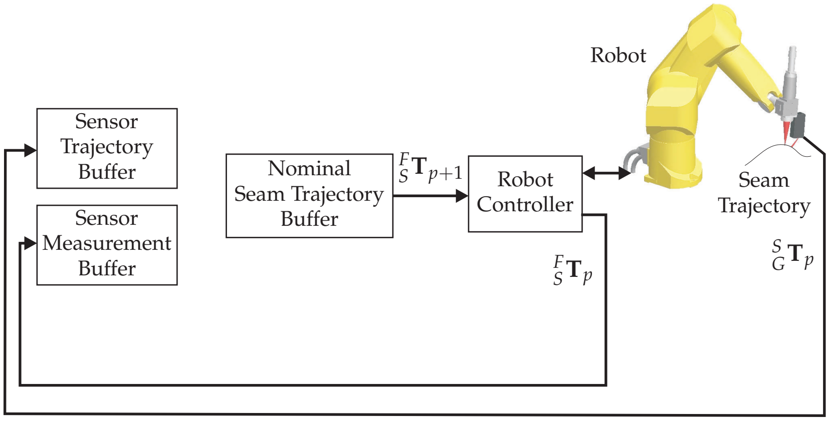


Figure 5.2: Schematic representation of the replay or re-teaching phase

5.3 Teaching algorithms

The sensor that is used in this work measures four degrees-of-freedom. The transformation ${}^{S}_{G}\mathbf{T}$ between the sensor and a location on the seam trajectory is computed using equation 3.12. The seam teaching algorithms that are described in the next sections can also be used with sensors that have the ability to measure additional rotational degrees-of-freedom, e.g. Iakovou et al. (2005). The calculation of transformation ${}^{S}_{G}\mathbf{T}$ should be adapted for such sensors.

The teaching algorithm can use every value stored in the Sensor Trajectory Buffer and the Sensor Measurement Buffer to compute a prediction ${}^{F}_{S}\mathbf{T}_{p+1}$ of the next location on the seam trajectory. How this prediction is computed depends on the used teaching algorithm. A number of teaching algorithms will be described in the next sections.

5.3.1 Adapted tractrix algorithm

A straightforward algorithm for teaching of 3D seam trajectories is the tractrix algorithm (Nayak and Ray, 1993). This algorithm has been adapted for teaching of 3D seam trajectories in a 'first teach, then weld' approach (Diphoorn, 2004). A graphical representation of the seam teaching algorithm is given in figure 5.3.

This figure shows a seam trajectory, with two known seam locations ${}^F_G\mathbf{T}_{p-1}$ and ${}^F_G\mathbf{T}_p$.

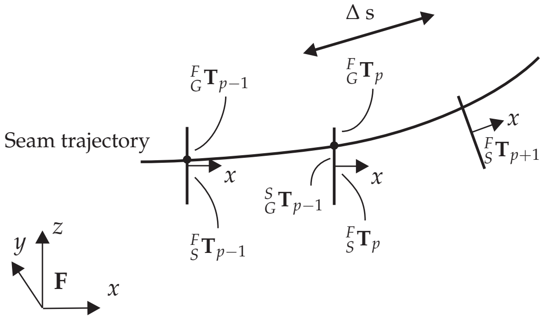


Figure 5.3: Schematic representation of the tractrix algorithm. The x-axis of the sensor tool frame is moving along the seam trajectory.

The seam location ${}^F_G\mathbf{T}_p$ and the sensor measurements s_x, s_y, s_z and s_{rx} at teach step p are used to make a prediction of the next seam location. The algorithm calculates the prediction ${}^F_S\mathbf{T}_{p+1}$ as

$$ {}^F_S\mathbf{T}_{p+1} = {}^F_G\mathbf{T}_p \cdot \mathbf{Rot_z}(\alpha) \cdot \mathbf{Rot_y}(\beta) \cdot \mathbf{Trans}(\Delta s, 0, 0). \tag{5.4} $$

The tractrix algorithm corrects the orientation of seam location ${}^F_G\mathbf{T}_p$ for teaching of 3D seams by rotating with an angle α around the z-axis and an angle β around the y-axis and moving a step Δs in the new x-direction. This way, the x-axis is directed along the seam. The angles α and β are calculated by

$$ \alpha = K_\alpha \arctan\left(\frac{s_{y,p}\cos(-s_{rx,p}) - s_{z,p}\sin(-s_{rx,p})}{\Delta s}\right), \tag{5.5} $$

and

$$ \beta = K_\beta \arctan\left(\frac{s_{y,p}\sin(-s_{rx,p}) + s_{z,p}\cos(-s_{rx,p})}{\Delta s}\right), \tag{5.6} $$

where $s_{y,p}$, $s_{z,p}$ and $s_{rx,p}$ are sensor measurements at teach step p and K_α and K_β have a value between 0 and 1. These values are a trade-off between the ability to follow sharp seam radii and smooth motion behaviour. Choosing these values too big (in combination with a small step size Δs) can cause fluctuating motion behaviour.

5.3.2 Polyfit algorithm

The tractrix algorithm is a simple and straightforward algorithm, but has its limitations. It only uses one previous seam location and does not have the possibility to filter a number of sensor measurements. This section presents a more advanced polyfit algorithm that gives the operator control on the number of points m that are used in the fit and the polynomial order n . A graphical representation of the polyfit seam teaching algorithm is shown in figure 5.4.

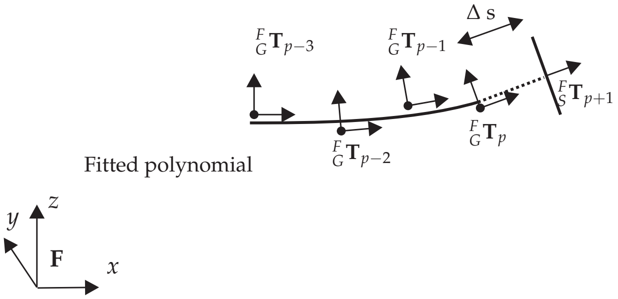


Figure 5.4: Schematic representation of the polyfit seam teaching algorithm

A polynomial of order n has the form

$$\mathbf{p}(p) = \mathbf{a}_n p^n + \mathbf{a}_{n-1} p^{n-1} + \ldots + \mathbf{a}_1 p + \mathbf{a}_0. \tag{5.7}$$

The last m position vectors $[{}^F_G\mathbf{P}_{p-m} \ \ \ldots \ \ {}^F_G\mathbf{P}_{p-1} \ {}^F_G\mathbf{P}_p]$ are used to find the coefficients $\mathbf{a}_0$ to $\mathbf{a}_n$ of the polynomial. These coefficients are computed in such a way that the error criterium

$$\sum_{j=0}^{m} |\mathbf{p}(p-j) -{}^F_C \mathbf{P}_{p-j}|, \tag{5.8}$$

is minimised in a least squares sense. The direction of the polynomial can be evaluated by differentiating equation 5.7 with respect to p as

$$\dot{\mathbf{p}}(p) = n \cdot \mathbf{a}_n p^{n-1} + (n-1) \cdot \mathbf{a}_{n-1} p^{n-2} + \ldots + \mathbf{a}_1. \tag{5.9}$$

The position ${}^F_S\mathbf{P}_{p+1}$ of the predicted sensor tool location is extrapolated as

$${}^F_S\mathbf{P}_{p+1} = \mathbf{p}(p) + \Delta s \frac{\dot{\mathbf{p}}(p)}{|\dot{\mathbf{p}}(p)|}. \tag{5.10}$$

The orientation ${}^{F}_{S}\mathbf{R}_{p+1}$ of the predicted sensor tool location will now be computed from its three orthonormal vectors. The last m surface normal vectors $[{}^{F}_{G}\mathbf{r}_{z,p} \cdots {}^{F}_{G}\mathbf{r}_{z,p-m}]$ of the seam trajectory are averaged to find

$$ {}^{F}_{S}\mathbf{r}_{z} = \frac{\sum_{j=1}^{m} {}^{F}_{C}\mathbf{r}_{z,p-j}}{|\sum_{j=1}^{m} {}^{F}_{C}\mathbf{r}_{z,p-j}|}. \tag{5.11} $$

${}^{F}_{S}\mathbf{r}_{x}$ is computed from the derivative of the polynomial fit as

$$ {}^{F}_{S}\mathbf{r}_{x} = \frac{\dot{\mathbf{p}}(n)}{|\dot{\mathbf{p}}(n)|}. \tag{5.12} $$

Because ${}^{F}_{S}\mathbf{r}_{y}$ should be perpendicular to ${}^{F}_{S}\mathbf{r}_{x}$ and ${}^{F}_{S}\mathbf{r}_{z}$ it is computed as

$$ {}^{F}_{S}\mathbf{r}_{y} = \frac{{}^{F}_{S}\mathbf{r}_{z} \times {}^{F}_{S}\mathbf{r}_{x}}{|{}^{F}_{S}\mathbf{r}_{z} \times {}^{F}_{S}\mathbf{r}_{x}|}. \tag{5.13} $$

To keep ${}^{F}_{S}\mathbf{R}_{n+1}$ orthonormal a correction is applied to find

$$ {}^{F}_{S}\mathbf{r}^{c}_{x} = {}^{F}_{S}\mathbf{r}_{y} \times {}^{F}_{S}\mathbf{r}_{z}. \tag{5.14} $$

The orientation ${}^{F}_{S}\mathbf{R}_{p+1}$ of the predicted sensor tool location is defined as

$$ {}^{F}_{S}\mathbf{R}_{n+1} = \left[\ {}^{F}_{S}\mathbf{r}^{c}_{x} \ \ {}^{F}_{S}\mathbf{r}_{y} \ \ {}^{F}_{S}\mathbf{r}_{z}\ \right]. \tag{5.15} $$

The orientation is filtered over the last m locations, which gives some suppression of noise. To limit deviations in the orientation during teaching, a factor K_{γ} between 0 and 1 is introduced. It is used to interpolate the orientation between the orientation ${}^{F}_{S}\mathbf{R}_{n}$ at the current location ($K_{\gamma} = 0$) and the orientation ${}^{F}_{S}\mathbf{R}_{n+1}$ at the predicted location ($K_{\gamma} = 1$). By choosing $K_{\gamma} = 0$, no orientation changes are allowed at all, which is convenient for sensors that do not require a certain orientation for measuring (Iakovou et al., 2005).

5.4 Seam trajectories

A number of different seam trajectories have been used as case studies for both experiments and simulations. These seam trajectories have different properties, making it easier to distinguish the different effects that may occur. The cases have been chosen in such a way that they provide a good representation of seam

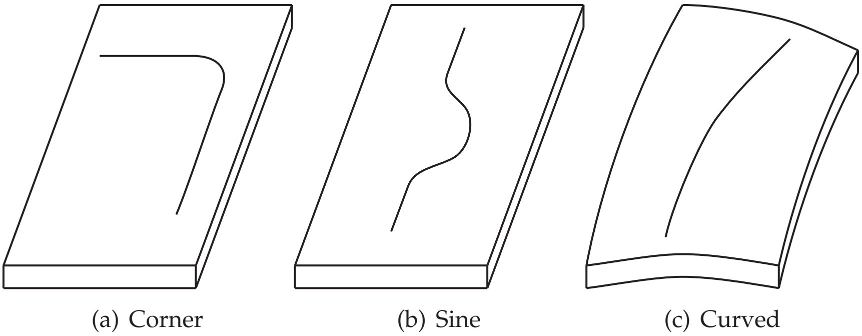

(a) Corner (b) Sine (c) Curved

Figure 5.5: Case seams

trajectories that may occur in an industrial environment. They are shown in figure 5.5.

These seams have been manufactured by laser cutting of sheet metal for real teaching experiments. These trajectories are also used in the simulation environment, so the results from experiments can be compared with the simulation results. A goal of the experiments is to obtain the accuracy of the laser focal point with respect to the seam trajectory during the re-teaching phase. The sensor is used both for teaching the seam trajectory during the teaching phase and for measuring the predicted accuracy of the laser focal point during the re-teaching phase. To do this a dummy laser welding head has been manufactured (figure 5.6), where the sensor can be mounted on different positions with respect to the robot flange. One of these positions represents the sensor tool position during teaching and another one represents the position of the laser focal point during welding.

5.5 Experimental results

The different seam trajectories have been taught with the seam tracking sensor using the polyfit seam teaching algorithm. During seam teaching, the sensor is mounted at the sensor tool location at the right of the dumy head. The sensor tool transformation is calibrated using the procedure described in section 4.3. After the teaching procedure has completed, the sensor is mounted at the laser tool position of the dummy head. The sensor tool calibration procedure is used to determine the laser tool transformation. The taught seam locations are replayed using the laser tool transformation. During this re-teaching phase, the sensor is used to measure the deviation of the laser tool frame with respect to the taught seam trajectory.

The used parameters during the experiments are shown in table 5.1.

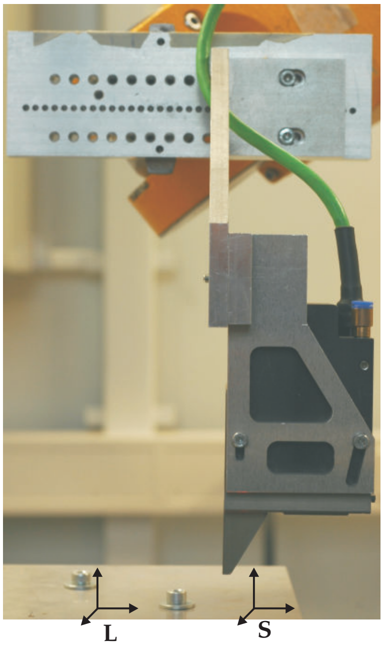


Figure 5.6: Dummy head. The sensor can be mounted on different positions with respect to the robot flange.

Sensor tool	[-258.032, -42.384, 162.278 mm 0, 90, -90 deg]
Laser tool	[-257.344, 12.25, 162.50 mm 0, 90, -90 deg]
Polyfit algorithm	Δs=2 mm, n=1, m=6, K_γ=0.25

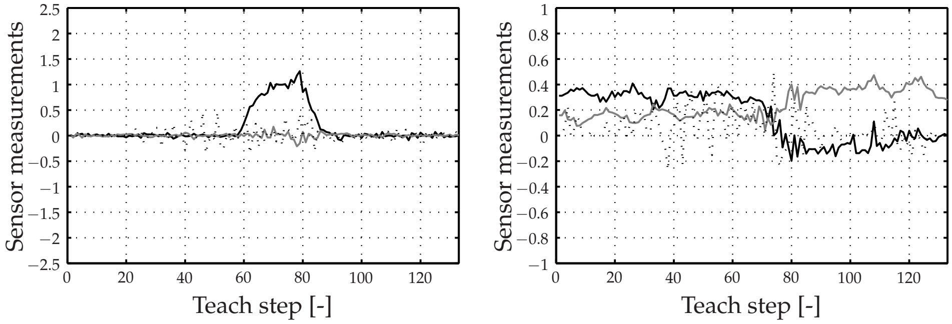
Table 5.1: Parameters used during the teaching experiments

5.5.1 Corner trajectory

The corner trajectory has been taught using the sensor tool transformation and replayed using the laser tool transformation. The sensor measurements during the teaching and re-teaching phase are shown in figure 5.7.

When the corner arrives during the teaching phase, the prediction of the seam trajectory starts to deviate from the actual seam trajectory. This results in a deviation of the s_y signal perpendicular to the seam trajectory. As the corner trajectory lies in a flat plane, the s_z signal does not significantly change. The measured rotation s_{rx} fluctuates a little bit, probably due to measurement noise. In the corner this fluctuation increases due to reflections of the sensor laser diode on the surface.

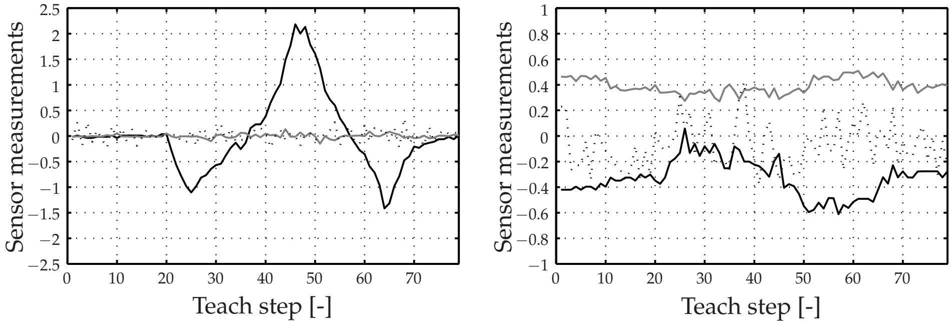
The sensor s_y measurement during the re-teaching phase show that the desired accuracy of 0.2 mm perpendicular to the seam trajectory is not met. The measurement s_z that represents the focal direction shows a deviation of up to 0.4 mm.



(a) Teaching

(b) Replaying

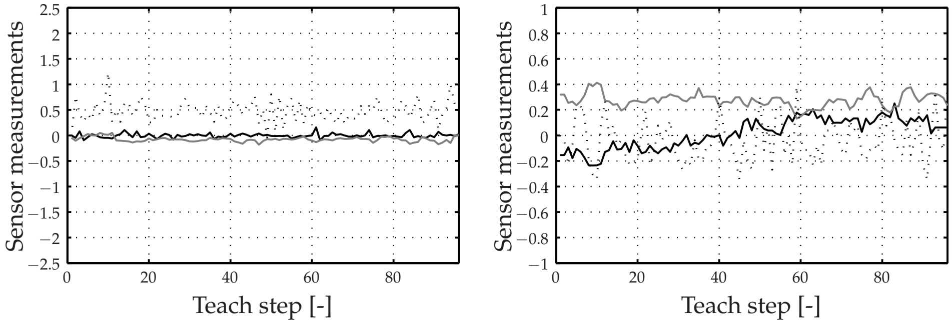
Figure 5.7: Experimental results of the teaching and re-teaching phase for the corner trajectory (— s_y [mm], — s_z [mm], · · · · s_{rx} [deg])



(a) Teaching

(b) Replaying

Figure 5.8: Experimental results of the teaching and re-teaching phase for the sine trajectory (— s_y [mm], — s_z [mm], · · · · s_{rx} [deg])



(a) Teaching

(b) Replaying

Figure 5.9: Experimental results of the teaching and re-teaching phase for the curved trajectory (— s_y [mm], — s_z [mm], · · · · s_{rx} [deg])

There may be several causes, that will be investigated more thoroughly later using simulations in section 5.6. The rotation measurement s_{rx} stays below 0.4 deg, which is not expected to cause any problems during laser welding.

5.5.2 Sine trajectory

The experimental results for the sine trajectory are shown in figure 5.8. During the teaching phase, the curvature of the sine trajectory results in a deviation of s_y. The s_z and s_{rx} measurements show expected behaviour.

During the re-teaching phase, the s_y sensor measurements shows a constant offset, during the straight parts of the sine trajectory. It fluctuates however at the curved parts. The s_z measurement shows an almost constant offset, which is probably be caused by a tool position error. The rotation measurement s_{rx} is roughly below 0.4 deg.

5.5.3 Curved trajectory

The results for the curved trajectory are shown in figure 5.9. During the teaching phase, the position measurements s_y and s_z stay close to zero as the curvature of the curved trajectory is only small. The rotation measurement s_{rx} shows an offset, as the orientation of the curved trajectory changes as well.

During the re-teaching phase, offsets in the position measurements s_y and s_z can be observed. The rotation measurement s_{rx} shows no unexpected behaviour.

The experiments that are carried out with the dummy head, show that the accuracy requirements of 0.2 mm perpendicular to the welding direction could not be achieved. In the next section, simulations are carried out, where various sources of errors are investigated that may have an influence on the accuracy of the laser spot with respect to the seam trajectory.

5.6 Simulation results and error analysis

The influence of various sources of errors is investigated using the simulation environment, because it allows to investigate the influence of single error sources. The errors, which have the most significant influence on the accuracy during welding are shown. In the simulations, a nominal tool definition for the Sensor and Laser tool is used, which is determined from the physical dimensions of the welding head and seam tracking sensor (table 5.2). The look-ahead distance between the Sensor tool and Laser tool is 55 mm. The tool axes are similar as in figure 3.3, where the x-axis is the welding direction. During the simulations, the same parameters have been chosen for the two teaching algorithms to be able to compare the different measurements.

Nominal Sensor tool	[-257,-43,163 mm 0, 90, -90 deg]
Nominal Laser tool	[-257,12,163 mm 0, 90, -90 deg]
Tractrix algorithm	Δs=2 mm, K_α=0.25, K_β=0.25
Polyfit algorithm	Δs=2 mm, n=1, m=6, K_γ=0.25

Table 5.2: Parameters used during the teaching simulations

5.6.1 Teaching and re-teaching

During the teaching phase, a virtual sensor is used that is mounted at the location of the Sensor tool. The measured seam locations and sensor values during the teaching phase are stored in the Sensor Trajectory Buffer and Sensor Measurement Buffer. Once the teaching phase has finished, the virtual sensor is mounted at the location of the Laser tool. The seam locations that were stored during the teaching phase are replayed using point-to-point movements. At each seam location, the virtual sensor is used as a measurement device to measure the deviation of the virtual laser tool frame to the seam trajectory. The location ${}^F_L\mathbf{T}_a$ of the sensor at the virtual Laser tool location is used in combination with the sensor model to compute the sensor measurements. It is computed as

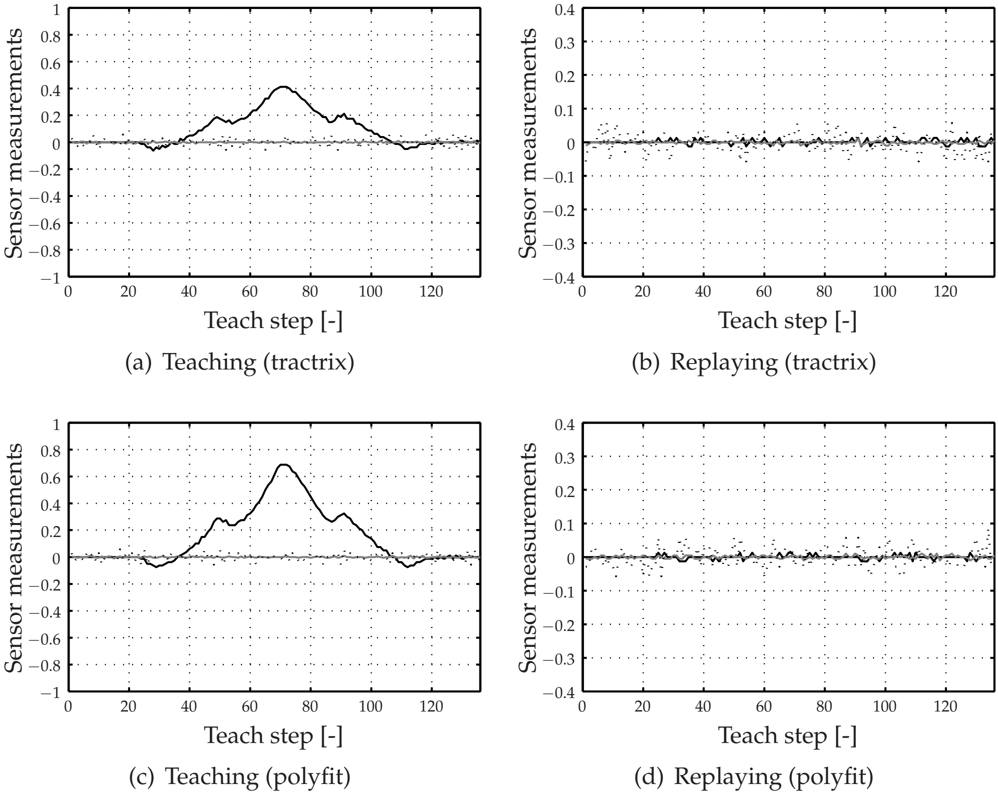
$$ {}^F_L\mathbf{T}_a = {}^B_F\mathbf{T}^{-1}\,{}^B_N\mathbf{T}_a \cdot {}^N_L\mathbf{T}_a. \tag{5.16} $$

First of all, simulations are performed where both the Sensor and Laser tool transformations are supposed to be perfectly known. Furthermore, no errors due to the robot geometric model are taken into account. Noise and quantisation are used on the sensor values, according to table 3.1. The corner, sine and curved seam trajectories will be taught using both the tractrix and polyfit teaching algorithm. Figure 5.10 shows the simulation results of the teaching and re-teaching phase for the corner trajectory.

Figure 5.11 shows simulation results for the sine trajectory.

During the teaching phase, the sensor values are close to zero at the straight parts of the seam trajectories. This is the case as the prediction of the next seam location is also on the seam trajectory. If the seam trajectories have a certain radius of curvature, there will be a deviation of the predicted next seam location to the seam trajectory. This deviation is measured by the sensor and is accounted for during the re-teaching phase. As expected, only the measurement errors due to sensor noise and quantisation are visible during this re-teaching phase, which shows the teaching algorithms work correctly for both 2D and 3D seam trajectories.

In the following sections, simulations have been performed to investigate the influence of tool transformation and robot geometry errors. The effects can be seen with all the mentioned seam trajectories and teaching algorithms. The most clear



(a) Teaching (tractrix)

(b) Replaying (tractrix)

(c) Teaching (polyfit)

(d) Replaying (polyfit)

Figure 5.10: Simulation results of teaching and replaying of the corner trajectory (Ideal) (— s_y [mm], — s_z [mm], · · ·· s_{rx} [deg])

results however are obtained with teaching the sine trajectory using the polyfit algorithm. Therefore this combination will be used in the following sections.

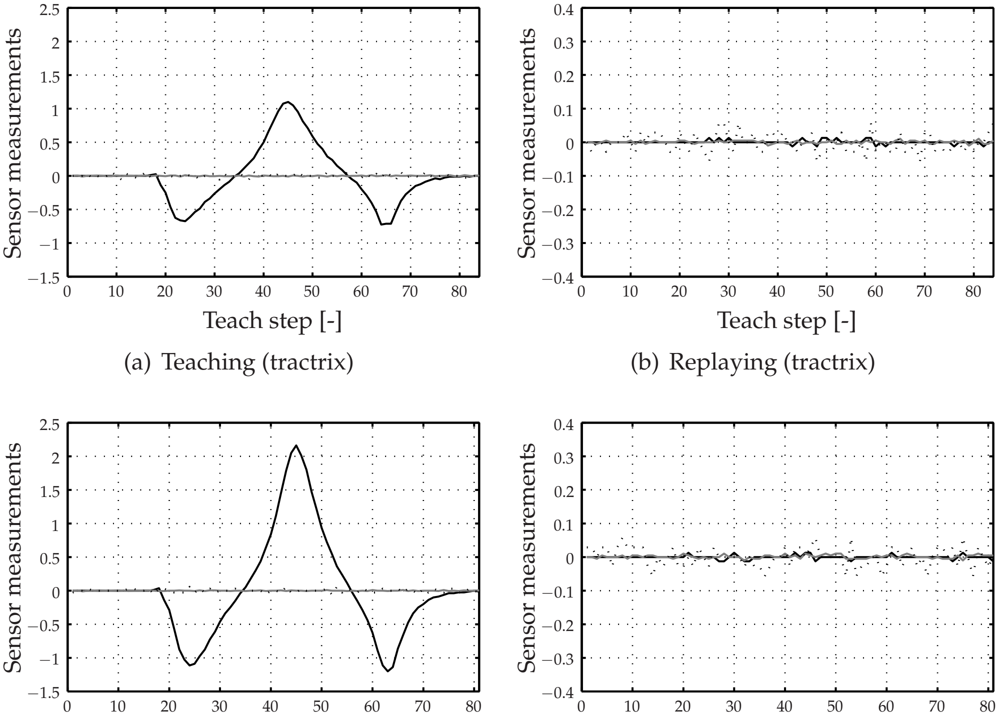
5.6.2 Tool transformation errors

This section shows the effect of errors in the used Sensor and Laser tool transformations.

Position error in Sensor tool transformation

The effect of a position error in the Sensor tool transformation is shown in figure 5.12. The actual Sensor tool transformation ${}^N_S T_a$ that is used is [-257.2, -43.2, 163.2 mm 0, 90, -90 deg]. It deviates a distance of -0.2, -0.2 and 0.2 mm in x, y, and z-direction of the nominal Sensor tool definition ${}^N_S T_n$.

A position error in the Sensor tool transformation does not have much influence during the teaching phase. It has a direct influence on the positional accuracy



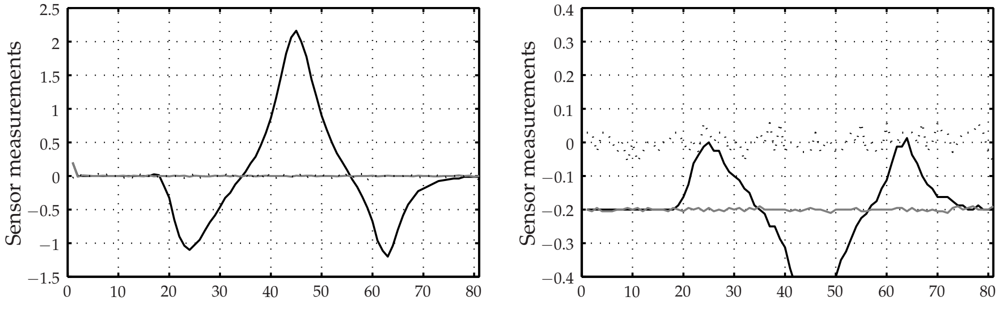
(a) Teaching (tractrix)

(b) Replaying (tractrix)

(c) Teaching (polyfit)

(d) Replaying (polyfit)

Figure 5.11: Simulation results of teaching and replaying of the sine trajectory (Ideal) (— s_y [mm], — s_z [mm], · · · · s_{rx} [deg])



(a) Teaching (polyfit)

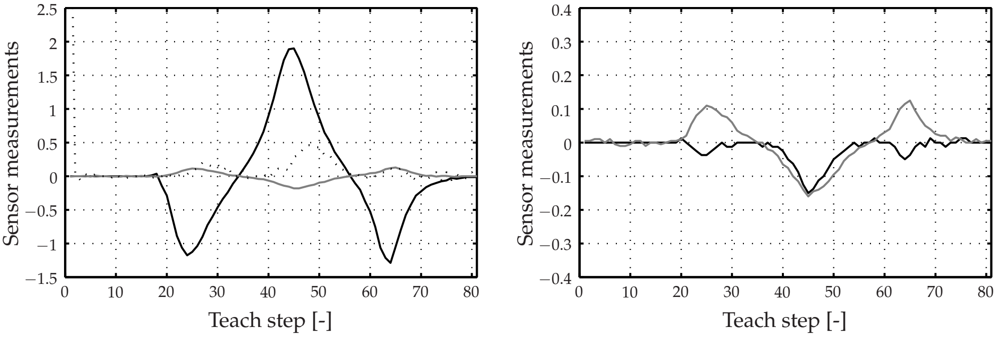
(b) Replaying (polyfit)

Figure 5.12: Simulation results of teaching and replaying of the sine trajectory (position error in Sensor tool transformation) (— s_y, — s_z, · · · · s_{rx})

during the re-teaching phase. Errors in different directions have a different effect. An error in the Sensor tool y-direction results in an offset of the s_y measurement on straight segments (start and end of the seam trajectory). An error in the x-direction is only visible on curved parts of the seam trajectory, because during re-teaching the Laser tool is moved too early or too late when a corner arrives. An error in the tool z-axis results in an error in the focal position of the laser. Generally, the s_y error perpendicular to the seam trajectory has the biggest effect on the welding quality and must be small. A position error does not have a noticeable effect on the orientation accuracy, because the seam teaching algorithm only makes small relative movements.

Orientation error in Sensor tool transformation

The effect of an orientation error in the Sensor tool transformation is shown in figure 5.13. The actual Sensor tool transformation ${}^{N}_{S}\mathbf{T}_a$ that is used is [-257 -43 163 5 95 -95]. The orientation deviates 5 degrees, around the different coordinate axes of the nominal Sensor tool definition ${}^{N}_{S}T_n$, which is a considerable deviation. It is expected that this is a worst-case deviation, the deviations that occur in practice should be smaller than this.



(a) Teaching (polyfit)

(b) Replaying (polyfit)

Figure 5.13: Simulation results of teaching and replaying of the sine trajectory (orientation error in the Sensor tool transformation) (— s_y [mm], — s_z [mm], $\cdots$ s_{rx} [deg])

Orientation errors have a noticeable effect on the sensor measurements during the teaching phase. As the orientation of the actual Sensor tool transformation differs from the nominal one, movements with respect to the nominal Sensor tool will result in a position error at the curved parts of the seam trajectory.

During the re-teaching phase, the orientation errors of the Sensor tool result in both position and orientation errors. The orientation offset s_{rx} is directly measured by the sensor. Position errors around 0.1 mm occur as a result of the used tool orientation error. It is expected that a smaller step size s decreases the effect of orientation errors in the Sensor tool, as the moved distance in a slightly wrong direction is smaller.

Laser tool transformation errors

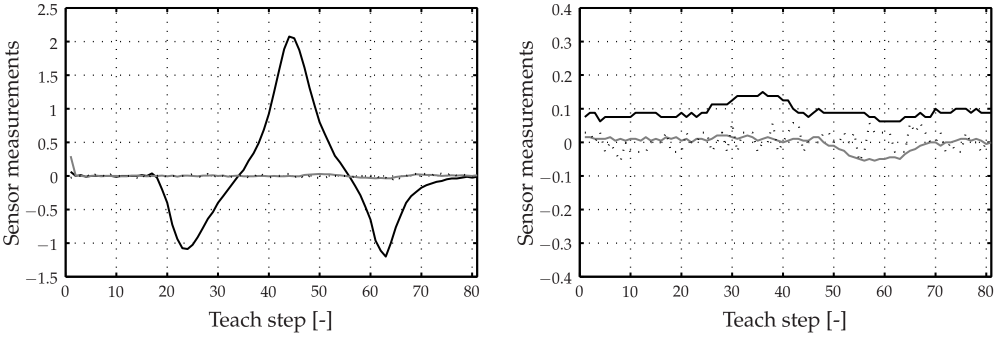
Errors in the position of the Laser tool transformation have a similar effect on the accuracy as a Sensor tool position error, because after a seam location is taught, the robot is moved to the wrong location during welding. Errors in the orientation of the Laser tool result in a constant orientation offset of the laser beam during welding.

5.6.3 Geometric robot errors

This section shows the effect of errors in the actual geometry of the robot arm. The three models (section 3.5.3) that are described by De Roo (2003) will be used in the following sections to compute the measurements using the sensor model (section 3.5.2).

Encoder offsets model

The average encoder offset error between the identified geometric model and the nominal kinematic model in the robot controller is about 0.06 degrees, which causes an absolute flange position error of almost 0.5 mm for the RX90 robot link length of 450 mm. The effect of using the encoder offsets geometric robot model on teaching and re-teaching is shown in figure 5.14.



(a) Teaching (polyfit)

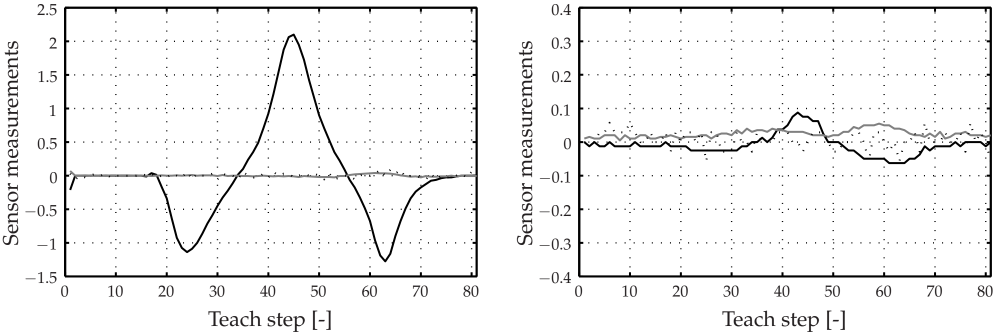
(b) Replaying (polyfit)

Figure 5.14: Simulation results of teaching and replaying of the sine trajectory (encoder offset geometric model) (— s_y [mm], — s_z [mm], $\cdots$ s_{rx} [deg])

The use of the encoder offsets model does not give a lot of difference in the teaching phase, so probably the orientation errors caused by the geometric robot errors are much smaller than the ones due to tool orientation errors. During the re-teaching phase, the position accuracy is shown to be badly influenced by encoder offset errors in the geometric robot model. The resulting position errors are about 0.1 mm. The biggest errors occur at the curved parts of the seam trajectory, where the robot has to make the largest joint movements.

Arm lengths model

The position errors in the robot tip are attributed to errors in the arm lengths of the robot model. This means that only the position of the robot tip is different from the nominal model, if this model is used. The effect of using the arm length robot model on teaching and re-teaching is shown in figure 5.15.



(a) Teaching

(b) Replaying

Figure 5.15: Simulation results of teaching and replaying of the sine trajectory (arm lengths geometric model) (— s_y [mm], — s_z [mm], · · ·· s_{rx} [deg])

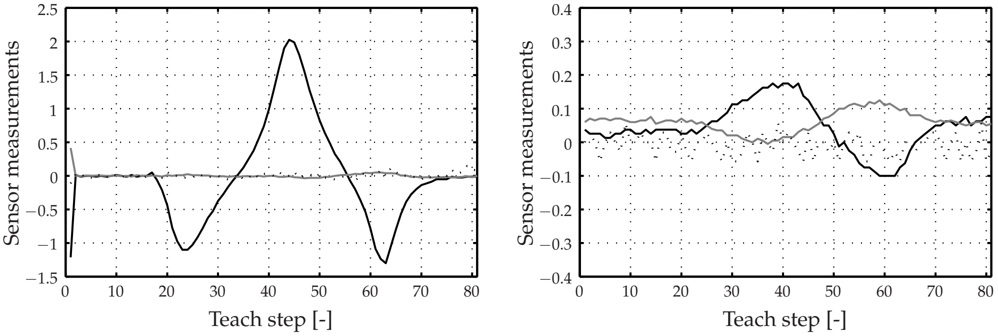
As mentioned there are no errors in the orientation of the robot tip, so therefore the teaching phase does not give much difference.

During the re-teaching phase, there are position errors up to 0.1 mm. At the straight seam parts, the position errors are small. The biggest errors occur at the curved parts of the seam trajectory, where the robot has to make the largest joint movements.

Link angles model

In this model, the errors in the robot tip are attributed to errors in the link angles of the robot model. The effect of using the link angles robot model on teaching and re-teaching is shown in figure 5.16.

Again there is nothing special to be seen during the teaching phase. During the re-teaching phase, the position errors are up to 0.2 mm. At the straight seam parts, the position errors are again small. The biggest errors occur at the curved parts of the seam trajectory, where the robot has to make the largest joint movements.



(a) Teaching

(b) Replaying

Figure 5.16: Simulation results of teaching and replaying of the sine trajectory (link angles geometric model) (— s_y [mm], — s_z [mm], $\cdots$ s_{rx} [mm])

5.7 Discussion

In this chapter the subject of seam-teaching using point-to-point movements is discussed. An adapted tractrix and a polyfit seam teaching algorithm have been described that can both be used for the automatic teaching of the geometry of a seam trajectory. The adapted tractrix algorithm is simple and straightforward, but in practice the polyfit teaching algorithm will only be used, because it gives the operator much more freedom on setting the filtering parameters.

A simulation environment has been developed, which allows to examine the influence of various errors that occur in a sensor-guided robotic laser welding environment. Simulations and real experiments have been performed, which shows that the teaching algorithms are working appropriately.

From the experiments and simulations several conclusions can be drawn:

- A position calibration error between the Laser and Sensor tool frame has a direct effect on the accuracy as it results in an offset during welding. A position error between the tools in y-direction results in an offset perpendicular to the seam trajectory. A position error in x-direction has an effect on the start and stop locations of the weld and has a larger effect at locations where the seam trajectory has a smaller radius of curvature. An error in z-direction results in an offset of the height of the laser focal point. It is therefore very important to calibrate the transformation between Laser and Sensor tool frame very accurately (better than 0.2 mm) to make robotic laser welding successful.

- Measurement errors in the sensor orientation result in a fluctuating robot motion during teaching. In simulation, this effect is much smaller than can be observed on the experimental setup. Non-linearities in the robot kinematics may play a role. For a further investigation, the simulation environment should be extended to include this effect.

- Geometric errors due to a different robot position between teaching and reteaching have a noticeable effect on the accuracy of the laser focal position. Geometric errors up to 0.2 mm occurred for the identified geometric robot models of De Roo (2003) and a look-ahead distance of 55 mm.

To decrease the effect of the geometric robot errors large robot joint movements should be avoided. This can be realised by choosing tool definitions that are close to the robot flange or by minimising the transformation between laser and sensor tool frame. The physical dimensions of the welding head and sensor usually determine the minimal distances that can be chosen. Several designs of welding heads are currently available or in development with integrated sensors to achieve close or overlapping sensor and laser tool frames (Iakovou et al., 2005; Falldorf Sensor, 2006). Another possibility to minimise joint movements is by preventing the orientation of the tool definition to change too rapidly as even small orientation changes of the tool can cause a major movement of the robot joints.

As explained in this chapter the accuracy requirements for laser welding make sensor-guided robotic laser welding more difficult, compared to conventional robotic welding applications like e.g. arc welding. The influence of all kind of errors that occur in a sensor-guided robotic laser is evaluated in this chapter. The simulation environment that is developed in this work is valuable for this purpose. Furthermore it is very convenient for testing and development of seam teaching algorithms.

Chapter 6

Real-time seam tracking

6.1 Introduction

The goal of this chapter is to use a seam tracking sensor that measures in front of the laser focal point for real-time seam tracking during laser welding. The difference with the previous chapter is that the sensor now measures simultaneously with the robot motion instead of measuring only when the robot is at rest, requiring the robot joint measurement and sensor image acquisition to be synchronised.

In chapter 5 it was shown that a location ${}^{F}_{G}\mathbf{T}$ on the seam trajectory relative to a generic frame **F** can be computed using equation 5.1. This equation can be written in its full form as

$$ {}^{F}_{G}\mathbf{T} = {}^{B}_{F}\mathbf{T}^{-1}\, {}^{B}_{N}\mathbf{T} \cdot {}^{N}_{S}\mathbf{T} \cdot {}^{S}_{G}\mathbf{T}, \tag{6.1} $$

where ${}^{B}_{F}\mathbf{T}$ is a known transformation from the robot base frame **B** to the generic frame **F** and ${}^{N}_{S}\mathbf{T}$ is a known fixed transformation from robot flange frame **N** to the sensor frame **S**. As shown in equation 3.29, transformation ${}^{B}_{N}\mathbf{T}$ can be computed from the robot joint angles. Transformation ${}^{S}_{G}\mathbf{T}$ is computed from a sensor measurement using equation 3.12. During the robot motion, both transformations ${}^{B}_{N}\mathbf{T}$ and ${}^{S}_{G}\mathbf{T}$ change in time. A seam location ${}^{F}_{G}\mathbf{T}$ can only be accurately computed if the sensor measurements and the robot joint measurements are known at the same time. This can be accomplished in two ways:

- Asynchronous: Let the robot perform a movement and wait until it stabilises. Because the robot is stabilised the location of the sensor frame in the robot workspace does not change in time. Then, a sensor measurement can be easily related to the corresponding robot position. This approach is used in chapter 5.

- Synchronous: The sensor measurement takes place during the robot motion. In this case, the time axes of the robot and the sensor need to be synchronised. If these are synchronised the robot joints can be interpolated to match the sensor measurement with the robot joints or vice versa.

Robot-sensor synchronisation can easily be achieved for a sensor that operates continuously and is interfaced directly to the robot, because the sensor data is immediately available when the robot needs it. Synchronisation becomes more challenging when complex sensor systems, like camera-based sensors with their own real-time clock are used. Section 6.5 describes the synchronisation mechanism that is used to synchronise the robot joint measurements with the image acquisition of the seam tracking sensor. Experiments have been carried out to determine the unknown time delay that is used in the synchronisation procedure.

In this chapter, a trajectory-based control strategy (section 6.3) is proposed. Instead of using the sensor measurements within the time-based control loop, which is often used in literature (section 6.2), the sensor measurements are related to the robot position to geometrically construct a seam trajectory. This control strategy is useful for real-time seam tracking where the sensor measures a certain distance ahead of the laser focal position where the measurements are used.

The trajectory-based control strategy uses a real-time setpoint generator, which computes smooth reference joint angles for the robot joint controller at fixed times. This setpoint generator is described in section 6.4.

Section 6.6 describe a real-time seam teaching algorithm that is used to accurately measure the seam trajectory during the robot motion. Furthermore, a real-time seam tracking algorithm is presented that uses a seam tracking sensor that measures ahead of the laser focal point to let the laser focal point track the just measured seam trajectory. Finally, conclusions are drawn.

6.2 Position-based control and image-based control

Visual servoing is the research area that considers the use of cameras and image-based sensors inside the control-loop of robots or manipulators. Seam tracking can therefore be considered as a form of visual servoing since an optical seam tracking sensor is used to correct the robot trajectory. This section describes the two control architectures (Sanderson and Weiss, 1980) that are distinguished in the visual servoing literature: position-based control and image-based control. The applicability of these control architectures for seam tracking will be discussed, which leads to a proposal for a new trajectory-based control architecture (section 6.3).

6.2.1 Position-based control

In position-based control, features are extracted from the sensor camera image and stored in a feature vector $\mathbf{f}^{\mathrm{m}}$. These features are used in conjunction with a geometric model of the target to determine the pose $\mathbf{p}^{\mathrm{m}}$ of the object with respect to the sensor. A control law is used to move the measured pose $\mathbf{p}^{\mathrm{m}}$ towards the desired pose $\mathbf{p}^{\mathrm{d}}$. Using the robot Inverse Geometric Model, the computed Cartesian locations are transformed to reference joint angles $\mathbf{q}_k^{\mathrm{d}}$, which are the reference input for a joint motion controller, which tracks the joint measurements $\mathbf{q}_k^{\mathrm{m}}$ in such a way that the tracking error $|\mathbf{q}_k^{\mathrm{d}} - \mathbf{q}_k^{\mathrm{m}}|$ on the specified path remains small.

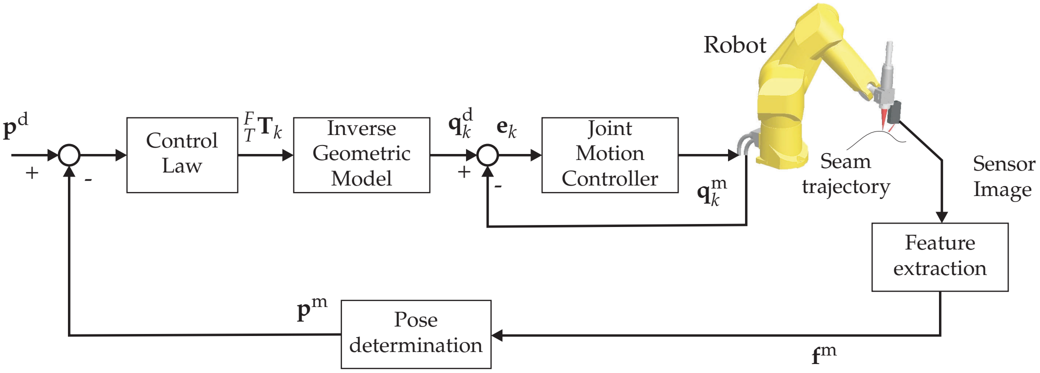


Figure 6.1: Block diagram of position-based control

6.2.2 Image-based control

In image-based servoing the pose determination step is omitted, and servoing is done on the basis of image features directly. Furthermore, the Inverse Geometric Model is not explicitly used, but the Control Law computes joint coordinates directly from the image features. This requires a complex Control Law, which uses the Jacobian matrix relating rate of change in pose to rate of change in feature space. The Control Law tracks the measured image features $\mathbf{f}^{\mathrm{m}}$ to the desired image features $\mathbf{f}^{\mathrm{d}}$. In general this Control Law is non-linear and cross-coupled such that the motion of one end-effector degree-of-freedom (DOF) will result in the complex motion of many camera features (Corke, 1994).

6.2.3 Discussion

The image-based approach may reduce computational delay, eliminate the necessity for image interpretation and eliminate errors in sensor modelling and camera calibration. However it does present a significant challenge to controller design since the plant is non-linear and highly coupled. An advantage of image-

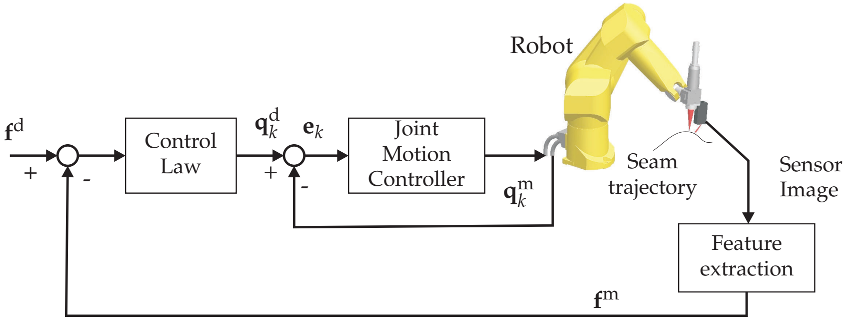

Robot
$\mathbf{f}^{\mathrm{d}}$
+
-
Control Law
$\mathbf{q}_k^{\mathrm{d}}$
$\mathbf{e}_k$
+
-
Joint Motion Controller
$\mathbf{q}_k^{\mathrm{m}}$
Seam trajectory
Sensor Image
Feature extraction
$\mathbf{f}^{\mathrm{m}}$

Figure 6.2: Block diagram of image-based control

based control is the relative robustness in the presence of image distortions and kinematic parameter variations in the manipulator Jacobian (Corke, 1994).

For seam tracking, the position-based approach seems the most straightforward one to use as both feature detection and pose determination is already performed by the sensor manufacturer. However, typical process application like welding and painting are normally done at a certain velocity, so an additional trajectory generator is needed to smoothly compute the desired pose $\mathbf{p}^{\mathrm{d}}$ at fixed time intervals in such a way that the velocity remains constant. Furthermore, seam tracking of 3D seam trajectories is considered, where laser welding is performed at high velocities. Using the position-based control approach this is difficult to achieve.

Both control architectures use the sensor measurements from a camera directly within the time-based control-loop of the robot controller. To guarantee stability, the time delays that occur in the system (communication time, image acquisition time and image processing time) are part of the feedback loop and need to be accurately known (Liu et al., 2004). Furthermore, the sensor and robot usually have different cycle times, so synchronisation and interpolation need proper attention.

To summarise, standard visual servoing techniques have a number of shortcomings if they are applied to seam tracking due to the high processing speeds and the 3D seam trajectories. Therefore a new trajectory-based control strategy is presented.

6.3 Trajectory-based control

Figure 6.3 shows a block diagram of the trajectory-based control strategy. The top part of this figure contains the robot trajectory generation, the bottom part contains the sensor integration part. Figure 6.3 will be explained, starting from the top left part.

The Tool Trajectory Buffer contains a list of locations (positions and orientations)

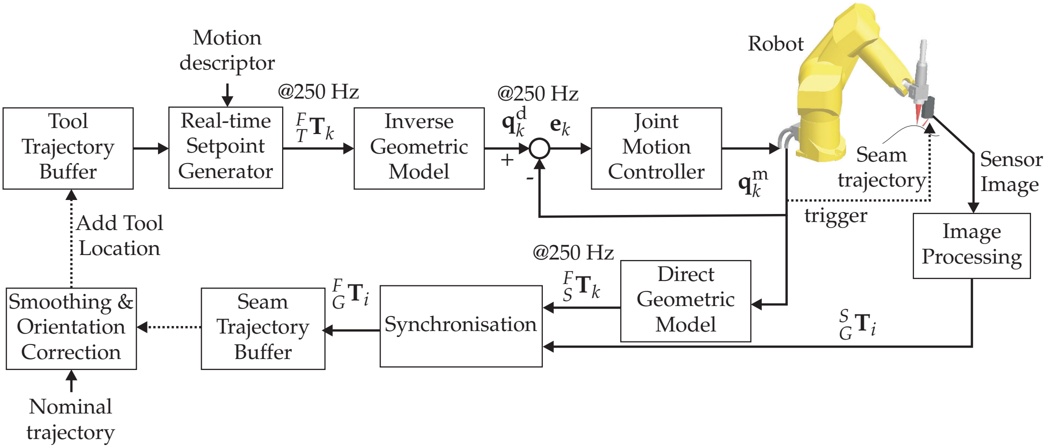


Figure 6.3: Block diagram of trajectory-based control

that the robot tool **T** (which can be **S** or **L**) has to pass through during the robot motion. These locations do not need to be equidistant. During the robot motion new locations can be added to the Tool Trajectory Buffer, thus extending the robot trajectory.

A real-time Setpoint Generator interpolates the locations in the Tool Trajectory Buffer and computes location setpoints $^F_T\mathbf{T}_k$ for every kth fixed time interval (4 ms in our case). The movement should be smooth as defined by the maximum acceleration, velocity and deceleration specified in the Motion Descriptor. A real-time Setpoint Generator that uses cubic interpolation based on quaternions is used, which is described in section 6.4.

From a location setpoint $^F_T\mathbf{T}_k$, the known tool transformation $^N_T\mathbf{T}$ and the known frame transformation $^B_F\mathbf{T}$, transformation $^B_N\mathbf{T}_k$ is calculated using equation 3.32. Next, a robot joint angle setpoint $\mathbf{q}_k^{\mathrm{d}}$ is calculated using the Inverse Geometric Model of equation 3.30.

The robot joint angle setpoints $\mathbf{q}_k^{\mathrm{d}}$ are the reference input for the joint motion controller, which is proprietary to Stäubli. It tracks the joint measurements $\mathbf{q}_k^{\mathrm{m}}$ in such a way that the tracking error $|\mathbf{q}_k^{\mathrm{d}} - \mathbf{q}_k^{\mathrm{m}}|$ on the specified path remains small.

The seam tracking sensor computes the transformation $^S_G\mathbf{T}_i$ from a camera image using image-processing, where i is a measurement index. If properly synchronised (section 6.5), $^S_G\mathbf{T}_i$ can be related to the measured joint angles $\mathbf{q}_k^{m}$ to compute a seam location $^F_G\mathbf{T}_i$ using equations 3.29 and 6.1. The result is stored in the Seam Trajectory Buffer. After the robot joints and sensor image are synchronised, the actual time upon which the seam locations were measured is not relevant anymore as the seam is defined by its geometric description only. Of course, the order in which the seam locations are obtained is of importance and it has to be assured that the number of points measured on the seam trajectory is sufficient in relation to the complexity of the seam. By moving the sensor tool along the seam trajectory and storing the obtained seam locations into the Seam Trajectory Buffer, the complete geometry of the seam trajectory is obtained. For real-time seam tracking, the Smoothing & Orientation Correction block is needed, which is described in section 6.6.

The trajectory-based control approach yields some major advantages, compared to the common time-based control approaches:

- The measured locations in the Seam Trajectory Buffer can be filtered in the position-domain instead of in the time-domain. This is more logical as seam trajectories are characterised by their radius of curvature. Curvature is meaningful in the position-domain, not in the time or frequency domain.
- Both historic and future information is available for filtering as the sensor measures ahead of the laser focal point. Phase-coherent or central filtering can be easily applied.
- The sensor image processing is not part of the robot motion control loop. The control structure is therefore independent of a variable delay after processing of the sensor image which may be caused by the image processing algorithm. Furthermore, stability issues are properly decoupled this way.
- It is easy to remove outliers. The sensor measurements may be incorrect (due to dust, spatter from the process, etc) or completely missing (due to sensor-robot communication errors). As long as the robot tool frame has not reached the end of the Tool Trajectory Buffer, the movement will continue.

The application of trajectory-based control for seam tracking is described in section 6.6, but first the Real-time Setpoint Generator and the used Synchronisation procedure are described in detail in the next sections.

6.4 Real-time Setpoint Generator

To use the trajectory-based control architecture, a real-time Setpoint Generator is needed. Several publications on trajectory generation are found in literature. Ahmed Bazaz and Tondu (1999) have presented a real-time trajectory generator that interpolates the trajectory in the joint-space. Khalil and Dombre (2002) describe a theoretical framework for real-time trajectory generation in Cartesian space. Unfortunately, only the positional part of the trajectory generation is worked out in detail in their framework. Abo-Hammour et al. (2002) describe Cartesian path generation using continuous genetic algorithms, which does not necessarily go exactly through the defined Tool Trajectory locations and lacks computational efficiency for real-time implementation. This section describes a real-time Setpoint Generator that is an extension of the trajectory generator of Sgarbi and Cammoun (1992). The trajectory is interpolated in Cartesian space and the velocity tracks a trapezoid profile. In their work, a main movement (e.g. translational) must be chosen and the other movements (e.g. rotational) are dependent. The real-time Setpoint Generator in this section is able to simultaneously choose limits on both the translational and rotational velocity that may differ on certain parts of the trajectory. Furthermore, these limits may be chosen differently on each segment of the trajectory.

The important requirements of the real-time Setpoint Generator are the generation of a smooth trajectory that has to be calculated in real-time. The interpolation function on a segment n between two locations ${}^F_T\mathbf{T}_n$ and ${}^F_T\mathbf{T}_{n+1}$ in the Tool Trajectory Buffer will be described using a cubic interpolation function to make sure that the first derivative is continuous between subsequent segments (figure 6.4). The interpolation function is determined by the interpolation parameter λ having a value between 0 (at the start of the segment) and 1 (at the end of the segment). To meet the real-time guarantees, simple and deterministic algorithms need to be used, iterative solutions do not suffice due to unknown computation times.

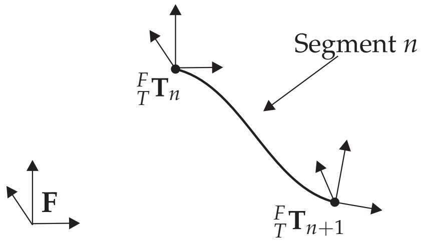


Figure 6.4: Cubic spline segment

6.4.1 Position interpolation

Because a cubic interpolation function is used, the position $\mathbf{p}_n$ on segment n as a function of the spline parameter λ can be written as

$$\mathbf{p}_n(\lambda) = \mathbf{a}_n\lambda^3 + \mathbf{b}_n\lambda^2 + \mathbf{c}_n\lambda + \mathbf{d}_n, \tag{6.2}$$

where the first derivative to λ is obtained as

$$\frac{d\mathbf{p}_n}{d\lambda}(\lambda) = 3\mathbf{a}_n\lambda^2 + 2\mathbf{b}_n\lambda + \mathbf{c}_n. \tag{6.3}$$

Four boundary conditions are used to determine the parameters $\mathbf{a}_n$, $\mathbf{b}_n$, $\mathbf{c}_n$ and $\mathbf{d}_n$:

$$\mathbf{p}_n(0) = {}^F_T\mathbf{P}_n, \qquad \mathbf{p}_n(1) = {}^F_T\mathbf{P}_{n+1}, \tag{6.4}$$

$$\frac{d\mathbf{p}_n}{d\lambda}(0) = {}^F_T\dot{\mathbf{P}}_n, \qquad \frac{d\mathbf{p}_n}{d\lambda}(1) = {}^F_T\dot{\mathbf{P}}_{n+1}, \tag{6.5}$$

where ${}^F_T\dot{\mathbf{P}}_n$ is computed using a central difference scheme using the next position ${}^F_T\mathbf{P}_{n+1}$ and previous position ${}^F_T\mathbf{P}_{n-1}$ from the Tool Trajectory Buffer as

$${}^F_T\dot{\mathbf{P}}_n^{\text{unscaled}} = \frac{{}^F_T\mathbf{P}_{n+1} +^F_T \mathbf{P}_{n-1}}{2}. \tag{6.6}$$

To be able to use this method for segments having a varying length, the vector length of ${}^F_T\dot{\mathbf{P}}_n^{\text{unscaled}}$ is scaled to the segment length with a scaling factor s_n to obtain

$${}^F_T\dot{\mathbf{P}}_n = s_n \cdot {}^F_T\dot{\mathbf{P}}_n^{\text{unscaled}}, \tag{6.7}$$

where

$$s_n = \frac{\left|{}^F_T\mathbf{P}_{n+1} -^F_T \mathbf{P}_n\right|}{\left|{}^F_T\dot{\mathbf{P}}_n^{\text{unscaled}}\right|}. \tag{6.8}$$

The linear velocity $v(t)$ can be evaluated at any point along the trajectory as

$$v(t) = \frac{d\lambda}{dt}\left|\frac{d\mathbf{p}_n}{d\lambda}(\lambda)\right|, \tag{6.9}$$

where $\frac{d\lambda}{dt}$ is the derivative of λ with respect to time. Using the bounds for the velocity from the Motion Descriptor, a maximum value $\frac{d\lambda}{dt}^{\max}$ can be calculated. The linear acceleration $a(t)$ can be approximated at any point along the trajectory as

$$a(t) = \frac{d^2\lambda}{dt^2}\left|\frac{d\mathbf{p}_n}{d\lambda}(\lambda)\right|. \tag{6.10}$$

Note that the second derivative of $\mathbf{p}_n$ to λ is neglected in this expression. It is small, provided that the higher order terms in equation 6.2 are small compared to the lower order terms. This is the case for moderately curved trajectory segments that are considered in this work.

6.4.2 Orientation interpolation

The orientation is interpolated in a similar way. In this thesis the cubic quaternion interpolation is used, which is used frequently in computer graphics (Shoemake, 1985; Barr et al., 1992; Hoffmann, 2002). The cubic interpolation function $Squad()$ on segment n is described in Eberly (1999) as

$$\begin{aligned} &Squad(\mathbf{Q}_n, \mathbf{Q}_{n+1}, \mathbf{Q}^i_n, \mathbf{Q}^i_{n+1}, \lambda)_n = \\ &\quad Slerp(\mathbf{Q}_n, \mathbf{Q}_{n+1}, \lambda)(Slerp(\mathbf{Q}_n, \mathbf{Q}_{n+1}, \lambda)^{-1} Slerp(\mathbf{Q}^i_n, \mathbf{Q}^i_{n+1}, \lambda))^{2\lambda(1-\lambda)}, \end{aligned} \tag{6.11}$$

where $Slerp()$ describes the *s*pherical *l*in*e*ar inte*r*polation between two unit quaternions, and $\mathbf{Q}_n$ and $\mathbf{Q}_{n+1}$ are the unit quaternions that describe the rotation of triads attached to the start and end of segment n, $\mathbf{Q}^i_n$ and $\mathbf{Q}^i_{n+1}$ are intermediate unit quaternions that make sure that the first derivative between successive segments is continuous, so the orientation changes smoothly. The spherical linear interpolation (Slerp()) between unit quaternions $\mathbf{Q}_n$ and $\mathbf{Q}_{n+1}$ is defined as

$$Slerp(\mathbf{Q}_n, \mathbf{Q}_{n+1}, \lambda) = \mathbf{Q}_n(\mathbf{Q}_n^{-1}\mathbf{Q}_{n+1})^{\lambda}. \tag{6.12}$$

The intermediate quaternions $\mathbf{Q}^i_n$ are chosen as (Eberly, 1999):

$$\mathbf{Q}^i_n = \mathbf{Q}_n exp\left(-\frac{log(\mathbf{Q}_n^{-1}\mathbf{Q}_{n+1}) + log(\mathbf{Q}_n^{-1}\mathbf{Q}_{n-1})}{4}\right). \tag{6.13}$$

The angular velocity $w(t)$ can be evaluated at any point along the trajectory as

$$w(t) = \frac{d\lambda}{dt}\frac{dr_n}{d\lambda}(\lambda), \tag{6.14}$$

where $\frac{dr_n}{d\lambda}(\lambda)$ is the differential tip rotation with respect to λ that is computed from equation 6.11 and its derivative to λ (Schwab, 2002). If $\mathbf{Q}$ is a quaternion and $\dot{\mathbf{Q}}$ is its derivative to λ, then

$$\frac{dr_n}{d\lambda}(\lambda) = |2\dot{\mathbf{Q}}\overline{\mathbf{Q}}|, \tag{6.15}$$

where $\overline{\mathbf{Q}}$ is the conjugate of $\mathbf{Q}$ (appendix B). The angular acceleration $\dot{w}(t)$ is approximated as

$$\dot{w}(t) = \frac{d^2\lambda}{dt^2}\frac{dr_n}{d\lambda}(\lambda). \tag{6.16}$$

Note that, similar as in equation 6.10, the second derivative of r_n to λ is neglected as it is small.

6.4.3 Determination of the interpolation parameter

The problem of trajectory generation is now reduced to finding an appropriate value of λ for every time step k that the joint motion controller needs a new setpoint. To generate a smooth trajectory, each λ_k should be computed by taking into account the bounds for linear and rotational acceleration, velocity and deceleration from the Motion Descriptor. The next value λ_{k+1} is calculated using

$$\lambda_{k+1} = \lambda_k + T_s \left. \frac{d\lambda}{dt} \right|_{k+1}, \tag{6.17}$$

where T_s is the used sample time and $\left.\frac{d\lambda}{dt}\right|_{k+1}$ is the desired derivative of λ with respect to time at time step $k+1$. Using equations 6.9 and 6.14, the maximum value $\frac{d\lambda}{dt}^{\max}$ can be evaluated every time step. Using equations 6.10 and 6.16, the maximum value $\frac{d^2\lambda}{dt^2}^{\max}$ can be evaluated. In this way the Cartesian bounds from the Motion descriptor are converted to bounds for the change of λ with respect to time. Using the acceleration bounds, the next value of $\left.\frac{d\lambda}{dt}\right|_{k+1}$ is bounded using

$$\text{abs}\left(\left.\frac{d\lambda}{dt}\right|_{k+1} - \left.\frac{d\lambda}{dt}\right|_{k} \right) < T_s \frac{d^2\lambda}{dt^2}^{\max}. \tag{6.18}$$

To reach a constant velocity the value of $\frac{d\lambda}{dt}$ is increased until it reaches $\frac{d\lambda}{dt}^{\max}$ while taking into account the acceleration bounds.

Because of the applied scaling in equation 6.7, the value of $\frac{d\lambda}{dt}$ must be scaled as well at the transition between the segments. The velocity, specified in equations 6.9 and 6.14, has to be continuous. At every transition from segment n to a next segment $n+1$, $\frac{d\lambda}{dt}_{k+1}$ is scaled as

$$\left.\frac{d\lambda}{dt}\right|_{k+1}^{n+1} = \left.\frac{d\lambda}{dt}\right|_{k+1}^{n} \cdot \frac{s_{n+1}}{s_n}, \tag{6.19}$$

which follows from the continuity of equation 6.9 at the end of segment n and the start of segment $n+1$.

A certain distance before reaching the last location in the Tool Trajectory Buffer, the Real-time Setpoint Generator should start to decelerate to make sure the velocity becomes zero at the endpoint while taking into account the bounds for deceleration. To achieve a constant deceleration, the length along the trajectory needs to be known, which can however not be analytically determined for 3D trajectories. To meet the real-time guarantees, the length and rotation along the trajectory are estimated using the absolute distance D^{end} and rotation r^{end} from the current location to the final location in the Tool Trajectory Buffer, which are defined as

$$\begin{array}{l} D^{\text{end}} = |\mathbf{p}_n(\lambda) -^F_T \mathbf{P}_{end}| \\ r^{\text{end}} = r_n(\lambda) - r_end \end{array}, \tag{6.20}$$

where ${}^F_T\mathbf{P}_{end}$ is the last position in the Tool Trajectory Buffer and r_end is the orientation angle of the last location in the Tool Trajectory Buffer. The Trajectory Generator starts the deceleration phase if D^{end} and r^{end} satisfy one of the following conditions:

$$\begin{array}{l} D^{\text{end}} <= \frac{v_k^2}{2a^{\max}} \\ r^{\text{end}} <= \frac{w_k^2}{2\dot{w}^{\max}}, \end{array} \tag{6.21}$$

where v_k and w_k are the current linear and rotational tip velocity and $a^{\max}$ and $\dot{w}^{\max}$ are the specified maximum linear and rotational tip deceleration from the Motion Descriptor. This close-to-end check has to be performed in real-time during every time step, therefore this simple and conservative method is used.

During the deceleration phase, either the position or the orientation is the limiting factor. If it is the position, the desired linear velocity v_{k+1}^d at the next setpoint is computed using

$$v_{k+1}^d = \sqrt{0.5 * D^{\text{end}} \cdot a^{\max}}, \tag{6.22}$$

and $\frac{d\lambda}{dt}_{k+1}$ is calculated as

$$\frac{d\lambda}{dt}_{k+1} = \frac{v_{k+1}^d}{|\dot{\mathbf{p}}_n(\lambda)|}. \tag{6.23}$$

If the orientation is the limiting factor, the desired rotational velocity w_{k+1}^d at the next setpoint is computed using

$$w_{k+1}^d = \sqrt{0.5 * r^{\text{end}} \cdot \dot{w}^{\max}}, \tag{6.24}$$

and $\frac{d\lambda}{dt}_{k+1}$ is calculated as

$$\frac{d\lambda}{dt}_{k+1} = \frac{w_{k+1}^d}{|\dot{\mathbf{r}}_n(\lambda)|}. \tag{6.25}$$

Summarised, the real-time Setpoint Generator performs the following computations at every time step k:

1. Check if the Trajectory Generator should start to slow down, using equation 6.21. If this is the case, compute $\frac{d\lambda}{dt}_{k+1}$ using equations 6.22, 6.23, 6.24 and 6.25.

2. Otherwise, the maximum values of $\frac{d\lambda}{dt}^{\max}$ and $\frac{d\lambda}{dt}^{\max}$ are evaluated using equations 6.9, 6.14, 6.10 and 6.16. Increase $\frac{d\lambda}{dt}_{k+1}$ until it reaches $\frac{d\lambda}{dt}^{\max}$, while taking into account equation 6.18.

3. Update λ_{k+1} using equation 6.17.

4. Compute the position and orientation of the next setpoint using equations 6.2 and 6.11.

6.4.4 Results

The trajectory-based control approach with the real-time Setpoint Generator have been implemented on an industrial Stäubli robot, with a CS8 controller and RTR-CAL software (Pertin and Bonnet-des-Tuves, 2004). The necessary calculations (Setpoint generation, inverse kinematics) are all performed within a real-time loop, with a sample time of 4 ms. This section shows several experiments that have been performed with the real-time Setpoint Generator.

In the experiments, Motion Descriptors are specified by the operator. They determine the maximum velocity, acceleration and deceleration along the trajectory. The operator uses the Motion Descriptor to specify the desired processing speed. The acceleration and deceleration values in the Motion Descriptor prevent sudden changes in velocity. The velocity, acceleration and deceleration values in the Motion Descriptor are specified separately for both translation and rotation.

Line trajectory

A line trajectory has been programmed (Figure 6.5), consisting of 5 locations (L_1 to L_5) and 4 segments. The linear velocity in the Motion Descriptor is varied at the segments. The linear acceleration and deceleration are limited to 1 $\mathrm{m/s^2}$. The trajectory consists of segments with varying lengths.

The trajectory has been performed and the linear tool velocity has been recorded. Figure 6.6 shows the ability of the Setpoint Generator to change the speed on a trajectory segment, according to the desired value in the Motion Descriptor, while maintaining the bounds for acceleration and deceleration. Furthermore it can be seen that the scaling factor of equation 6.8 is correctly applied for, as the segment lengths between segment 1 and 2 are different, but the velocity remains continuous.

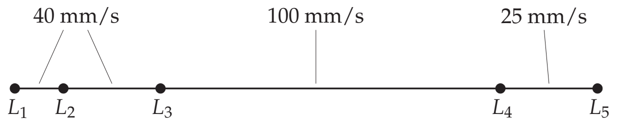


Figure 6.5: Line trajectory

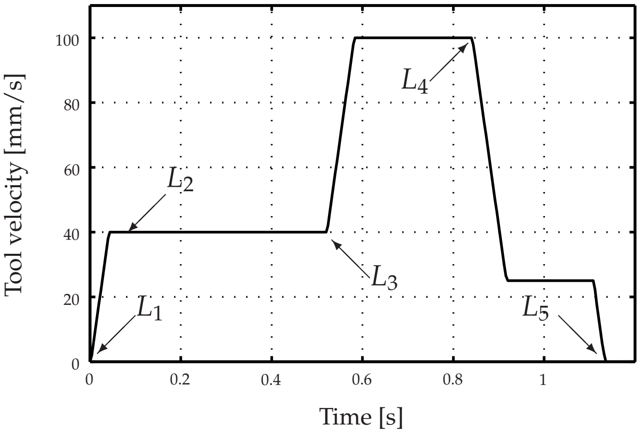


Figure 6.6: Linear tool velocity on the line trajectory

Corner trajectory

A corner trajectory (Figure 6.7) has been programmed consisting of 7 locations and 6 segments, with a radius *R* of 50 mm at the corner. The orientation in the corner changes with 90 degrees around the z-axis.

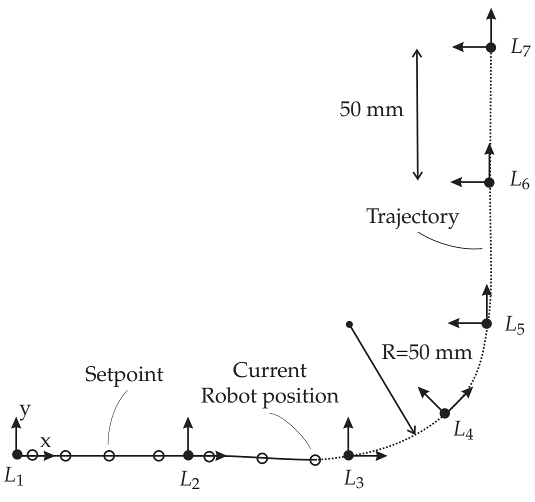


Figure 6.7: Corner trajectory

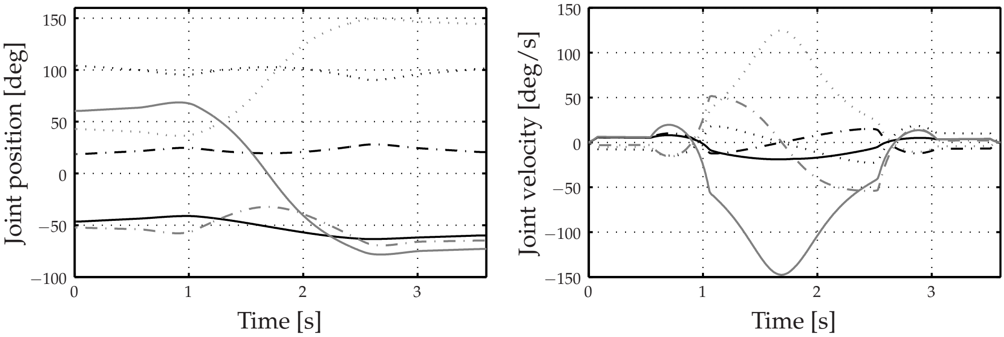
A Motion Descriptor is used, which limits the linear velocity to 100 mm/s (desired for the process) and the rotational velocity to 60 deg/s (to prevent large joint velocity in the corner). Furthermore the linear acceleration and deceleration are limited to 1 m/s^2. The joint position and joint velocity that are generated using the Setpoint Generator are given in figures 6.8(a) and 6.8(b).

It can be observed that both the joint angles and joint velocities are continuous, which gives a smooth robot motion. The reference and measured tool velocity (computed from the reference and measured joint positions and velocities) are given in figures 6.9(a) and 6.9(b). The robot accelerates and keeps the linear velocity at 100 mm/s. Because the robot needs to rotate in the corner, the linear velocity decreases, when the rotational velocity reaches its maximum of 60 deg/s. At the end of the trajectory the robot smoothly decelerates. The measured tip velocity shows a noisy behaviour, which is caused by numerical differentiation of $\mathbf{q}_k^{\mathrm{m}}$ to obtain the joint velocities.

Using the reference joint angles $\mathbf{q}_k^{\mathrm{d}}$ and measured joint angles $\mathbf{q}_k^{\mathrm{m}}$, an estimate of the tracking accuracy of the robot tip during this movement is made for the considered corner trajectory. This shows if the tracking accuracy is sufficient for the application of laser welding. To show how the tracking accuracy depends on the trajectory generation, the tip position error for a Stäubli RX130 robot without a tip load is shown in figure 6.10. The largest position errors (0.2 mm) can be observed in the welding direction (X-direction) during the acceleration and deceleration phases, which is the least critical error for moderately curved trajectories. The largest lateral (Y-direction) position error (0.1 mm) occurs when the joint velocity changes sign (around 1.7 s), which is a stick effect. The largest focal (Z-direction) position error (0.1 mm) occurs during the acceleration and deceleration phases. At the transition between the limiting linear velocity and rotational velocity, an acceleration of the joints results in a tip tracking error in all direction. The overall tracking performance is very satisfying for the application of laser welding, especially in the constant velocity regions it is mostly better than 0.05 mm.

6.5 Robot-sensor synchronisation

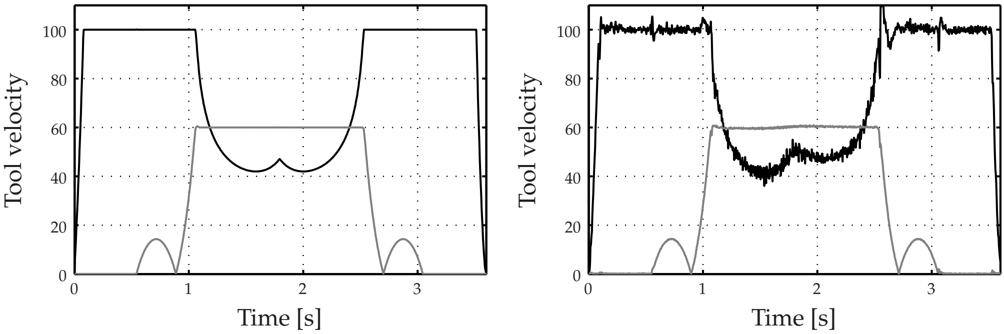
An essential component in the trajectory-based control strategy is the synchronisation between the robot joint measurements and the sensor image acquisition. Ethernet-based solutions for clock synchronisation of networked measurement and control systems exist (IEEE1588, 2002), but are complex and require dedicated hardware. Furthermore, standards like IEEE1588 are not widely adopted in industry yet. More specifically, it was not available for the sensor and robot platforms used in this work. Instead, a straightforward Ethernet-based synchronisation procedure has been implemented that offers exactly the needs as requested for trajectory based control. The seam tracking sensor is equipped with an addi-



(a) Joint position

(b) Joint velocity

Figure 6.8: Reference joint position and velocities for the corner trajectory (— Joint 1, -·- Joint 2, · · · · Joint 3, — Joint 4, -·- Joint 5, · · · Joint 6)



(a) Reference

(b) Measured

Figure 6.9: Reference and measured tip velocity for the corner trajectory (— Linear [mm/s], — Rotational [deg/s])

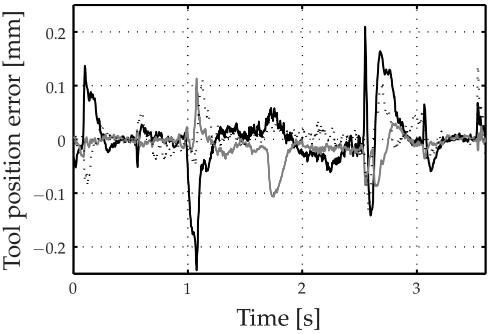


Figure 6.10: Tip position error, computed from the joint measurements (— X, — Y, · · · · Z)

tional network card, which is dedicated for synchronisation purposes. The time intervals are based on the equipment used, but the synchronisation principle is generic. Experiments have been performed to measure the fixed time delay that is needed in the synchronisation procedure. The determination of the time delay is necessary irrespective of the applied synchronisation method.

6.5.1 Synchronisation procedure

In figure 6.11 the used synchronisation procedure is shown. The robot controller and the seam tracking sensor have their own time axes. The measured robot joint angles are available at fixed time intervals T_r (4 ms in our case). The times $t_k^R = k \cdot T_r$ and corresponding robot joint measurements $\mathbf{q}_k^{\mathrm{m}}$ are stored in a cyclic buffer on the robot controller.

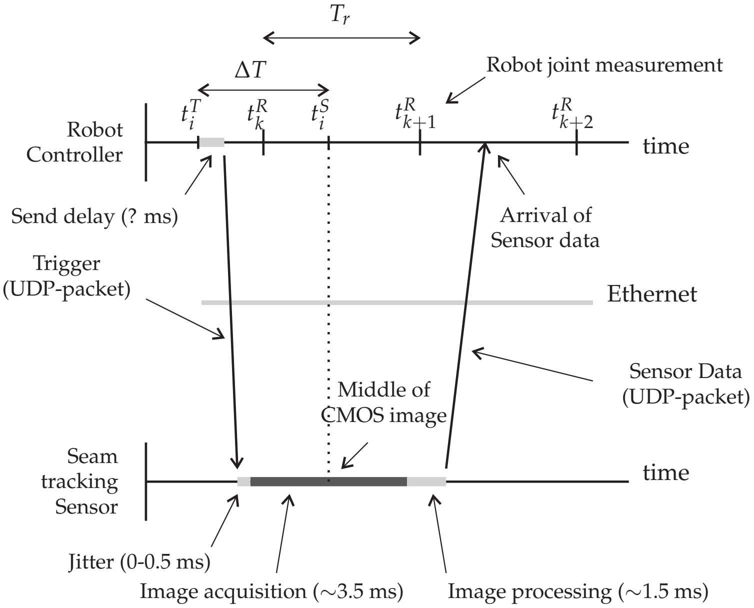


Figure 6.11: Synchronisation method

The acquisition of a single CMOS image is triggered, by sending a trigger UDP (User Datagram Protocol) packet to the sensor. The robot controller sends these trigger packets at times t_i^T, where i is the trigger packet index. On arrival of a trigger packet at the sensor, the image acquisition is started. The image acquisition takes (depending on the field-of-view) a certain time. The image processing time depends strongly on the chosen feature detection algorithm and CPU that is used for image processing. After the image processing has been completed, the sensor data ${}^S_G\mathbf{T}_i$ is transmitted to the robot controller. Both the trigger packet and the sensor data packet contain the index i. So once a sensor data packet arrives at the robot controller, the corresponding t_i^T is known.

During the robot motion, the sensor is normally positioned symmetrically with respect to the seam, implying that the seam point is located somewhere in the middle of the CMOS image. Since the CMOS chip is read column-wise from one side to the other, detection of the seam point will take place about half-way during the image acquisition. In figure 6.11 it can be seen that the estimated time t_i^S at which the sensor measurement took place is computed as

$$t_i^S = t_i^T + \Delta T, \tag{6.26}$$

where t_i^T is the time at which the trigger packet was sent to the sensor and ΔT is a time delay. In section 6.5.2 experiments are described to determine ΔT.
Let $\mathbf{q}_k^{\mathrm{m}}$ and $\mathbf{q}_{k+1}^{\mathrm{m}}$ be robot joint measurements at times t_k^R and t_{k+1}^R respectively and t_i^S lies in the interval between t_k^R and t_{k+1}^R. Then the robot joint angles $\mathbf{q}^{\mathrm{m}}(t_i^S)$ can be approximated using linear interpolation as

$$\mathbf{q}^{\mathrm{m}}(t_i^S) = \frac{(t_i^S - t_k^R)\mathbf{q}_k^{\mathrm{m}} + (t_{k+1}^R - t_i^S)\mathbf{q}_{k+1}^{\mathrm{m}}}{t_{k+1}^R - t_k^R}. \tag{6.27}$$

The synchronisation procedure is summarised as follows:

1. The robot controller sends trigger packets to the sensor at times t_i^T.
2. On arrival of the sensor data at the robot controller, the estimated time t_i^S when the image acquisition took place is calculated using equation 6.26.
3. Find the interval, where t_i^S lies between t_k^R and t_{k+1}^R.
4. Compute the interpolated joint angles $\mathbf{q}^{\mathrm{m}}(t_i^S)$ using equation 6.27.
5. A seam location ${}^F_G\mathbf{T}$ can be calculated from the interpolated joint angles and the received sensor data using equation 3.12 and 6.1. Only the order of incoming seam locations needs to be known to construct the seam trajectory.

The total time delay ΔT depends on the equipment used, but is expected to be constant with a certain amount of jitter (variation in time delay). In our case, ΔT consists of half the image acquisition time, which takes (depending on the field-of-view) about $\frac{3.5}{2} = 1.75$ ms for a full frame of 512 x 256 pixels. There is a communication delay, which is in the order of 0.1 ms. Furthermore, ΔT also accounts for the fact that the times t_k^R at which the joint measurements are available at the robot controller are different from the actual time that the encoders are measured. Because ΔT is experimentally determined, this delay (and others that may exist) are also taken into account.

The sensor checks at a rate of 4 kHz whether a trigger packet has arrived, which gives a jitter of 0.25 ms. There is another jitter of 0.25 ms before the image acquisition actually starts. Therefore, the total jitter between receiving the trigger packet and the start of the image acquisition is 0.5 ms. Although the use of the UDP-protocol on a switched network does not guarantee a fixed time delivery of packets on the network, packet delivery time is low (less than 0.1 ms) compared to the image acquisition time for a moderate network load and therefore the jitter caused by the UDP communication is also less than 0.1 ms. The total jitter in ΔT is therefore about 0.6 ms.

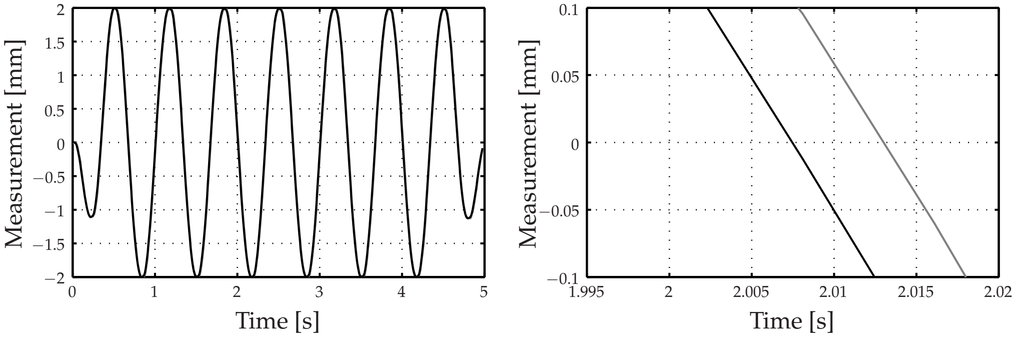
The presented synchronisation procedure has the following advantages:

- Only the arrival of the trigger packet is important, which makes the procedure independent of a varying image processing time after the image acquisition.
- The low jitter in the system makes it very suitable for the high accuracy requirements of laser welding.
- The trigger packets can be sent completely independently of the robot sample time. It is not necessary to send them at fixed time intervals.
- The interpolation is calculated after arrival of the sensor data at the robot controller. If a packet does not arrive (e.g. due to a communication error) it is automatically ignored.

6.5.2 Measuring the time delay

To measure the time delay ΔT, the following experiment has been performed. An object with a straight seam has been put in the middle of the field-of-view of the sensor. The sensor is moved perpendicular to the seam trajectory using a sine-motion with an amplitude of 2 mm. The measured displacements y^R of the sensor tool frame should now correspond to the measured displacements y^S of the sensor. The sensor readings y^S are recorded with respect to the times t_i^T and the sensor tool displacements y^R (calculated from the robot joints measurements) are recorded with respect to the times t_k^R. In figure 6.12(a) the result of this measurement for a frequency of 1.5 Hz is shown. The time delay between the two sine-measurements should correspond to the time delay ΔT. As expected both measurements closely fit. To see the time delay figure 6.12(b) shows the same measurement, but is zoomed in around 2 s.

The sensor measurements $y^S(t_i^T)$ are known at times t_i^T and the robot measurements $y^R(t_k^R)$ are known at times t_k^R. To be able to compare them, they need to be known with respect to a common time or index. Therefore, both the sensor



(a) Complete motion

(b) Zoomed around 2 s

Figure 6.12: Measured displacement of a sine motion with a frequency of 1.5 Hertz (— $y^R(t_k^R)$, — $y^S(t_i^T)$)

measurements and robot measurements are upsampled to 5 kHz. The common times are denoted by $t_j = j \cdot 0.2$ ms. The residual errors $e(t_j, \Delta T)$ are defined as

$$e(t_j, \Delta T)^2 = \left(y^S(t_j + \Delta T) - y^R(t_j)\right)^2, \tag{6.28}$$

where $y^S(t_j)$ and $y^R(t_j)$ are the upsampled sensor and robot measurements respectively and ΔT is a variable time delay. The mean-squared error (MSE) is a function of ΔT and is defined as

$$\mathrm{MSE}(\Delta T) = \frac{1}{n} \sum_{j=1}^{n} e(t_j, \Delta T)^2, \tag{6.29}$$

where n is the total number of measurements. The value of ΔT is sought, where the MSE has a minimum. For the sine motion at 1.5 Hertz the MSE as a function of ΔT is given in figure 6.13.

The value of ΔT where the MSE has a minimum is found with a resolution of 0.2 ms. To further increase the resolution a second order polynomial has been fitted through the MSE values to accurately find the minimum. For the sine motion at 1.5 Hertz, the time delay is found to be 4.9 ms using this method.

It should be noted that ΔT can only be correctly calculated if the robot measurements at the joint level correspond with the sensor measurement at the robot tip. This is the case when the robot accurately tracks the reference sine motion, but does not apply when the robot shows flexible behaviour of the joints and links, i.e. at high frequencies. On the other hand, at low frequencies, the signal-to-noise ratio is expected to be low as actually a phase shift is measured, which can not be

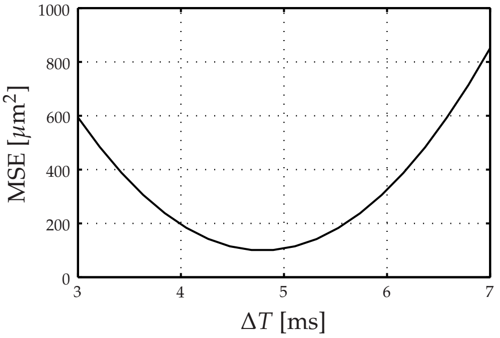


Figure 6.13: MSE as a function of the delay time for a sine motion with a frequency of 1.5 Hertz

interpreted accurately as a time delay at these frequencies. To see these effects, the time delay is calculated for sine motions at different frequencies ranging from 0.1 Hertz to 10 Hertz. In figure 6.14, the time delay is plotted as a function of the frequency.

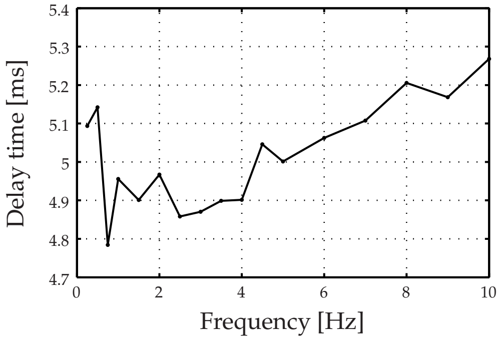


Figure 6.14: Computed delay time as a function of the frequency of the sine motion

An almost constant time delay is found for frequencies in the range of 1 Hertz to 4 Hertz. If the sine frequency is increased the delay time increases due to elastic robot behaviour. At low sine-frequencies the computed delay time becomes inaccurate due to the lower signal-to-noise ratio.

To show the accuracy of the synchronisation method, a delay time of $\Delta T = 4.9$ ms is used in the synchronisation procedure. A sine motion at 1.5 Hertz is carried out and the residues between the synchronised robot tip measurement and sensor measurement have been plotted in figure 6.15.

The maximum value of the residues is about 30 μm, which is in the same order as the robot repeatability. No remaining sine wave at 1.5 Hz can be observed, which shows that the delay time is correctly applied.

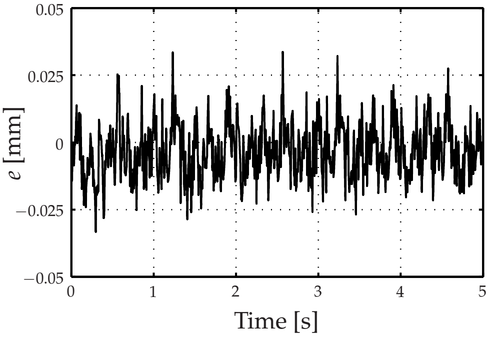


Figure 6.15: Residual error of the sine motion at 1.5 Hertz after synchronisation

The sine motions were performed on a static location on the seam trajectory. The accuracies that were obtained in these experiments can be translated to accuracies that occur during measurements of curved seam trajectories with a robot that is moving at typical laser welding speeds (figure 6.16).

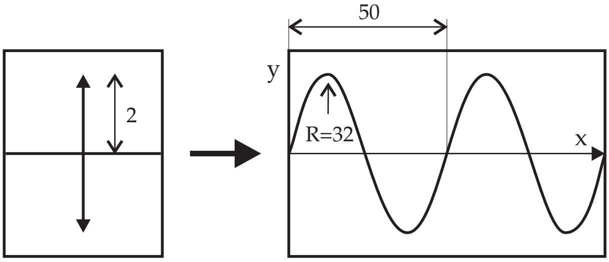


Figure 6.16: Static sine motion translated to curved sine trajectory

For a sine motion of 5 Hertz with an amplitude of 2 mm, the residues at this frequency are about 40 μm, which is well below the desired accuracy for laser welding. Suppose this sine motion was performed at a linear velocity of 250 mm/s, then in one period a distance x of 50 mm would have been travelled. The distance y perpendicular to the welding direction x can then be described as

$$y(x) = 2 \sin \frac{5 \cdot 2\pi \cdot x}{250}. \tag{6.30}$$

The radius of curvature R for a curve written in the form $y = f(x)$ (Kreyszig, 1991) can be computed using

$$R = \frac{[1 + \frac{dy}{dx}^2]^{\frac{3}{2}}}{|\frac{d^2y}{dx^2}|}. \tag{6.31}$$

The minimum radius of curvature is at the top (or bottom) of the sine, at x=12.5 mm or x=37.5 mm. It is computed as $R \approx 32$ mm. Suppose a synchronisation accuracy of 40 μm is desired at a welding speed of 250 mm/s, the radius of curvature should be larger than this. The computed radius of curvature is much better than the radius of curvature (typical value of 200 mm) that the robot can achieve, provided that the dynamic tracking accuracy stays in the order of 40 μm at these speeds. For lower welding speeds the radius of curvature that can be accurately measured is even better, which shows the accuracy of the proposed synchronisation method is more than sufficient for robotic laser welding.

6.6 Real-time seam tracking

6.6.1 Seam teaching and tracking

The trajectory-based control approach can be used for a number of different operations:

- Teaching a seam trajectory, with prior knowledge of its geometry, e.g. from CAD-data
- Teaching an unknown seam trajectory
- Real-time tracking a seam trajectory, with prior knowledge of its geometry
- Real-time tracking an unknown seam trajectory

The prior knowledge of the seam trajectory is included in the nominal seam trajectory, which is an approximation of the actual seam trajectory that the laser focal point has to track during laser welding. The nominal seam trajectory can be obtained in several ways:

- By manual programming
- From a previous seam-teaching procedure
- From a CAD file or off-line programming software

The seam locations of the nominal and actual seam trajectories are both stored in buffers that contain a number of seam locations. In practice, the nominal seam locations are expected to be within a few millimetres from the actual seam trajectory, which implies that if the sensor is positioned on the nominal seam trajectory, the actual seam trajectory is in the sensor's field-of-view.

The trajectory-based control structure only differs slightly for the mentioned seam teaching and seam tracking operations. In the case of teaching of a seam trajectory, with prior knowledge of its geometry, the Smoothing & Orientation

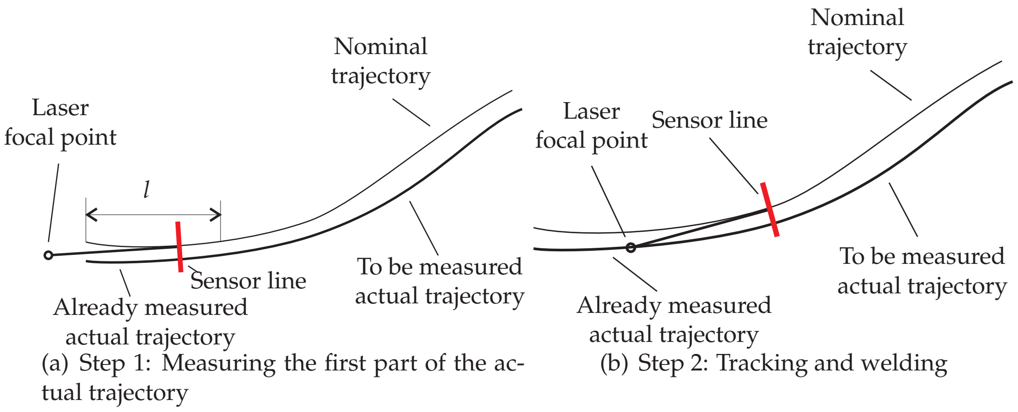
Correction block in figure 6.3 is not used, because the tool trajectory locations are known beforehand and the seam trajectory only needs to be recorded. For the other three procedures, the control loop has to be closed by on-line calculation and addition of locations to the Tool Trajectory Buffer. A proper smoothing must be taken care of to prevent oscillatory motion behaviour, because the trajectory-based control strategy contains no other bandwidth limitations. In the case of teaching an unknown seam trajectory, the Tool Trajectory Buffer needs to be filled with estimated seam locations, somewhere ahead of the current sensor location, which are extrapolated from the measured seam locations.

For real-time tracking of a seam trajectory, the sensor is used to obtain the location of the actual seam trajectory. The laser spot needs to be kept on this seam trajectory. Obviously, the seam trajectory has to remain in the sensors field-of-view, e.g. by rotating slightly around the laser tool. Real-time seam tracking without any prior knowledge of the geometry of the seam trajectory is considered to be impractical for safety reasons. In this work, real-time seam tracking is only performed when a nominal seam trajectory is already present to prevent the welding head to hit obstacles in the work cell due to erroneous sensor measurements, e.g. scratches. Only small deviations from a nominal trajectory are allowed. The nominal seam trajectory will be kept in the sensors field-of-view to make sure its measurements can be used.

6.6.2 Real-time seam tracking algorithm

The real-time seam tracking procedure can be divided into two subsequent steps (figure 6.17). First, the sensor is positioned at the starting point of the nominal seam trajectory. The sensor is moved along the nominal seam trajectory for a certain distance (the look-ahead distance), while the actual seam trajectory is measured. During this step, only the positioning of the sensor tool is of importance as laser processing does not take place yet. Once the first part of the actual seam trajectory is known, the laser focal point will be positioned at the starting point of the actual seam trajectory. During the second step, the measured actual seam trajectory is welded, while the sensor simultaneously tracks the nominal seam trajectory and measures the remaining part of the actual seam trajectory. Once the sensor has passed the end point of the nominal seam trajectory, no additional locations will be added to the actual seam trajectory anymore. The remaining part of the actual seam trajectory will be welded using the last known orientation.

1. To perform the first step, the look-ahead distance l has to be known, which is computed from the positional parts of the known sensor tool transfor-



(a) Step 1: Measuring the first part of the actual trajectory

(b) Step 2: Tracking and welding

Figure 6.17: Schematic overview of the real-time seamtracking algorithm

mation ${}^{N}_{S}\mathbf{T}$ and laser tool transformation ${}^{N}_{L}\mathbf{T}$ as

$$l = |{}^{L}_{S}\mathbf{P}| = |{}^{N}_{S}\mathbf{P} -{}^{N}_{L}\mathbf{P}|. \tag{6.32}$$

The locations of the nominal seam trajectory that are within a distance l from the start of the nominal seam trajectory are added to the Tool Trajectory Buffer using the sensor tool transformation ${}^{N}_{S}\mathbf{T}$. The sensor is moved along this trajectory, while measuring the actual seam locations. These locations are stored in the Seam Trajectory Buffer to be welded in the second part of the procedure.

During the second part, the laser focal point has to be focussed on any of these locations, while keeping the sensor on the nominal seam trajectory. Therefore the orientation of these locations has to be corrected (section 6.6.3). Furthermore, the corrected seam locations will be filtered (section 6.6.4), to smoothen the robot motion, to decrease the influence of sensor noise and to prevent unstable motion behaviour.

At the end of the first step, the corrected and filtered actual seam locations are added to the Tool Trajectory Buffer using the laser tool transformation ${}^{N}_{L}\mathbf{T}$. The laser focal point is moved to the first location in the Tool Trajectory Buffer, which corresponds to the start point of the actual seam trajectory.

2. In the second step, laser welding is performed and the laser focal point will move along the actual seam trajectory. During this movement, new seam locations are measured. The orientations of these seam locations are adjusted and filtered and the result is added to the Tool Trajectory Buffer. When the sensor moves past the end point of the nominal seam trajectory,

no more locations will be added to the Tool Trajectory Buffer and the process continues until the laser focal point reaches the end of the Tool Trajectory Buffer.

To maintain a safe operation, the following safety conditions are applied:

- Synchronised seam locations are only added to the Seam Trajectory Buffer if they are at a minimal distance (typically 0.2 mm) from the previous seam location to make sure the seam locations are not too close at low robot speeds, so filtering is always performed with locations that are at a minimal distance.
- Invalid sensor measurements are automatically ignored, because they are not added to the Seam Trajectory Buffer. If a series of invalid sensor measurements occurs, the robot motion will eventually automatically stop when the laser focal point reaches the end of the Tool Trajectory Buffer.
- Filtered seam locations are only added to the Tool Trajectory Buffer, if they are within 3 mm of the nominal seam trajectory.

6.6.3 Orientation correction

The orientation of the locations in the Seam Trajectory Buffer will be corrected in such a way that if the laser focal point is positioned at a seam location, the sensor tool frame intersects the nominal seam trajectory. This is illustrated in figure 6.18, where the sensor measures a seam location ${}^F_G\mathbf{T}_i$ on the actual seam trajectory.

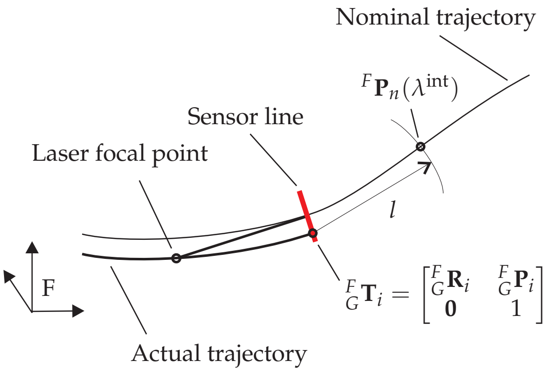

Nominal trajectory
${}^F\mathbf{P}_n(\lambda^{\text{int}})$
Sensor line
Laser focal point
l
F
${}^F_G\mathbf{T}_i = \begin{bmatrix} {}^F_G\mathbf{R}_i & {}^F_G\mathbf{P}_i \\ \mathbf{0} & 1 \end{bmatrix}$
Actual trajectory

Figure 6.18: Orientation correction during real-time seamtracking

When the laser focal point reaches location ${}^F_G\mathbf{T}_i$, its rotational part ${}^F_G\mathbf{R}_i$ should be such that the sensor intersects the nominal trajectory at position ${}^F\mathbf{P}_n(\lambda^{\text{int}})$, which is at a distance l from the origin of ${}^F_G\mathbf{T}_i$, so

$$|{}^F\mathbf{P}_n(\lambda^{\text{int}}) - {}^F_G\,\mathbf{P}_i| = l. \tag{6.33}$$

The intersection ${}^F\mathbf{P}_n(\lambda^{\text{int}})$ is computed numerically using the seam model of the nominal trajectory (section 3.5.1). For segment n on the nominal trajectory, which contains the intersection ${}^F\mathbf{P}_n(\lambda^{\text{int}})$, the following two conditions apply:

$$|{}^F\mathbf{P}_n -{}^F_G \mathbf{P}_i| \leq l, \tag{6.34}$$

and

$$|{}^F\mathbf{P}_{n+1} -{}^F_G \mathbf{P}_i| \geq l, \tag{6.35}$$

where ${}^F\mathbf{P}_n$ and ${}^F\mathbf{P}_{n+1}$ are the positions on the nominal trajectory at the start and end of segment n.

Using equations 3.7 and 3.17, the position ${}^F\mathbf{P}_n(\lambda)$ and its derivative $\frac{\partial}{\partial\lambda}{}^F\mathbf{P}_n(\lambda)$ can be evaluated for any value of λ. An iterative Newton-Raphson root-finding method (Abramowitz and Stegun, 1972) is used to numerically compute the value of λ^{int} on the segment n that satisfies equation 6.33. At every iteration step j, a new value λ_{j+1} is computed as

$$\lambda_{j+1} = \lambda_j - \Delta\lambda_j, \tag{6.36}$$

where

$$\Delta\lambda_j = \frac{|{}^F\mathbf{P}_n(\lambda_j) -{}^F_G \mathbf{P}_i| - l}{|\frac{\partial}{\partial\lambda}{}^F\mathbf{P}_n(\lambda_j)|}, \tag{6.37}$$

until the following convergence criterium is met

$$||{}^F\mathbf{P}_n(\lambda_{j+1}) -{}^F_G \mathbf{P}_i| - l| < \epsilon. \tag{6.38}$$

The convergence error denoted by ϵ is a small number, that defines the accuracy at which ${}^F\mathbf{P}_n(\lambda^{\text{int}})$ is found (typically 0.1 mm). In order to improve the convergence speed and robustness of the algorithm, the interval $\Delta\lambda_j$ is divided by two if

- The computed λ_{j+1} is not on the segment between 0 and 1.
- The convergence error increased, so if $||{}^F\mathbf{P}_n(\lambda_{j+1}) -{}^F_G \mathbf{P}_i| - l| > ||{}^F\mathbf{P}_n(\lambda_j) -{}^F_G \mathbf{P}_i| - l|$.

Once the intersection ${}^{F}\mathbf{P}_n(\lambda^{\text{int}})$ has been found, the remaining step is to compute the orientation. The corrected orientation ${}^{F}_{G}\mathbf{R}_i^{\text{cor}}$ can be computed from the uncorrected orientation ${}^{F}_{G}\mathbf{R}_i^{\text{uncor}}$ as

$${}^{F}_{G}\mathbf{R}_i^{\text{cor}} = {}^{F}_{G}\mathbf{R}_i^{\text{uncor}}\mathbf{R}^{\text{rel}}. \tag{6.39}$$

If the origin of the laser tool is positioned at location ${}^{F}_{G}\mathbf{T}_i$, the intersection ${}^{F}\mathbf{P}_n(\lambda^{\text{int}})$ can be computed relative to the laser frame using

$$\begin{bmatrix} {}^{L}\mathbf{P}_n(\lambda^{\text{int}}) \\ 1 \end{bmatrix} = {}^{F}_{G}\,\mathbf{T}_i^{-1} \begin{bmatrix} {}^{F}\mathbf{P}_n(\lambda^{\text{int}}) \\ 1 \end{bmatrix}. \tag{6.40}$$

The position ${}^{L}_{S}\mathbf{P}$ of the sensor frame relative to the laser frame is computed from the known laser tool transformation ${}^{N}_{L}\mathbf{T}$ and sensor tool transformation ${}^{N}_{S}\mathbf{T}$ using equation 4.2. Obviously, the laser tool location has to be adjusted such that the local position of the sensor frame ${}^{L}_{S}\mathbf{P}$ coincides with the intersection ${}^{L}\mathbf{P}_n(\lambda^{\text{int}})$. Using the vectors ${}^{L}\mathbf{P}_n(\lambda^{\text{int}})$ and ${}^{L}_{S}\mathbf{P}$, the orientation can be expressed as a rotation with angle θ around a unit 3D vector $\mathbf{u}$, where the angle θ between the vectors is computed as

$$\theta = \arccos\left(\frac{{}^{L}\mathbf{P}_n(\lambda^{\text{int}})}{|{}^{L}\mathbf{P}_n(\lambda^{\text{int}})|} \bullet \frac{{}^{L}_{S}\mathbf{P}}{|{}^{L}_{S}\mathbf{P}|} \right), \tag{6.41}$$

and the unit vector $\mathbf{u}$, which is normal to both vectors is computed as

$$\mathbf{u} = \frac{{}^{L}\mathbf{P}_n(\lambda^{\text{int}}) \times {}^{L}_{S}\mathbf{P}}{|{}^{L}\mathbf{P}_n(\lambda^{\text{int}}) \times {}^{L}_{S}\mathbf{P}|}. \tag{6.42}$$

Obviously no rotation correction is needed if $\theta \approx 0$, which means that the two vectors are already in line with each other. Using section 3.3.1, the relative rotation matrix $\mathbf{R}^{\text{rel}}$ can be computed from $\mathbf{u}$ and θ.

In practice, there are two cases in which there is no intersection of the sensor with the nominal seam trajectory.

- If the length of the nominal seam trajectory is smaller than the look-ahead distance, the sensor has already passed the end of the nominal seam trajectory once the laser reaches the start of the actual trajectory. In this case no orientation correction is applied.
- If the distance between the laser tool and the end of the nominal seam trajectory is smaller than the lookahead distance, the sensor has passed the end of the seam trajectory. In this case, the last applied orientation correction is maintained.

6.6.4 Filtering

The corrected seam locations contain noise and may be close to each other, especially if the robot moves at slow speeds. They can not be directly added to the Tool Trajectory Buffer, because by design, the trajectory-based control strategy does not contain bandwidth limiting components. This would result in an irregular robot motion as the Setpoint Generator interpolates a tool trajectory that goes exactly through the specified Tool Trajectory locations.

Therefore, the corrected seam locations will be filtered and only added to the Tool Trajectory Buffer if they are a certain distance from each other. The positions and orientations in the Seam Trajectory Buffer are filtered separately. For the position, a polynomial of order n will be used, which has the form

$$\mathbf{P}(p) = \mathbf{a}_n p^n + \mathbf{a}_{n-1} p^{n-1} + \ldots + \mathbf{a}_1 p + \mathbf{a}_0. \tag{6.43}$$

In practice, a polynomial order $n = 2$ is used. The filtered seam locations are not needed until the laser focal point reaches them. Therefore, both historic and future seam locations are available and a phase-coherent central filtering is used. To filter the position at index p, $m+1$ position vectors from the Seam Trajectory Buffer $[{}^F_G\mathbf{P}_{p-m/2} \ \ldots \ {}^F_G\mathbf{P}_p \ \ldots \ {}^F_G\mathbf{P}_{p+m/2}]$ are used to find the polynomial coefficients $\mathbf{a}_0$ to $\mathbf{a}_n$. These coefficients are computed in such a way that the error criterium

$$\sum_{j=p-\frac{m}{2}}^{p+\frac{m}{2}} |\mathbf{P}(j) - {}^F_G\mathbf{P}_j|, \tag{6.44}$$

is minimised in a least squares sense. Using the position vectors ${}^F_G\mathbf{P}_p$, matrix equation 6.43 can be written in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$:

$$\begin{bmatrix} (p-m/2)^n & \cdots & (p-m/2) & 1 \\ \vdots & & \vdots & \vdots \\ (p)^n & \cdots & (p) & 1 \\ \vdots & & \vdots & \vdots \\ (p+m/2)^n & \cdots & (p+m/2) & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}_n \\ \vdots \\ \mathbf{a}_0 \end{bmatrix} = \begin{bmatrix} {}^F_G\mathbf{P}_{p-m/2} \\ \vdots \\ {}^F_G\mathbf{P}_p \\ \vdots \\ {}^F_G\mathbf{P}_{p+m/2} \end{bmatrix}. \tag{6.45}$$

In the general case that matrix $\mathbf{A}$ is not square, the polynomial coefficients $\mathbf{x} = [\mathbf{a}_n \cdots \mathbf{a}_0]$ are found using the pseudo-inverse of matrix $\mathbf{A}$ as

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \cdot \mathbf{b}, \tag{6.46}$$

and the filtered position at index p is computed using equation 6.43. It should be noted that the above mentioned procedure may theoretically only be used for equidistant positions, which may not be the case if sensor measurements are missing or if the sensor velocity changes along the trajectory. Because the number of points used for filtering is large compared to the filter order, the effect of this simplification is small in practice.

To filter the orientations of the seam locations a simple but effective approach is used. At index p, the orientations from index $p - m/2$ to index $p + m/2$ are available, therefore the components of the quaternion representations of the $m + 1$ orientations are simply averaged and normalised, which nicely smoothes the orientation. Suppose $\mathbf{Q}_p$ is the quaternion representation of rotation matrix ${}^{F}_{G}\mathbf{R}_p$ at index p, the quaternion average $\mathbf{Q}_p^{\text{avg}}$ equals

$$\mathbf{Q}_p^{\text{avg}} = \frac{\sum_{j=p-\frac{m}{2}}^{p+\frac{m}{2}} \mathbf{Q}_j}{m+1}, \tag{6.47}$$

and the normalised quaternion average $\mathbf{Q}_p^{\text{navg}}$ equals

$$\mathbf{Q}_p^{\text{navg}} = \frac{\mathbf{Q}_p^{\text{avg}}}{|\mathbf{Q}_p^{\text{avg}}|}. \tag{6.48}$$

6.6.5 Limitations

The following limitations need to be kept in mind, when using the trajectory-based control approach for real-time seam tracking:

- The seam trajectory can not be tracked if no solution exists for correcting the orientation of a seam location in such a way that the sensor is positioned on the nominal seam trajectory. This may happen if the curvature at some parts of the seam trajectory is smaller than the look-ahead-distance.

- If the current robot location is close to the last location in the Tool Trajectory Buffer, the Setpoint Generator will start the deceleration phase and will not allow to add locations to the Tool Trajectory Buffer anymore. The risk that this may happen is larger if the look-ahead distance is small at high robot speeds or if the average time for calculating the orientation correction, filtering and safety checks takes more than the available CPU-time.

6.6.6 Experimental results

In this section, the results of several teaching and tracking experiments will be presented and compared. In these experiments, the dummy laser welding head (section 5.4) will be used to mount the sensor on two different locations on the robot flange, so it can be used for teaching and tracking, but also for measuring the accuracy of the laser position in a separate movement. Three different seam trajectories will be used:

- Line trajectory
- Corner trajectory
- Curved Sine trajectory

3D plots of these seam trajectories are given with the experiments.

For each trajectory, both a ′teaching′ and a ′tracking′ experiment are performed. During the ′teaching′ experiments, the actual seam trajectory is found by moving the sensor along the nominal seam trajectory. During this movement, the measured seam locations are stored in the Seam Trajectory Buffer. After the movement has completed, the seam locations in the Seam Trajectory Buffer are filtered and added to the Tool Trajectory Buffer. The taught trajectory can be welded in an additional processing step by moving the laser along the locations in the Tool Trajectory Buffer. Compared to the teaching procedures in chapter 5 this gives a considerable increase in teaching speed as teaching is done during the robot motion instead of using point-to-point movements.

Different parameters have been used in the three teaching experiments. These parameters are shown in table 6.1.

Parameter	Line	Corner	Curved
Velocity (mm/s)	100	30	50
Seam location distance (mm)	0.2	0.2	0.2
Add location distance (mm)	10	3	3
Fit order (eq. 6.43)	2	2	2
Fit points m	32	32	32

Table 6.1: Parameters used during the teaching experiments

In the ′tracking′ experiments, the orientations of the measured seam locations are corrected and the result is filtered and added to the Tool Trajectory Buffer in one processing step. Real-time tracking of the corner trajectory did not succeed as the curvature of this trajectory was quite small, so that no correct sensor measurements were obtained anymore when the sensor reached the corner. The parameters of the real-time tracking experiments are given in table 6.2.

Parameter	Line	Corner	Curved
Velocity (mm/s)	100	n/a	50
Seam location distance (mm)	0.2	n/a	0.2
Add location distance (mm)	10	n/a	3
Fit points m	32	n/a	32

Table 6.2: Parameters used during the real-time tracking experiments

Line trajectory

The nominal trajectory for the line trajectory consists of two seam locations, one at the start and one at the end of the trajectory. The sensor measurements during teaching are shown in figure 6.19(b). The sensor measurements are a few millimetres distinct from zero, which shows that the nominal trajectory was at a few millimetres distance from the actual seam trajectory. Furthermore the curve in the s_z measurement shows that the trajectory was bent a little.

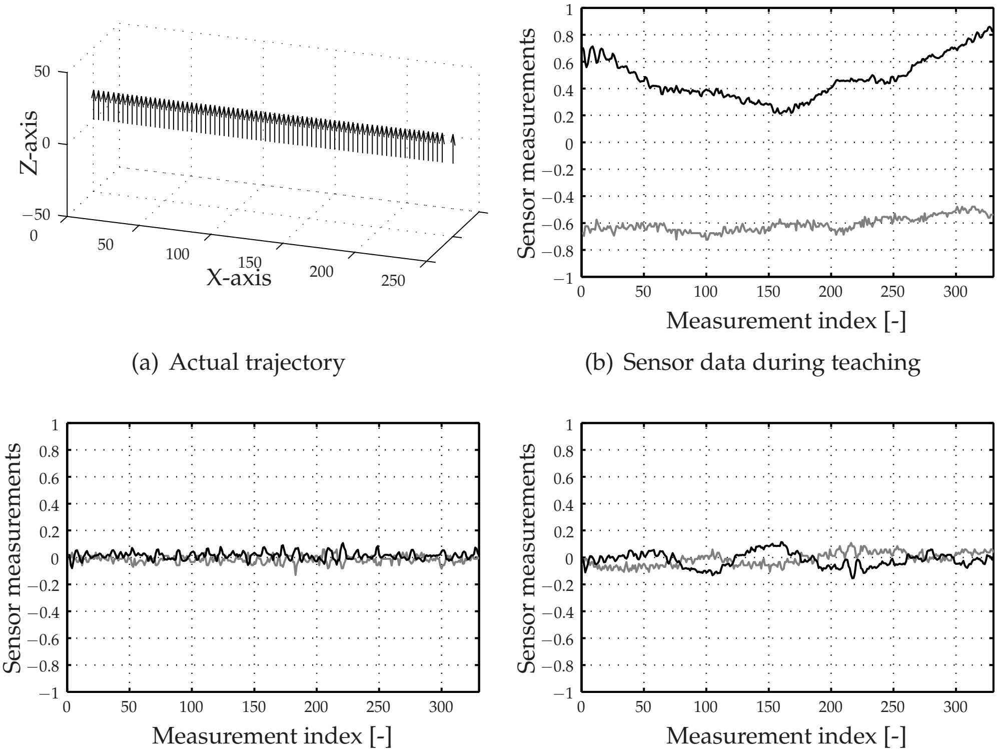
Teaching experiment

After the 'teaching' experiment has completed, the taught trajectory has been replayed with the sensor still mounted at the sensor tool position of the dummy welding head. Figure 6.19(c) shows that the sensor measurements during replay with the sensor tool are within 0.05 mm, approaching the robot repeatability (0.025 mm). This shows that the teaching procedure is correct and that accurate welding results can be obtained with a welding head, which has the laser and sensor tool at the same position.

Next, the taught trajectory has been replayed with the sensor mounted at laser tool position of the dummy welding head. Figure 6.19(d) shows that the sensor measurements are mainly within 0.1 mm, which shows that the welding accuracy is well suited for laser welding. The slight increase in the measurements is caused by the fact that the seam locations are measured with a different tool transformation than the one that is used during replay. Therefore the robot joint positions during replay are also slightly different, which causes a small geometric effect on the accuracy.

Tracking experiment

After the real-time tracking procedure has completed, the tracked trajectory has also been replayed with the sensor mounted at the sensor position and the laser position of the dummy welding head. Figure 6.20(a) shows the sensor measurements during replay with the sensor tool are mainly within 0.05 mm. Figure 6.20(b) shows the sensor measurements during replay with the sensor mounted at the laser tool position of the dummy welding head. The measurements on the right are somewhat larger than the ones on the left (mainly within 0.1 mm). These results are very similar to the results that are achieved with teaching as almost no orientation corrections are needed for a line trajectory.



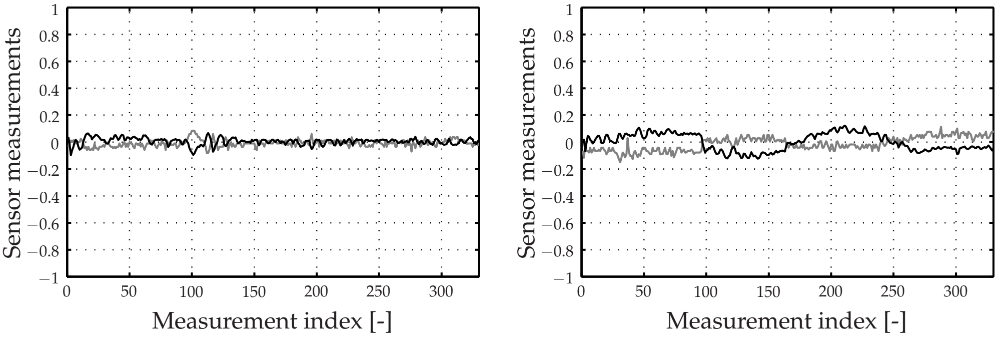
(a) Actual trajectory

(b) Sensor data during teaching

(c) Replay with Sensor tool

(d) Replay with Laser tool

Figure 6.19: Experimantal results of the real-time teaching algorithm on the line trajectory (— s_z [mm], — s_y [mm])



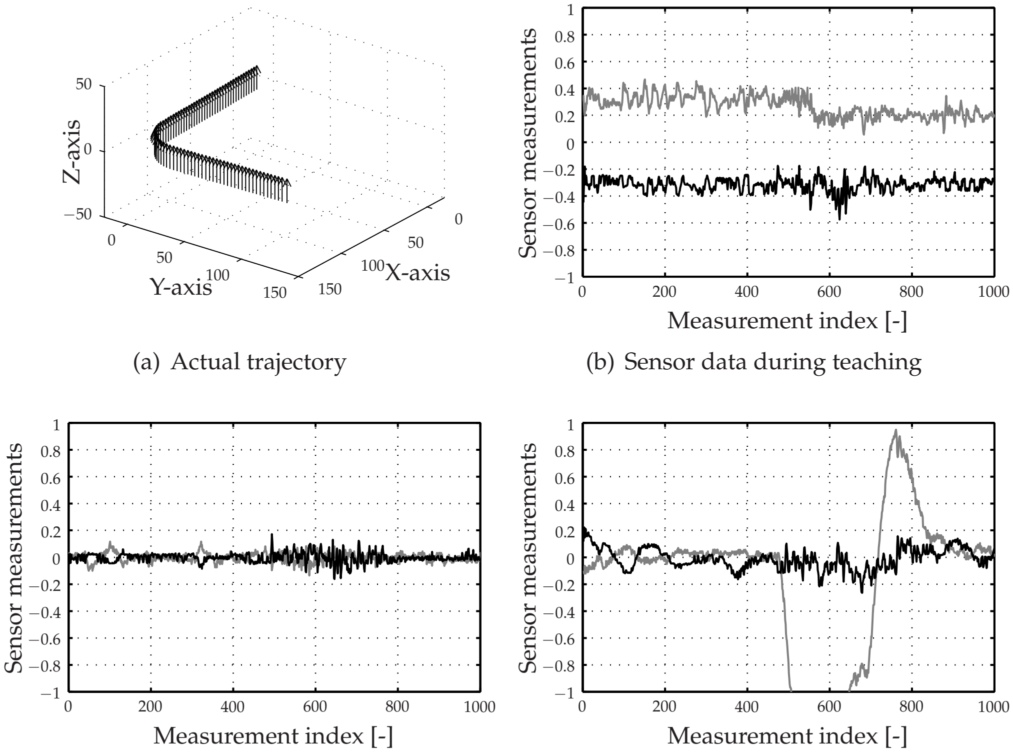
(a) Replay with Sensor tool

(b) Replay with Laser tool

Figure 6.20: Experimental results of the real-time tracking algorithm on the line trajectory (— s_z [mm], — s_y [mm])

Corner trajectory

To get a nominal trajectory for the corner trajectory, the polyfit point-to-point teaching procedure that is described in chapter 5 is used. After the point-to-point teaching has completed, a nominal trajectory is available. The actual path locations of the corner trajectory and the sensor measurements during the teaching phase are shown in figures 6.21(a) and 6.21(b). These sensor measurements are more irregular than the ones that are obtained with the line trajectory, which is caused by sensor measurement noise. The line trajectory is an overlap weld, which is easily detectable by the sensor, and the corner trajectory is a butt weld, which is more sensitive to optical reflections.



(a) Actual trajectory

(b) Sensor data during teaching

(c) Replay with Sensor tool

(d) Replay with Laser tool

Figure 6.21: Experimental results of the real-time teaching algorithm on the corner trajectory (— s_z [mm], — s_y [mm])

Teaching experiment

Figure 6.21(c) shows the sensor measurements during replay with the sensor tool. These measurements are mainly within 0.1 mm, which shows that even heavily curved trajectories can be accurately welded provided the corrections are applied with the same tool that is used for measuring. Figure 6.21(d) shows the sensor measurements during replay with the sensor mounted at the laser tool position of the dummy welding head. The measurements on the right are much larger in the corner (more than 1 mm) than the ones on the left. The large curvature in the corner causes geometric errors if the replayed laser tool differs from the sensor tool that is used for measuring.

Curved sine trajectory

To get a nominal trajectory for the curved sine trajectory, the polyfit point-to-point teaching procedure that is described in chapter 5 is used again. The actual path locations of the curved sine trajectory and the sensor measurements during the teaching phase are shown in figures 6.22(a) and 6.22(b). Even though this trajectory is an overlap weld as well, the sensor measurements are much more irregular than the ones that are obtained with the line trajectory. It is expected that this is caused by vibrations of the robot or the mechanical interface, caused by moving along a curved trajectory.

Teaching experiment

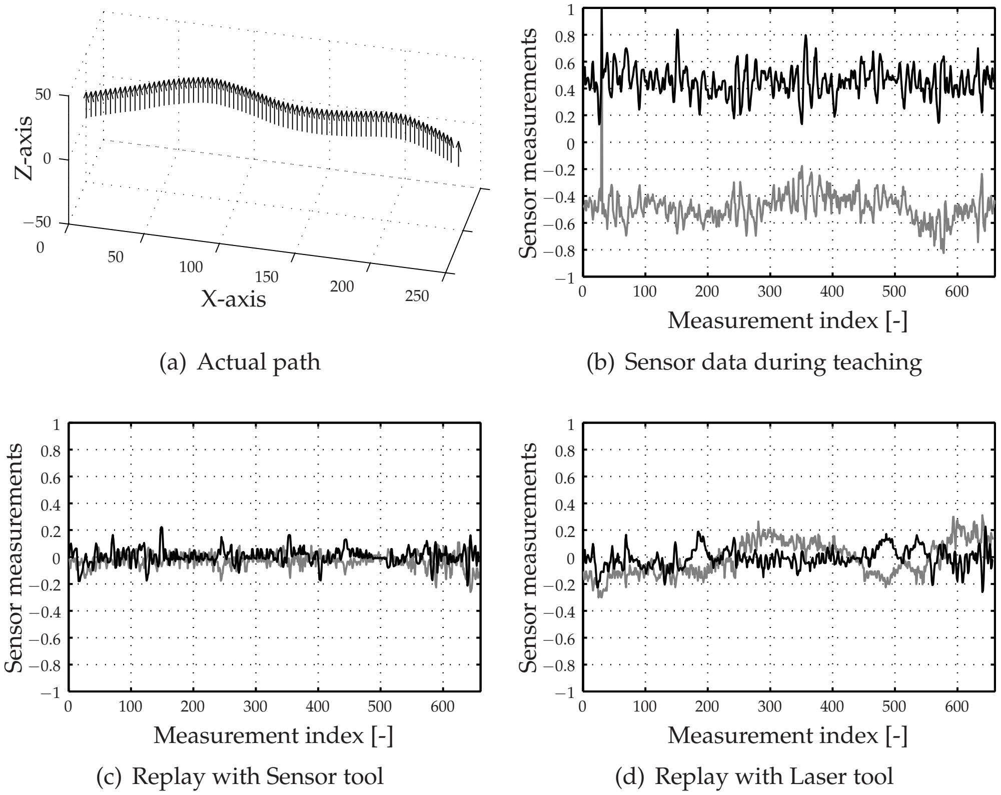
Figure 6.22(c) shows the sensor measurements during replay with the sensor tool. These measurements are mainly within 0.2 mm, which is appropriate for laser welding at the sensor tool location. Figure 6.22(d) shows the sensor measurements during replay with the sensor mounted at the laser tool position of the dummy welding head. The inaccuracy increases again in this case, some errors are about 0.3 mm, which is more than the tight requirement mentioned in the introduction. As a result, some welds may fail, but this accuracy is still quite useful for laser welding of a wide range of industrial products, e.g. using larger focal distance.

Tracking experiment

The sensor measurements during replay of the tracked seam trajectory are shown in figure 6.23. The errors during replay with the sensor tool in figure 6.23(a) are mainly within 0.4 mm, the errors during replay with the laser tool in figure 6.23(b) are somewhat larger (mainly within 0.5 mm). In both cases, the accuracy is considerably worse than the requirement of 0.2 mm, which is a geometric robot effect caused by the fact that the robot orientation during the sensor measurements is different than the robot orientation when the laser tool reaches the measured location.

6.7 Discussion

In this chapter, a trajectory-based control architecture for real-time seam tracking has been presented. This trajectory-based control architecture has several advantages over the time-based methods that are frequently used in industry. To



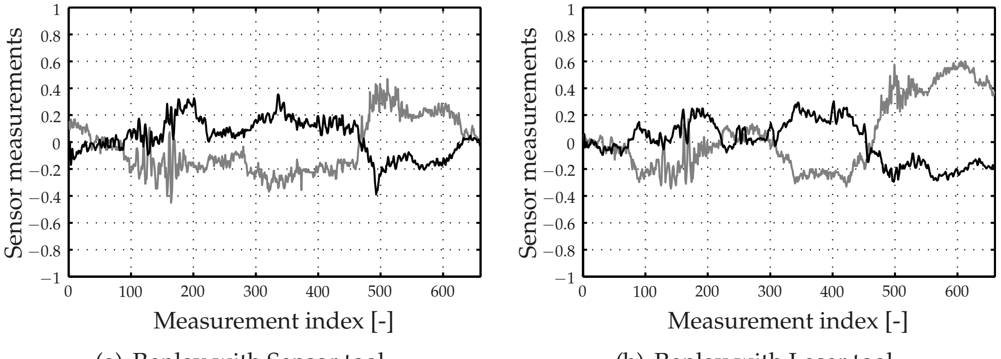
(a) Actual path

(b) Sensor data during teaching

(c) Replay with Sensor tool

(d) Replay with Laser tool

Figure 6.22: Experimental results of the real-time teaching algorithm on the curved sine trajectory (— s_z [mm], — s_y [mm])



(a) Replay with Sensor tool

(b) Replay with Laser tool

Figure 6.23: Experimental results of the real-time tracking algorithm on the curved sine trajectory (— s_z [mm], — s_y [mm])

be able to use it a real-time Setpoint Generator and a method to synchronise the robot joint measurements and the sensor image acquisition are needed. A Cartesian Setpoint Generator based on cubic interpolation was implemented (section 6.4).

A synchronisation method has been presented (section 6.5) for a seam tracking sensor that is integrated with an industrial robot controller. It uses Ethernet UDP-communication, which makes it fast and cheap. Experiments have been carried out to determine the time-delay between the robot joint measurements and the sensor measurements, which is found to be 4.9 ms with our hardware. The jitter in the system is around 0.6 ms, which makes the synchronisation method very suitable for the application of laser welding.

A real-time seam tracking algorithm has been implemented using the trajectory-based control approach. The orientation of the synchronised locations is adapted to keep the sensor within the field-of-view of the nominal seam trajectory when the laser reaches the synchronised location. The control loop is closed by filtering the corrected locations and adding the filtered locations to the Tool Trajectory Buffer. It shows that the trajectory-based control approach is well suited for real-time seam tracking.

Real-time teaching and real-time tracking experiments have been performed using a line trajectory, a corner trajectory and a curved sine trajectory. From these experiments the following conclusions can be drawn:

- The real-time seam tracking procedure is well suited for welding of moderately curved seam trajectories. The procedure is fast, because measuring and welding are done simultaneously in the same processing step. The procedure requires a sensor that measures ahead of the laser focal point. The obtained accuracy depends on the orientation correction, which is a function of the curvature of the seam trajectory. They range from 0.1 mm for a line trajectory to 0.5 mm for the curved sine trajectory.

- The real-time teaching procedure is well suited for accurate welding of heavily curved seam trajectories, provided that the corrections are done with the same tool transformations as the measurements (small or zero look-ahead distance). The procedure requires two processing steps, first measuring, then welding. Accuracies better than 0.1 mm are achieved, which are almost independent of the curvature of the seam trajectory.

It is expected that a considerable improvement of the accuracies can be achieved by using more accurate geometric robot models. The actual implementation of such models is out of the scope of this thesis.

Chapter 7

Conclusions and discussion

7.1 Conclusions

This section summarises the conclusions from the previous chapters. The conclusions are ordered by the different topics that are described in three different chapters in this thesis.

7.1.1 Tool calibration procedures

From the tool calibration experiments and simulations (chapter 4), the following conclusions are drawn:

- An automatic laser tool calibration procedure has been developed and implemented that computes the transformation ${}^{N}_{L}\mathbf{T}$ between the robot flange and the laser tool frame using a coaxial camera attached to the laser welding head and a calibration object. Experiments have been performed, which show that the resulting accuracy of the calibration procedure is mainly determined by robot geometric errors and that because of these errors, the positional accuracy of transformation ${}^{N}_{L}\mathbf{T}$ is limited to about 1 mm.

- An automatic sensor tool calibration procedure has been developed and implemented that computes the transformation ${}^{N}_{S}\mathbf{T}$ between the robot flange and the sensor tool frame. Both experiments and simulations show that the resulting accuracy of the calibration procedure is mainly determined by robot geometric errors as well and that therefore the accuracy of transformation ${}^{N}_{S}\mathbf{T}$ is limited to about 1 mm.

- A combined tool calibration procedure has been developed and implemented that computes the transformation ${}^{L}_{S}\mathbf{T}$ between the laser and sensor tool frame without moving the robot. This method is more accurate ($<$0.1 mm) than the two previous methods, because it is a direct measurement that is not influenced by robot geometric errors.

Using the result of the first two procedures, it is obviously not possible to compute the transformation ${}^{L}_{S}\mathbf{T}$ within the accuracy requirements of 0.2 mm. Therefore the third procedure is needed to accurately determine the transformation ${}^{L}_{S}\mathbf{T}$. In the beginning of chapter 4, it was shown that it is not problematic that the transformations ${}^{N}_{L}\mathbf{T}$ and ${}^{N}_{S}\mathbf{T}$ have positional errors bigger than 0.2 mm as long as the transformation ${}^{L}_{S}\mathbf{T}$ is accurate, which is achieved with the third procedure.

7.1.2 Seam teaching using point-to-point movements

In chapter 5 the subject of seam-teaching using point-to-point movements is discussed. An adapted tractrix and a polyfit seam teaching algorithm have been described that can both be used for the automatic teaching of the geometry of a seam trajectory. The adapted tractrix algorithm is simple and straightforward, but in practice the polyfit teaching algorithm will be used, because it gives the operator much more freedom on setting the filtering parameters.

A simulation environment has been developed, which has been used to design and test the teaching algorithms and to examine the influence of various errors that occur in a sensor-guided robotic laser welding environment. Simulations and experiments have been performed, which show that the teaching algorithms work appropriately. From these point-to-point teaching experiments and simulations the following conclusions are drawn:

- A position calibration error between the Laser and Sensor tool frames has a direct effect on the accuracy as it results in an offset during welding. A position error between the tools in y-direction results in an offset perpendicular to the seam trajectory. A position error in x-direction has an effect on the start and stop locations of the weld and has a larger effect at locations where the seam trajectory has a smaller radius of curvature. An error in z-direction results in an offset of the height of the laser focal point. It is therefore important to accurately (better than 0.2 mm) calibrate the transformation between Laser and Sensor tool frames.

- Measurement errors in the sensor orientation result in a fluctuating robot motion during teaching. In simulation, this effect is much smaller than observed experimentally. Non-linearities in the robot kinematics may play a role. For a further investigation, the simulation environment should be extended to include these non-linearities.

- A different robot position between teaching and re-teaching caused by errors in the robot geometric model has a noticeable effect on the accuracy of the laser focal position. Errors up to 0.2 mm occurred for the identified geometric robot models of De Roo (2003) and a look-ahead distance of 55 mm.

The resulting accuracy of the laser focal point relative to the seam is important. Using sensors, this accuracy is improved, i.e. it has become better than the absolute accuracy of the robot arm. The resulting accuracy however, is still not approaching the repeatability of the robot arm. This work shows that this is mainly determined by errors in the geometric robot model.

To decrease the effect of errors in the geometric robot model, large robot joint deviations between teaching and welding should be avoided. This can be realised by choosing tool definitions that are close to the robot flange or by minimising the transformation between laser and sensor tool frame. The physical dimensions of the welding head and sensor usually determine the minimal look-ahead distance that can be chosen. Several designs of welding heads are currently available or in development with integrated sensors to achieve close or overlapping sensor and laser tool frames, e.g. Iakovou et al. (2005); Falldorf Sensor (2006). Another possibility to minimise joint movements is by preventing the orientation of the tool definition to change too much as even small orientation changes of the tool can cause a large movement of the robot joints.

The accuracy requirements for laser welding make sensor-guided robotic laser welding more difficult, compared to conventional robotic welding applications like e.g. arc welding. The influence of all kind of errors that occur in a sensor-guided robotic laser welding system should be known when using a sensor for automatic teaching. The simulation environment that is developed in this work is valuable for this purpose. Furthermore it is very convenient for testing and development of seam teaching algorithms.

7.1.3 Real-time seam tracking

In chapter 6, a trajectory-based control architecture for real-time seam tracking has been presented. This trajectory-based control architecture has several advantages over the time-based methods that are frequently used in industry. To be able to use it a real-time Setpoint Generator and a method to synchronise the robot joint measurements and the sensor image acquisition are needed. A Cartesian Setpoint Generator based on cubic interpolation was implemented (section 6.4). A synchronisation method has been presented (section 6.5) for a seam tracking sensor that is integrated with an industrial robot controller. It uses Ethernet UDP-communication, which makes it fast and cheap. Experiments have been carried out to determine the time-delay between the robot joint measurements and the sensor measurements, which is found to be 4.9 ms in our particular case. The jitter in the system is about 0.6 ms, which makes the synchronisation procedure very suitable for the application of laser welding.

A real-time seam tracking algorithm has been implemented using the trajectory-based control approach. The orientation of the synchronised locations is corrected to keep the sensor within the field-of-view of the nominal seam trajectory when the laser reaches the synchronised location. The control loop is closed by filtering the corrected locations and adding the filtered locations to the Tool Trajectory Buffer.

Teaching and real-time tracking experiments have been performed using a line trajectory, a corner trajectory and a curved sine trajectory. From these experiments the following conclusions can be drawn:

- The real-time seam tracking procedure is well suited for welding of moderately curved seam trajectories. The procedure is fast, because measuring and welding are done in the same processing step. The procedure requires a sensor that measures ahead of the laser focal point. The obtained accuracy depends on the orientation correction, which is a function of the curvature of the seam trajectory. They differ between 0.1 mm for a line trajectory and 0.5 mm for the curved sine trajectory.
- The real-time teaching procedure is well suited for accurate welding of heavily curved seam trajectories, provided that the corrections are done with the same tool transformations as the measurements (small or zero look-ahead distance). The procedure requires two processing steps, first measuring, then welding. Accuracies better than 0.1 mm are achieved, which are almost independent of the curvature of the seam trajectory.

7.2 Discussion and suggestions for further research

It is expected that a considerable improvement of the accuracy of the tool calibration procedures, the seam teaching procedures and the real-time seam tracking procedures can be achieved when more accurate geometric robot models are available. However, improving the accuracy of these robot models is not straightforward for the following reasons:

1. The geometric model structure has to be extended with all parameters that play a significant role on the tip accuracy and the correct values of these parameters has to be determined. Common parameters that are already used in geometric robot models are e.g. arm lengths, encoder offsets and link angles. An example of a more complex parameter that must be included is elasticity due to tip mass, which is a function of the arm position (horizontal or upright).
2. The geometric model parameters may change from time to time (due to changes in temperature, tip mass, etc). Therefore in practice, an automated procedure is needed to calibrate the geometric robot model on the work floor and to check if the geometric robot model is still calibrated.

3. Symbolic expressions for calculating the Inverse Geometric Model can usually not be derived for more complex geometric robot models. Instead, iterative solutions need to be computed, which are more computer intensive and can therefore not be used without effort within the real-time control loop of the robot controller.

One approach to overcome the latter difficulty would be to compute the actual Inverse Geometric Model with a first order approximation, when an actual Direct Geometric Model is available. At a certain Cartesian location, the nominal Inverse Geometric Model is used to compute the nominal value of the robot joints. Using these nominal joint values with the actual Direct Geometric Model, a Cartesian deviation can be computed, which is used to compute the actual robot joint values. This approach would require the computation of twice the nominal IGM and once the actual DGM within the real-time control loop of the robot controller instead of originally only once the nominal IGM. This gives some increase in computation time, but the total computation time is fixed and much smaller than the time that would be needed for an iterative solution of the actual IGM.

A number of promising techniques are recognised to improve the accuracy and usability of robotic laser welding:

- Development of an integrated and compact welding head. (Iakovou et al., 2005). The previous section showed that a welding head with integrated sensors that measure close to or at the laser focal point gives a great increase in the welding accuracy. Besides this advantage, the welding head that is developed in this research measures with a triangle around the laser focal point and does not require a specific orientation of the sensor relative to the seam trajectory. Therefore, it can be used to measure and weld specially shaped trajectories like rectangles, circles, spirals, etc.

- Iterative Learning Control (Hakvoort et al., 2006). ILC is a control strategy used to increase the tracking accuracy of the robot joint motion controller, by repeating the trajectory and learning from errors made in previous runs. For laser welding this has the advantage that the tracking accuracy of the laser spot is considerably increased (better than 0.05 mm), even during welding of complex trajectories, e.g. sharp corners. The accuracy increase comes at the expense of a decrease in cycle time, as the welding trajectory needs to be taught at least once before it can be welded. Furthermore, a welding head with an integrated sensor that measures close to or at the laser focal point is needed.

- Improved laser beam manipulation (Hardeman et al., 2006). In this project, laser welding scanning heads are developed, where the laser spot can be moved in one or more directions by means of a moving mirror. These scanning heads contain integrated seam tracking sensors and a motorised mirror. Using an integrated feedback controller, the laser spot is kept on the seam trajectory.

- Off-line programming for laser welding (Waiboer, 2007). Off-line programming software can be used to generate welding trajectories for complex 3D products. In the software, 3D models of the welding cell, robot, external manipulator, welding head, work table, product and clamping materials can be used to create a collision-free welding trajectory. A disadvantage of off-line programming software is the deviation between the product position in the software and in the real work cell. Off-line programming software is very useful to generate complex welding trajectories, without spending expensive time on the real welding equipment. Next, the work from this thesis can be applied using sensors to teach the actual seam with the accuracy that is needed for laser welding.
- Synchronous external axes support. By positioning the product on an external manipulator that moves synchronously with the robot arm, fast robot arm movements can be decreased considerably for complex products by using a combined motion of the robot and product on the manipulator, thus decreasing the dynamic tracking errors. Furthermore, the external manipulator can decrease gravity effects during welding by keeping the melt pool horizontally. The combination of external axes and sensors for seam teaching and seam tracking is a challenge.

7.3 Contributions

This thesis concludes with a summary of the contributions of this work:

- Automatic Laser tool calibration method for determining the transformation between the robot flange and the laser tool frame.
- Automatic Sensor tool calibration method for determining the transformation between the robot flange and the sensor tool frame.
- Combined Laser-Sensor tool calibration method for determining the transformation between the sensor and laser tool frame.
- Polyfit Seam-teaching algorithm for automatic teaching of the seam trajectory.
- Synchronisation procedure for robot-sensor synchronisation.
- Trajectory-based control architecture for real-time seam tracking

- Real-time Setpoint Generator that smoothly computes the robot trajectory.
- Real-time seam tracking algorithm for correcting small errors from a predefined seam trajectory.
- Modular software environment based on Ethernet socket communication.
- 24LASER Graphical User Interface to define and execute robotic laser welding jobs.
- Simulation environment with geometric models of the seam trajectory, seam tracking sensor, robot, robot controller and the teaching algorithms.

Appendix A

Software

A.1 Robot software

A.1.1 RobotLib

The RobotLib library is written in C++ and contains the tools and mathematics needed to work with robots. It depends on the freely available Newmat library (Davies, 2002) for basic matrix and vector manipulation (add, subtract, multiply, etc). The RobotLib library consists of a number of C++ classes:

- Mdesc. The motion descriptor contains the velocity, acceleration and deceleration bounds for a movement in Joint space or Cartesian space.
- Quaternion. To describe orientation, quaternions or Euler parameters are used. This class contains the quaternion mathematics and the conversions from and to different representations of orientation (Euler angles, rotation matrix, etc).
- Transform. A transformation contains a 3D vector for the position and a quaternion for the orientation.
- Tool. A tool is a special type of transform, namely a transform which is attached to the robot flange or Null frame **N**.
- Frame. A frame is a special type of transform, namely a transform which is attached to the robot base frame **B**.
- Location. A location is a special type of a transform used to describe the location of a 3D point in space, e.g. a seam location.
- Robot. This class contains the kinematic parameters for the Stäubli RX90 and RX130 robot and symbolic functions for calculating the forward kinematics, inverse kinematics and tool frame Jacobian matrix. It also contains

the nominal and maximum velocity, acceleration and deceleration bounds for the robot joints.

- RobotConfig. A 6-axis revolute joint robot has a maximum of eight ways to reach a 3D location in space. The robot configuration describes the configuration of the robot shoulder (Lefty, Righty), elbow (Above, Below) and wrist (Positive, Negative).
- Trajectory. A robot trajectory consists of a number of points, which the robot has to track from start to end. A trajectory can be defined in Joint space, Cartesian space or as a list of Joint setpoints at a specific time cycle (4 ms in our case). At every point a different motion descriptor and equipment can be specified.
- TrajGenerator. The Stäubli RTRCAL requires new joint angle and joint velocity setpoints to be specified every 4 ms. The trajectory generator calculates these setpoints in real-time every 4 ms from the robot trajectory. It is described in detail in section 6.4.

A.1.2 RobotSocket

The RobotSocket layer is programmed as a C++ class and uses Ethernet socket communication. On the side of the robot controller, a RobotSocket server waits for RobotSocket clients that connect to it.

The RobotSocket server listens on a specific port (default 4000) for incoming TCP request packets. Request packets need to have a special structure in which e.g. a Command ID is included to specify the requested robot command. If the request packet is identified as being a correct request packet, the RobotSocket server carries out the request based on the Command ID. If the request was carried out correctly, the RobotSocket server responds with a reply packet. If an error occurred during processing of the request the RobotSocket server responds with an error packet.

A list of important client commands is given next:

- joints = Here()
- EnablePower()
- DisablePower()
- JointMovej(joints)
- ToolMovej(loc, tool, frame)
- ToolMovel(loc, tool, frame)
- etc.

A.2 Sensor software

A.2.1 Communicator

The Falldorf seam tracking sensor can extract a number of features from a single image. The most important ones that are used in this work are:

- X-y-z position of left seam edge
- X-y-z position of right seam edge
- Orientation angle of the seam
- Width of the seam
- Sheet height mismatch between left and right side
- Onsheet bit, which becomes greater than zero if there is a seam within the sensors field-of-view
- Column-wise profile of the camera-image
- Column-wise intensity at the camera-image

The Falldorf Inspector software that is used for image processing stores these features in a part of shared memory and allows users to write a user application to read the features and use them for their purpose. A piece of software called Communicator is written for the following tasks:

- Display current sensor data, sensor profile and intensity profile.
- Communicate sensor features through the Ethernet to clients using the SensorSocket layer (section 2.4.5).
- Choose between real sensor data, user-supplied data or simulated data generated using models of the seam tracking sensor and seam trajectories. The modelling of sensor and seam trajectory is described in section 3.5.

A.2.2 SensorSocket

The SensorSocket software layer is written in C++ and is based on Ethernet socket communication. On the sensor computer, a SensorSocket server waits for clients that connect to it. 24LASER uses a SensorSocket client to connect and send commands to the sensor computer.

The SensorSocket server listens on a specific port (default 2000) for incoming TCP request packets. Request packets need to have a special structure. If the request packet is identified as being a correct request packet, the SensorSocket server carries out the request based on the Command ID within the request packet. If the request was carried out correctly, the SensorSocket server responds with a reply packet. If an error occurred during processing of the request the SensorSocket server responds with an error packet.

Appendix B

Transformations

B.1 Homogenous transformation matrix

A transformation describes the location (position and orientation) of a coordinate frame with respect to a reference frame. Transformations are indicated by the symbol **T** and may have a leading superscript, that defines the reference frame they refer to. The leading subscript defines the frame they describe, e.g. transformation ${}^{A}_{B}\mathbf{T}$ describes frame **B** with respect to frame **A**.

In literature, a 4x4 homogenous transformation matrix is often used to describe a transformation. A transformation ${}^{A}_{B}\mathbf{T}$ can be written as

$$ {}^{A}_{B}\mathbf{T} = \begin{bmatrix} {}^{A}_{B}\mathbf{R} & {}^{A}_{B}\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{B.1} $$

where ${}^{A}_{B}\mathbf{R}$ is a 3x3 rotation matrix that describes the orientation of frame **B** with respect to frame **A** and ${}^{A}_{B}\mathbf{P}$ is a 3D position vector that describes the position of the origin of frame **B** with respect to frame **A**. In the remainder of this work the leading subscript and superscript of a transformation (or its components) may be omitted to make the equations more clear. In cases where it is necessary to differentiate between different frames the subscript and superscript will be used.

The position vector **P** contains the elements P_x, P_y and P_z, in x, y and z-direction respectively, as

$$ \mathbf{P} = [P_x \quad P_y \quad P_z]^T. \tag{B.2} $$

The rotation matrix **R** consists of three orthonormal vectors $\mathbf{r_x} = [r_{xx} \quad r_{xy} \quad r_{xz}]^T$, $\mathbf{r_y} = [r_{yx} \quad r_{yy} \quad r_{yz}]^T$ and $\mathbf{r_z} = [r_{zx} \quad r_{zy} \quad r_{zz}]^T$, that describe the unit vectors along the three coordinate axes. It can be written as

$$\mathbf{R} = [\mathbf{r_x} \quad \mathbf{r_y} \quad \mathbf{r_z}] = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix}. \tag{B.3}$$

The homogenous transformation matrix can now be written in element-form as

$$\mathbf{T} = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} & P_x \\ r_{xy} & r_{yy} & r_{zy} & P_y \\ r_{xz} & r_{yz} & r_{zz} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.4}$$

B.2 Properties of homogenous transformation matrices

Homogenous transformation matrices have some useful properties, that are used throughout this work. These properties will be described in this section. Their derivations can be found in many textbooks on robotics, e.g. (Khalil and Dombre, 2002) or (Craig, 1986).

B.2.1 Orthonormality

Since the rotation matrix **R** contains three orthonormal vectors, the length of each vector should be

$$|\mathbf{r_x}| = |\mathbf{r_y}| = |\mathbf{r_z}| = 1, \tag{B.5}$$

and each vector can be deduced from the cross product of the other two, so

$$\mathbf{r_x} = \mathbf{r_y} \times \mathbf{r_z} \tag{B.6}$$

$$\mathbf{r_y} = \mathbf{r_z} \times \mathbf{r_x} \tag{B.7}$$

$$\mathbf{r_z} = \mathbf{r_x} \times \mathbf{r_y}. \tag{B.8}$$

B.2.2 Inverse

The rotation matrix **R** is orthogonal, which means that its inverse is equal to its transpose

$$\mathbf{R}^{-1} = \mathbf{R}^T. \tag{B.9}$$

The inverse of a transformation ${}^A_B\mathbf{T}$ represents the location of frame **A** with respect to frame **B**. It can be obtained as

$$ {}^A_B\mathbf{T}^{-1} = {}^B_A\mathbf{T} = \begin{bmatrix} {}^A_B\mathbf{R}^T & -{}^A_B\mathbf{R}^T \cdot {}^A_B\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{B.10}$$

B.2.3 Multiplication

A position vector ${}^B\mathbf{P}$ that is known with respect to frame **B** can be transformed to a different frame **A** using the transformation ${}^A_B\mathbf{T}$ as

$$\begin{bmatrix} {}^A\mathbf{P} \\ 1 \end{bmatrix} = {}^A_B\mathbf{T} \begin{bmatrix} {}^B\mathbf{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_B\mathbf{R}{}^B\mathbf{P} + {}^A_B\mathbf{P} \\ 1 \end{bmatrix}. \tag{B.11}$$

The multiplication of two transformation matrices ${}^A_B\mathbf{T}$ and ${}^B_C\mathbf{T}$ gives

$$ {}^A_C\mathbf{T} = {}^A_B\mathbf{T} \cdot {}^B_C\mathbf{T} = \begin{bmatrix} {}^A_B\mathbf{R} & {}^A_B\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^B_C\mathbf{R} & {}^B_C\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^A_B\mathbf{R} \cdot {}^B_C\mathbf{R} & {}^A_B\mathbf{R} \cdot {}^B_C\mathbf{P} + {}^A_B\mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{B.12}$$

Note that the matrix multiplication is non-commutative (${}^A_B\mathbf{T} \cdot {}^B_C\mathbf{T} \neq {}^B_C\mathbf{T} \cdot {}^A_B\mathbf{T}$).

B.2.4 Pure translation

Let **Trans**(a,b,c) be the transformation matrix of a pure translation, where a,b and c denote the translation along the x,y and z-axes respectively. This transformation is expressed as

$$\mathbf{Trans}(a,b,c) = \begin{bmatrix} & & & a \\ & \mathbf{I}(3) & & b \\ & & & c \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.13}$$

where $\mathbf{I}(3)$ is a 3x3 identity matrix.

B.2.5 Pure rotation

Let **rot**(**u**,θ) be the rotation matrix and **Rot**(**u**,θ) be the transformation representing a rotation of an angle θ about an axis, with unit vector $\mathbf{u} = [u_x \quad u_y \quad u_z]^T$, located at the origin of the frame. They are related as

$$\mathbf{Rot}(\mathbf{u},\theta) = \begin{bmatrix} \mathbf{rot}(\mathbf{u},\theta) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{B.14}$$

The rotation matrix $\mathbf{rot}(\mathbf{u}, \theta)$ is expressed as

$$\mathbf{rot}(\mathbf{u}, \theta) = \begin{bmatrix} u_x^2(1{-}C\theta) + C\theta & u_x u_y(1{-}C\theta) - u_z S\theta & u_x u_z(1{-}C\theta) + u_y S\theta \\ u_x u_y(1{-}C\theta) + u_z S\theta & u_y^2(1{-}C\theta) + C\theta & u_y u_z(1{-}C\theta) - u_x S\theta \\ u_x u_z(1{-}C\theta) - u_y S\theta & u_y u_z(1{-}C\theta) + u_x S\theta & u_z^2(1{-}C\theta) + C\theta \end{bmatrix}, \tag{B.15}$$

where $C\theta = \cos(\theta)$ and $S\theta = \sin(\theta)$. The transformations that correspond to a pure rotation with an angle θ around the principle axes can easily be derived from equations B.14 and B.15. The transformation **Rot**(**x**,θ) is the rotation around the x-axis, by using $\mathbf{u} = [1 \;\; 0 \;\; 0]^T$. Transformation **Rot**(**y**,θ) is the rotation around the y-axis, by using $\mathbf{u} = [0 \;\; 1 \;\; 0]^T$ and **Rot**(**z**,θ) is the rotation around the z-axis, by using $\mathbf{u} = [0 \;\; 0 \;\; 1]^T$.

B.3 Representations of orientation

Different representations exist to describe the orientation between two coordinate frames, like yaw-pitch-roll, Euler angles, axis-angle, Euler parameters or quaternions and direction cosines (Khalil and Dombre, 2002). Khalil and Dombre show how these forms can be computed from one to another. In this work the quaternion and Euler angles representation are frequently used. Therefore the conversions between the rotation matrices and these representations will be given.

B.3.1 Quaternions

The quaternions are also called Euler parameters or Olinde-Rodrigues parameters. A quaternion **Q** has a scalar part q_s and a 3D vector part $\mathbf{q_v} = [q_{v_1} \;\; q_{v_2} \;\; q_{v_3}]^T$. For a rotation with angle θ around a unit 3D vector **u**, the unit quaternion **Q** can be written as

$$\mathbf{Q} = \begin{bmatrix} q_s \\ \mathbf{q_v} \end{bmatrix} = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ \mathbf{u} \cdot \sin(\frac{\theta}{2}) \end{bmatrix}. \tag{B.16}$$

A rotation matrix **R** can be computed from a quaternion **Q** as

$$\mathbf{R} = \begin{bmatrix} 2(q_s^2 + q_{v_1}^2) - 1 & 2(q_{v_1}q_{v_2} - q_s q_{v_3}) & 2(q_{v_1}q_{v_3} + q_s q_{v_2}) \\ 2(q_{v_1}q_{v_2} + q_s q_{v_3}) & 2(q_s^2 + q_{v_2}^2) - 1 & 2(q_{v_2}q_{v_3} - q_s q_{v_1}) \\ 2(q_{v_1}q_{v_3} - q_s q_{v_2}) & 2(q_{v_2}q_{v_3} + q_s q_{v_1}) & 2(q_s^2 + q_{v_3}^2) - 1 \end{bmatrix}. \tag{B.17}$$

The inverse problem is computing the quaternion components for a given rotation matrix **R**. They are given as

$$q_s = \frac{1}{2}\sqrt{r_{xx} + r_{yy} + r_{zz} + 1} \tag{B.18}$$
$$q_{v_1} = \frac{1}{2}\mathrm{sign}(r_{yz} - r_{zy})\sqrt{r_{xx} - r_{yy} - r_{zz} + 1} \tag{B.19}$$
$$q_{v_2} = \frac{1}{2}\mathrm{sign}(r_{zx} - r_{xy})\sqrt{-r_{xx} + r_{yy} - r_{zz} + 1} \tag{B.20}$$
$$q_{v_3} = \frac{1}{2}\mathrm{sign}(r_{xy} - r_{yx})\sqrt{-r_{xx} - r_{yy} + r_{zz} + 1}, \tag{B.21}$$

where sign() is the sign of the value between parenthesis. The quaternion product of two quaternions $\mathbf{Q}_1$ and $\mathbf{Q}_2$ is defined as

$$\mathbf{Q}_1\mathbf{Q}_2 = \begin{bmatrix} q_{s,1}q_{s,2} - \mathbf{q}_{\mathbf{v},\mathbf{1}} \cdot \mathbf{q}_{\mathbf{v},\mathbf{2}} \\ q_{s,1}\mathbf{q}_{\mathbf{v},\mathbf{2}} + q_{s,2}\mathbf{q}_{\mathbf{v},\mathbf{1}} + \mathbf{q}_{\mathbf{v},\mathbf{1}} \times \mathbf{q}_{\mathbf{v},\mathbf{2}} \end{bmatrix}. \tag{B.22}$$

Multiplication is not commutative, i.e. the products $\mathbf{Q}_1\mathbf{Q}_2$ and $\mathbf{Q}_2\mathbf{Q}_1$ are not necessarily equal. The conjugate $\overline{\mathbf{Q}}$ of quaternion $\mathbf{Q}$ is defined as

$$\overline{\mathbf{Q}} = \begin{bmatrix} q_s \\ -\mathbf{q}_\mathbf{v} \end{bmatrix}. \tag{B.23}$$

In the case that $\mathbf{Q}$ is a unit quaternion, the inverse $\mathbf{Q}^{-1}$ is equal to the conjugate.

B.3.2 Euler angles

The Euler angles specify three successive rotations around a principal coordinate axis. There are twelve meaningful choices for the order of rotations, the specific choice should always be explicitly mentioned. In this work the X-Y-Z Euler angles form is used. A rotation matrix **R** is described by successively rotating around the X-axis, Y-axis and Z-axis as

$$\begin{aligned} \mathbf{R} &= \mathbf{rot}(\mathbf{x}, \gamma) \cdot \mathbf{rot}(\mathbf{y}, \beta) \cdot \mathbf{rot}(\mathbf{z}, \varphi) \\ &= \begin{bmatrix} C\gamma C\varphi - S\gamma C\beta S\varphi & -C\gamma S\varphi - S\gamma C\beta C\varphi & S\gamma S\beta \\ S\gamma C\varphi + C\gamma C\beta S\varphi & -S\gamma S\varphi + C\gamma C\beta C\varphi & -C\gamma S\beta \\ S\gamma S\varphi & S\gamma C\varphi & C\gamma \end{bmatrix}, \end{aligned} \tag{B.24}$$

where $C\gamma = \cos(\gamma)$, $S\gamma = \sin(\gamma)$, $C\beta = \cos(\beta)$, $S\beta = \sin(\beta)$, $C\varphi = \cos(\varphi)$ and $S\varphi = \sin(\varphi)$.

The inverse problem is finding the Euler angles γ, β and φ for a given rotation matrix **R**. They are computed as

$$\gamma = \arctan 2(n_z, -a_z) \tag{B.25}$$

$$\beta = \arctan 2(s_z, a_z \cos(\gamma) - n_z \sin(\gamma)) \tag{B.26}$$

$$\varphi = \arctan 2(n_x \cos(\gamma) + a_x \sin(\gamma), n_y \cos(\gamma) + a_y \sin(\gamma)). \tag{B.27}$$

If two or three of the computed angles are larger than 90°, γ is decreased by 180° and equations B.26 and B.27 are computed again.

Appendix C

Geometric robot models

In this appendix, geometric models are derived for the Stäubli RX90 and RX130 robot arms. Symbolic expressions are given for the Direct Geometric Model and the Inverse Geometric Model.

C.1 Direct Geometric Model

The Direct Geometric Model (**DGM**) of a robot defines the transformation from the robot base (**B**) to the robot flange (**N**) as a function of its joint coordinates. In this work, the location of the flange is specified as a homogenous transformation matrix and transformation ${}^{B}_{N}\mathbf{T}$ can be expressed as

$$ {}^{B}_{N}\mathbf{T} = \mathbf{DGM}(\mathbf{q}), \tag{C.1} $$

where **q** is the vector of robot joint angles. Equation C.1 has a single solution, i.e. every vector of joint angles corresponds to one position and orientation of the robot flange or end-effector.

For serial six-axis robots, such as those in the Stäubli RX series, equation C.1 can be written as a successive multiplication of transformations:

$$ {}^{B}_{N}\mathbf{T} = {}^{B}_{1}\mathbf{T}(q_1) \cdot {}^{1}_{2}\mathbf{T}(q_2) \cdot {}^{2}_{3}\mathbf{T}(q_3) \cdot {}^{3}_{4}\mathbf{T}(q_4) \cdot {}^{4}_{5}\mathbf{T}(q_5) \cdot {}^{5}_{6}\mathbf{T}(q_6) \cdot {}^{6}_{N}\mathbf{T}, \tag{C.2} $$

where ${}^{B}_{1}\mathbf{T}(q_1)$ is the transformation from frame **B** to a coordinate frame that is fixed to robot joint 1, ${}^{i-1}_{i}\mathbf{T}(q_i)$ are the transformations from coordinate frames that are attached to joint $i-1$ and joint i and ${}^{6}_{N}\mathbf{T}$ is the transformation from a coordinate frame that is attached to robot joint 6 and the robot flange.

A graphical geometric description for robots of the Stäubli RX-series can be found in figure C.1. From this figure, symbolic expressions for the **DGM** can be derived

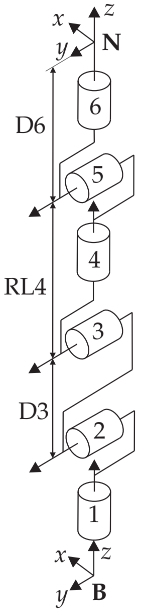


Figure C.1: Geometric description for robots of the Stäubli RX-series. By design, the joint axes 4,5 and 6 intersect in one point, allowing to derive symbolic expressions for the IGM.

Robot geometric parameters	RX90	RX130
D3 (mm)	450	625
RL4 (mm)	450	625
D6 (mm)	85	110

Table C.1: Geometric parameters of the Stäubli RX90 and RX130

(as described in Khalil and Dombre (2002)). The geometric parameters for the Stäubli RX90 and RX130 are given in table C.1.

In the remainder of this appendix, a number of sines and cosines are defined as

$$
\begin{aligned}
S1 &= \sin(q_1) & C1 &= \cos(q_1)\\
S2 &= \sin(q_2) & C2 &= \cos(q_2)\\
S3 &= \sin(q_2) & C3 &= \cos(q_2)\\
S23 &= \sin(q_2+q_3) & C23 &= \cos(q_2+q_3)\\
S4 &= \sin(q_4) & C4 &= \cos(q_4)\\
S5 &= \sin(q_5) & C5 &= \cos(q_5)\\
S6 &= \sin(q_6) & C6 &= \cos(q_6)
\end{aligned}
$$

The resulting optimal symbolic equations for the DGM are computed as follows:

$$
\begin{aligned}
T4612 &= C5 * S6\\
T4632 &= S5 * S6\\
T3612 &= C4 * T4612 + S4 * C6\\
T3613 &= C4 * S5\\
T3632 &= S4 * T4612 - C4 * C6\\
T3633 &= S4 * S5\\
T1314 &= D3 * C2\\
T1334 &= D3 * S2\\
T1612 &= S23 * T4632 - C23 * T3612\\
T1613 &= C23 * T3613 + S23 * C5\\
T1614 &= S23 * RL4 + D3 * S2\\
r_{yx} &= C1 * T1612 + S1 * T3632\\
r_{zx} &= C1 * T1613 - S1 * T3633\\
r_{yy} &= S1 * T1612 - C1 * T3632\\
r_{zy} &= S1 * T1613 + C1 * T3633\\
r_{yz} &= S23 * T3612 + C23 * T4632\\
r_{zz} &= C23 * C5 - S23 * T3613\\
r_{xx} &= r_{yy} * r_{zz} - r_{zy} * r_{yz}\\
r_{xy} &= r_{yz} * r_{zx} - r_{yx} * r_{zz}\\
r_{xz} &= r_{yx} * r_{zy} - r_{yy} * r_{zx}\\
P_x &= C1 * T1614 + r_{zx} * D6\\
P_y &= S1 * T1614 + r_{zy} * D6\\
P_z &= C23 * RL4 + D3 * C2 + r_{zz} * D6.
\end{aligned}
$$

The computations for the DGM consist of 6 sines, 6 cosines, 20 additions and 41 multiplications. With the robot controller (Celeron 566 Mhz) this can be computed in less than 0.1 ms.

C.2 Inverse Geometric Model

The Inverse Geometric Model (**IGM**) or inverse kinematics function of a robot or manipulator is the set of relations that gives the joint variables **q** corresponding to a specified location ${}^{B}_{N}\mathbf{T}$ of the end-effector. In this thesis, the Inverse Geometric Model for the robot arm is used frequently. It is the inverse of equation C.1 and can be formulated as

$$\mathbf{q} = \mathbf{IGM}({}^{B}_{N}\mathbf{T}). \tag{C.3}$$

In general, equation C.3 has multiple solutions and its complexity depends on the geometry of the robot. Equation C.3 has no solution if ${}^{B}_{N}\mathbf{T}$ is outside the robot workspace. Typically, for six degree-of-freedom robots with only revolute joints, where three joint axes intersect at a point, there are eight solutions for each configuration of the robot shoulder (above, below), elbow (lefty, righty) and wrist (positive, negative) (Khalil and Dombre, 2002). The symbolic expressions for equation C.3 for robots from the Stäubli RX series are derived in Khalil and Dombre (2002). The resulting symbolic expressions are given next.

First of all, the transformation ${}^{B}_{6}\mathbf{T}$ between the robot base frame and robot joint axis 6 is computed as

$${}^{B}_{6}\mathbf{T} ={}^{B}_{N} \mathbf{T} \cdot{}^{N}_{6} \mathbf{T}, \tag{C.4}$$

where transformation ${}^{N}_{6}\mathbf{T}$ can be written in Euler RxRyRz notation as [0 0 -D6 mm 0 0 0 deg], i.e. this transformation is a pure translation with a distance $D6$ from the robot flange to the intersection point of axis 4, 5 and 6 (figure C.1).

The position ${}^{B}_{6}\mathbf{P}$ of the intersection of joint 4, 5 and 6 is only a function of the joint variables, q_1, q_2 and q_3. This type of structure allows the decomposition of the six degree-of-freedom problem of equation C.2 into two three degree-of-freedom problems representing a position equation and an orientation equation. The position problem, which is a function of q_1, q_2 and q_3 is first solved, then the orientation problem allows to determine q_4, q_5 and q_6.

Position equation

The position equation from which q_1, q_2 and q_3 need to be solved is derived from equation C.2 as

$$\begin{bmatrix} {}^{B}_{6}P_x \\ {}^{B}_{6}P_y \\ {}^{B}_{6}P_z \\ 1 \end{bmatrix} ={}^{B}_{1} \mathbf{T}(q_1) \cdot{}^{1}_{2} \mathbf{T}(q_2) \cdot{}^{2}_{3} \mathbf{T}(q_3) \begin{bmatrix} {}^{3}_{4}P_x \\ {}^{3}_{4}P_y \\ {}^{3}_{4}P_z \\ 1 \end{bmatrix}. \tag{C.5}$$

For Stäubli RX robots (figure C.1), this is written as

$$\begin{bmatrix} {}^{B}_{6}P_x \\ {}^{B}_{6}P_y \\ {}^{B}_{6}P_z \\ 1 \end{bmatrix} = \begin{bmatrix} C1(S23 \cdot RL4 + D3 \cdot S2) \\ S1(S23 \cdot RL4 + D3 \cdot S2) \\ C23 \cdot RL4 + D3 \cdot C2 \\ 1 \end{bmatrix}, \tag{C.6}$$

which gives two solutions for q_1:

$$q_1 = \arctan2({}^{B}_{6}P_y, {}^{B}_{6}P_x) \tag{C.7}$$

$$q_1' = q_1 + \pi. \tag{C.8}$$

Using the solution for q_1, we can deduce that

$$\begin{aligned} B1 &= {}^{B}_{6}P_x \cdot C1 + {}^{B}_{6}P_y \cdot S1 \\ X &= -2 \cdot {}^{B}_{6}P_z \cdot D3 \\ Y &= -2 \cdot B1 \cdot D3 \\ Z &= (RL4)^2 - (D3)^2 - ({}^{B}_{6}P_y)^2 - (B1)^2 \end{aligned}$$

$$\begin{cases} C2 = \dfrac{YZ - \epsilon X\sqrt{X^2+Y^2-Z^2}}{X^2+Y^2} \\ S2 = \dfrac{XZ + \epsilon Y\sqrt{X^2+Y^2-Z^2}}{X^2+Y^2} \end{cases} \quad \text{with} \quad \epsilon = \pm 1.$$

If $X^2 + Y^2 - Z^2$ is smaller than 0 there exists no solution for the IGM in the robot workspace and the further computation of the IGM will be cancelled. Otherwise, two solution for q_2 are computed as

$$q_2 = \arctan2(S2, -C2). \tag{C.9}$$

and q_3 is computed as

$$q30 = \arctan2(S3, -C3), \tag{C.10}$$

with

$$\begin{cases} C3 = \dfrac{-B1 \cdot S2 + {}^{B}_{6}P_z \cdot C2}{RL4} \\ S3 = \dfrac{-{}^{B}_{6}P_x \cdot S2 - B1 \cdot C2 + D3}{RL4} \end{cases}.$$

Orientation equation

Since, q_1, q_2 and q_3 are known, the orientation part of equation C.2 is now written as

$$ {}^{3}_{B}\mathbf{R}(q_1, q_2, q_3) \cdot {}^{B}_{6}\,\mathbf{R} = {}^{3}_{6}\mathbf{R}(q_4, q_5, q_6), \tag{C.11}$$

where the left part of the equation is known and the right part remains to be computed. Pre-multiplying equation C.11 with ${}^{4}_{3}\mathbf{R}(q_4)$ gives

$$ {}^{4}_{3}\mathbf{R}(q_4) \cdot {}^{3}_{B}\mathbf{R}(q_1, q_2, q_3) \cdot {}^{B}_{6}\,\mathbf{R} = {}^{4}_{6}\mathbf{R}(q_5, q_6). \tag{C.12}$$

For the Stäubli RX robots, equation C.12 equals

$$\begin{bmatrix} C4 & 0 & -S4 \\ -S4 & 0 & -C4 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{F} & \mathbf{G} & \mathbf{H} \end{bmatrix} = \begin{bmatrix} C6C5 & -S6C5 & S5 \\ S6 & C6 & 0 \\ C6S5 & -S6S5 & C5 \end{bmatrix}, \tag{C.13}$$

where $\mathbf{F}$ is written as

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} C23(C1 \cdot {}^{B}_{0}r_{xx} + S1 \cdot {}^{B}_{0}r_{xy}) + S23 \cdot {}^{B}_{0}r_{xz} \\ -S23(C1 \cdot {}^{B}_{0}r_{xx} + S1 \cdot {}^{B}_{0}r_{xy}) + C23 \cdot {}^{B}_{0}r_{xz} \\ S1 \cdot {}^{B}_{0}r_{xx} - C1 \cdot {}^{B}_{0}r_{xy} \end{bmatrix}. \tag{C.14}$$

$\mathbf{G}$ and $\mathbf{H}$ are obtained from $\mathbf{F}$ by replacing $[r_{xx}\ r_{xy}\ r_{xz}]$ by $[r_{yx}\ r_{yy}\ r_{yz}]$ and $[r_{zx}\ r_{zy}\ r_{zz}]$ respectively.

Two solutions are computed for q_4 as

$$q_4 = \arctan(H_z, -H_x) \tag{C.15}$$
$$q_4' = q_4 + \pi. \tag{C.16}$$

The solution for q_5 is computed as

$$q50 = \text{arctan2}(-S5, C5), \tag{C.17}$$

with

$$C5 = H_y$$
$$S5 = -C4 * H_x + S4 * H_z.$$

Finally, the solution for q_6 is computed as

$$q_6 = \arctan2(S6, C6) \tag{C.18}$$

with

$$S6 = -C4 * F_z - S4 * F_x$$
$$C6 = -C4 * G_z - S4 * G_x.$$

The computations for the IGM are more complex than for the DGM. With the robot controller (Celeron 566 Mhz) the IGM can be computed in less than 0.25 ms.

Singularities

In the regular case, the Stäubli RX robots have eight solutions for the IGM. By examining the IGM solution of the Stäubli RX robots it can be observed that the robot has the following singular positions:

- Shoulder singularity: Occurs when the intersection of joint 4, 5 and 6 lies on the z-axis of the base frame. In that case, ${}^{B}_{6}P_x ={}^{B}_{6} P_y = 0$, which corresponds to $S23 \cdot RL4 + D3 \cdot S2 = 0$. For $S23 \cdot RL4 + D3 \cdot S2 < 0$, the shoulder configuration is called 'righty', otherwise it is called 'lefty'.
- Elbow singularity: Occurs when $C3 = 0$. If the shoulder configuration is 'righty' and $q_3 < 0$ or if the shoulder configuration is 'lefty' and $q_3 > 0$, the elbow configuration is called 'above'. Otherwise, the elbow configuration is called 'below'.
- Wrist singularity: Occurs when $q_5 = 0$. The wrist configuration is called 'negative' if $q_5 < 0$, otherwise it is called 'positive'.

If desired, the above-mentioned expressions are used to select one of the eight possible IGM solutions for a specific robot arm configuration.

Publications

De Graaf, M. W., Aarts, R. G. K. M., Jonker, J. B., and Meijer, J. (**2006**). "Real-time trajectory generation for sensor-guided robotic laser welding." In "Proceedings of the Symposium on Robot Control (Syroco) 2006," IFAC, Bologna.

De Graaf, M. W., Aarts, R. G. K. M., Jonker, J. B., and Meijer, J. (**2007**a). "Ethernet-based communication framework for sensor integration on industrial robots." In "Proceedings of the First International Conference on Robot Communication and Coordination," Create-Net, Athens.

De Graaf, M. W., Aarts, R. G. K. M., Jonker, J. B., and Meijer, J. (**2007**b). "Real-time seam tracking for robotic laser welding using trajectory-based control." *Submitted to Control Engineering Practice*.

De Graaf, M. W., Aarts, R. G. K. M., Meijer, J., and Jonker, J. B. (**2004**a). "The influence of geomtric errors in the nominal robot model on sensor-guided robotic laser welding." In "Proceedings of the 23rd International Congress on Applications of Lasers and Electro-Optics 2004 (ICALEO)," Laser Institute of America.

De Graaf, M. W., Aarts, R. G. K. M., Meijer, J., and Jonker, J. B. (**2004**b). "Modeling the seam teaching process for robotic laser welding." In P. Drews, editor, "Proceedings of the Mechatronics & Robotics 2004 conference," APS - European Centre for Mechatronics.

De Graaf, M. W., Aarts, R. G. K. M., Meijer, J., and Jonker, J. B. (**2005**a). "Robot-sensor synchronization for real-time seam tracking in robotic laser welding." In "Proceedings of the Third International WLT-Conference on Lasers in Manufacturing," Munich.

De Graaf, M. W., Aarts, R. G. K. M., Meijer, J., and Jonker, J. B. (**2005**b). "Robust Process-controller for Nd:YAG laser welding." In "Proceedings of the 24th International Congress on Applications of Lasers and Electro-Optics 2005 (ICALEO)," .

Bibliography

Abo-Hammour, Z. S., Mirza, N. M., Mirza, S. M., and Arif, M. (2002). "Cartesian path generation of robot manipulators using continuous genetic algorithms." *Robotics and Autonomous Systems*, 41: 179–223.

Abramowitz, M. and Stegun, I. A. (1972). *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. Dover Publications, New York.

Ahmed Bazaz, S. and Tondu, B. (1999). "Minimum time on-line joint trajectory generator based on low order spline methods for industrial manipulators." *Robotics and Autonomous Systems*, 29: 257–268.

Andersen, H. J. (2001). *Sensor Based Robotic Laser Welding - Based on Feed Forward and Gain Scheduling Algorithms*. Ph.D. thesis, Aalborg University.

Bae, K. and Park, J. (2006). "A study on development of inductive sensor for automatic weld seam tracking." *Journal of Materials Processing Technology*, 176: 111–116.

Bae, K. Y., Lee, T. H., and Ahn, K. C. (2002). "An optical sensing system for seam tracking and weld pool control in gas metal arc welding of steel pipe." *Journal of Materials Processing Technology*, 120: 458–465.

Barnhill, R. E. and Riesenfled, R. F. (1974). *Computer Aided Geometric Design*. Academic Press New York.

Barr, A. H., Currin, B., Gabriel, S., and Hughes, J. F. (1992). "Smooth interpolation of orientations with angular velocity constraints using quaternions." In "Proceedings of the 19th annual conference on Computer graphics and interactive techniques," pages 313–320. ACM Press, New York, USA.

Bauchspiess, A., Absi Alfaro, S. C., and Dobrzanski, L. A. (2001). "Predictive sensor guided robotic manipulators in automated welding cells." *Journal of Materials Processing Technology*, 109: 13–19.

Carozzi, C., Magnani, G., and Nicolodi, S. (1995). "Implementation of sensor-based control algorithms in an industrial robot controller." *Control engineering practice*, 3: 1307–1313.

Chou, J. C. K. and Kamel, M. (**1988**). "Quaternions approach to solve the kinematic equation of rotation AaAx = AxAb of a sensor mounted robotic manipulator." In "Proceedings of the 1988 IEEE International Conference on Robotics and Automation," pages 656–662. Philadelphia.

Chou, J. C. K. and Kamel, M. (**1991**). "Finding the position and orientation of a sensor on a robot manipulator using quaternions." *The International Journal of Robotics Research*, 10: 240–254.

Conrad, K. L., Shiakolas, P. S., and Yih, T. C. (**2000**). "Robotic calibration issues: Accuracy, repeatability and calibration." In "Proceedings of the 8th Mediterranean Conference on Control & Automation (MED 2000)," .

Corke, P. I. (**1994**). *High-Performance Visual Closed-Loop Robot Control*. Ph.D. thesis, University of Melbourne, Dept. Mechanical and Manufacturing Engineering.

Craig, J. J. (**1986**). *Introduction to robotics: Mechanics & Control*. Addison-Wesley publishing company, Inc.

Cremers, A. (**2006**). *Dynamic simulation of a Stäubli RX90 robot with on-the-fly linearization*. Master's thesis, University of Twente. Internal report WA-1069.

Daniilidis, K. (**1999**). "Hand-Eye Calibration Using Dual Quaternions." *The International Journal of Robotics Research*, 18(3): 286–298.

Davies, R. B. (**2002**). "Newmat10: A matrix library in C++." URL `http://www.robertnz.net/`.

De Roo, S. (**2003**). *Testing the trajectory performance of the Stäubli RX90 robot for laser welding*. Master's thesis, University of Twente. Internal report WA-889.

Denavit, J. and Hartenberg, R. S. (**1955**). "A kinematic notation for lower-pair mechanisms based on matrices." *ASME Journal of Applied Mechanics*, pages 215–221.

Diphoorn, A. (**2004**). *The influence of geometric errors in the nominal robot model on sensor-guided robotic laser welding*. Master's thesis, University of Twente. Internal report WA-943.

Duley, W. (**1999**). *Laser Welding*. Willey Interscience publications.

During, H. J. (**2005**). *Development and implementation of a real-time seam teaching algorithm*. Master's thesis, University of Twente. Internal report WA-988.

Eberly, D. (**1999**). "Quaternion Algebra and Calculus."

Falldorf Sensor (**2006**). URL `http://www.falldorfsensor.com`.

Fridenfalk, M. and Bolmsjö, G. (**2003**). "Design and validation of a universal 6D seam tracking system in robotic welding based on laser scanning." *Industrial robot: An international journal*, 30(5): 437–448.

Hakvoort, W. B. J., Aarts, R. G. K. M., Van Dijk, J., and Jonker, J. B. (**2006**). "Iterative Learning Control for improved end-effector accuracy of an industrial robot." In "Proceedings of Symposium on Robot Control," IFAC.

Hamilton, W. R. (**1853**). "Lectures on Quaternions." Hodges and Smith.

Hardeman, T., Aarts, R. G. K. M., and Jonker, J. B. (**2006**). "Modelling and Identification of Robots with Joint and Drive Flexibilities." In "IUTAM Symposium on Vibration Control of Nonlinear Mechanisms and Structures," Springer Netherlands.

Hoffmann, G. (**2002**). *Applications of Quaternions*. University of Braunschweig.

Huissoon, J. P. (**2002**). "Robotic laser welding: seam sensor and laser focal frame registration." *Robotica*, 20: 261–268.

Iakovou, D., Aarts, R., and Meijer, J. (**2005**). "Sensor integration for robotic laser welding processes (Paper #2301)." In "Proceedings of the International Congress on Applications of Lasers and Electro-Optics (ICALEO), Miami, USA," .

IEEE1588 (**2002**). "1588: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems." Technical report, IEEE Instrumentation and Measurement Society.

IPL98 (**1998**). "Image Processing Library." URL `http://www.mip.sdu.dk/ipl98/`.

Khalil, W. and Dombre, E. (**2002**). *Modeling, Identification & Control of Robots*. Hermes Penton Ltd.

Kim, J. S., Son, Y. T., Cho, H. S., and Koh, K. I. (**1996**). "A robust visual seam tracking system for robotic arc welding." *Mechatronics*, 6: 141–163.

Kim, P., Rhee, S., and Lee, C. H. (**1999**). "Automatic seam teaching of welding robot for free-formed seam using laser vision system." *Optics and Lasers in Engineering*, 31: 173–182.

Kreyszig, E. (**1991**). *Differential Geometry*. New York: Dover.

Krypton (**2001**). *Krypton help pages on robot performance testing*. Krypton Electronic Engineering N.V.

Lee, S. and Ro, S. (1996). "A Self-Calibration Model for Hand-Eye Systems with Motion Estimation." *Mathematical and Computer Modelling*, 24(5/6): 49–77.

Liu, Y., Hoover, A. W., and Walker, I. D. (2004). "A timing model for vision-based control of industrial robot manipulators." In "IEEE Transactions on Robotics," .

Lu, C. P., Mjolsness, E., and Hager, G. D. (1996). "Online Computation of Exterior Orientation with Application to Hand-Eye Calibration." *Mathematical and Computer Modelling*, 24(5/6): 121–143.

Luo, H. and Chen, X. (2005). "Laser visual sensing for seam tracking in robotic arc welding of titanium alloys." *International Journal of Advanced Manufacturing Technology*, 26: 1012–1017.

Nayak, N. and Ray, A. (1993). *Intelligent seam tracking for robotic welding*. Advances in Industrial Control. Springer-Verlag.

Nunes, U., Faia, P., and De Almeida, A. T. (1996). "Sensor-based 3-D autonomous surface-following control." *Mathematics and computers in simulation*, 41: 429–444.

Olde Benneker, J. (2000). "Toelaatbare toleranties bij het laserlassen van aluminiumlegeringen." Technical report, University of Twente, Enschede. Technical report WA-683.

Park, F. C. and Martin, B. J. (1994). "Robot sensor calibration: Solving AX equals XB on the Euclidian Group." In "IEEE Transactions on Robotics and Automation," pages 717–721.

Paul, R. P. (1982). *Robot manipulators: mathematics, programming and control*. MIT Press, Cambridge, MA, USA.

Pertin, F. and Bonnet-des-Tuves, J. M. (2004). "Real time Robot Controller Abstraction Layer." In "Proceedings of the International Symposium on Robotics 2004," .

Rensen, M. B. (2006). *Development and implementation of a real-time seam tracking algorithm*. Master's thesis, University of Twente. Internal report WA-1044.

Roeleveld, T. (2004). *Design of a coaxial head for laser welding*. Master's thesis, University of Twente. Internal report WA-931.

Römer, G. R. B. E. (2002). "Hoogvermogen lasers voor het bewerken van metalen." Vereniging FME-CWM.

Sanderson, A. C. and Weiss, L. E. (**1980**). "Image-based visual servo control using relational graph error signals." In "Proceedings of the International Conference on Cybernetics and Society," Cambridge.

Schwab, A. L. (**2002**). *Dynamics of Flexible Multibody Systems*. Ph.D. thesis, Delft University of Technology.

Sercos (**1995**). *Sercos interface: Technical Short Description*. Sercos.

Sgarbi, F. and Cammoun, R. (**1992**). "Real time trajectory generation using filtering techniques." In "Proceedings of the 2nd International Conference on Automation, Robotics and Computer Vision," Singapore.

Shiu, Y. C. and Ahmad, S. (**1989**). "Calibration of wrist-mounted robotic sensors by solving homongenous transform equations of the from AX=XB." In "IEEE Transactions on Robotics and Automation," .

Shoemake, K. (**1985**). "Animating rotation with quaternion curves." In "Proceedings of the 12th annual conference on Computer graphics and interactive techniques," pages 245–254. ACM Press, New York, USA.

Stäubli (**2004**). *VAL3 Reference Manual version 4.0*. Stäubli, Faverges.

Su, Y. (**2000**). "Laser welding of stainless steel in different orientations." In "Proceedings of the 19th International Congress on Applications of Lasers and Electro-Optics 2000 (ICALEO)," pages C210–C218. Dearborn, USA.

Thorne, H. F. (**1999**). "Tool center point calibration for spot welding guns." US Patent (US5910719).

Trolltech (**2005**). "Qt - GUI software toolkit." URL `http://www.trolltech.com/`.

Tsai, R. Y. and Lenz, R. K. (**1989**). "A new technique for fully autonomous and efficient 3D robotics Hand/Eye calibration." In "IEEE Transactions on Robotics and Automation," pages 345–357.

Van der Heijden, F. (**1994**). *Image Based Measurement Systems: Object Recognition and Parameter Estimation*. John Wiley.

Van der Velde, F. (**2005**). *Development of an automatic calibration procedure for the sensor TCP*. Master's thesis, University of Twente. Internal report WA-989.

Van der Zee, T. W. (**2003**). *Modeling the seam teaching process for laser welding applications*. Master's thesis, University of Twente. Internal report WA-879.

Van Tienhoven, J. (**2004**). *Automatic tool center point calibration for a laser welding robot*. Master's thesis, University of Twente. Internal report WA-919.

Waiboer, R. R. (**2007**). *Dynamic Modelling, Identification and Simulation of Industrial Robots - for Off-line Programming of Robotised Laser Welding*. Ph.D. thesis, University of Twente - Netherlands Institute for Metals Research.

Wind River (**2002**). "VxWorks Programmer's Guide version 5.5." URL `http://www.windriver.com`.

XML (**2003**). "Extensible Markup Language (XML)." URL `http://www.w3.org/XML/`.

Yu, J. Y. and Na, S. J. (**1997**). "A study on vision sensors for seam tracking of height-varying weldment. Part 1: Mathematical model." *Mechatronics*, 7: 599–612.

Yu, J. Y. and Na, S. J. (**1998**). "A study on vision sensors for seam tracking of height-varying weldment. Part 2: Applications." *Mechatronics*, 8: 21–36.

Zhuang, H. and Shiu, Y. C. (**1993**). "A noise-tolerant algorithm for robotic hand-eye calibration with or without orientation measurement." In "IEEE Transactions on Systems, Man and Cybernetics," pages 1168–1175.

Summary

Robotic laser welding is a promising joining technique for welding of continuous seams in 3D products. High quality joints can be realised provided that the manipulation of the laser beam and the product tolerances meet strict criteria. The requirements for the positioning accuracy of the laser beam can be fulfilled using seam-tracking sensors. This thesis describes how such sensors can be integrated in a robotic laser welding system for automatic teaching of the seam trajectory, but also to correct small errors (e.g. clamping errors, heat deviation errors) from a pre-defined seam trajectory.

Both the sensor and laser are tools that are attached to the robot flange. The sensor measures relative to its coordinate frame, but the corrections need to be applied in the laser coordinate frame. The transformations (position and orientation) between these coordinate frames and the robot flange need to be accurately calibrated. For this purpose, three tool calibration procedures are developed. In two of these procedures, the transformation of the sensor tool frame and the laser tool frame relative to the robot flange are obtained, by making a number of movements above special calibration objects. Both experiments and simulations are performed, which shows that the accuracy of these procedures is influenced by errors in the robot geometric model. To overcome these accuracy limitations, a third calibration procedure is developed, which is a direct measurement of the sensor tool frame relative to the laser tool frame that does not depend on robot movements.

Two point-to-point seam teaching algorithms have been described for automatic teaching of seam trajectories without prior knowledge of their geometry. With these algorithms, 3D seam trajectories can be welded in two steps: first sensor-guided teaching, then blind welding. A simulation environment has been developed, which allows to see the influence of various errors that occur in a sensor-guided robotic laser welding environment. Both experiments and simulations have been performed to demonstrate that the teaching algorithms are working appropriately.

To use a seam-tracking sensor that measures ahead of the laser focal point for real-time seam tracking, a new trajectory-based control approach is presented. In this control approach, sensor measurements are related to the robot position to build a geometric seam trajectory, which is followed by the laser focal point.

To be able to use the trajectory-based control approach a real-time Setpoint Generator based on cubic interpolation and a method to synchronise the robot joint measurements and the sensor image acquisition were implemented. A real-time seam-tracking algorithm has been implemented, where the orientation of a measured seam location is corrected to keep the sensor within the field-of-view of a pre-defined seam trajectory once the laser reaches the measured location. The corrected locations are filtered in the position-domain. The laser moves at a constant velocity through these filtered and corrected locations, while the sensor simultaneously measures new locations. Experiments have been performed to show that the trajectory-based control approach is well suited for real-time seam tracking.

For real-time seam tracking, the accuracy of the laser focal relative to the seam trajectory is influenced by the curvature of the seam trajectory and the look-ahead distance between the sensor and laser due to geometric robot errors. Welding heads with integrated sensors can not be used for real-time seam tracking due to the small or zero look-ahead distance. Therefore, a real-time teaching algorithm that fits within the trajectory-based control framework is developed and implemented. Especially for largely curved seam trajectories, the accuracy is considerably increased with this method.

Samenvatting

Gerobotiseerd laserlassen is een veelbelovende techniek voor het lassen van continue 3D naden. Een kwalitatief goede verbinding kan worden gerealiseerd, als stricte criteria aan de positionering van de laser bundel en de product toleranties worden gesteld. Deze positioneringseisen kunnen worden gerealiseerd door gebruik te maken van naadvolgsensoren. Dit proefschrift beschrijft hoe deze sensoren kunnen worden geïntegreerd in een gerobotiseerd laserlassysteem voor het automatisch inleren van het lastraject, maar ook om kleine correcties (b.v. opspanfouten, vervormingsfouten) ten opzichte van een nominaal traject uit te voeren.

Zowel de sensor als de laser zijn gereedschappen die zijn verbonden aan de robot-flens. De sensor meet in zijn eigen coordinatensysteem. De correcties moeten echter worden toegepast in het laser coordinatensysteem. De transformaties (positie en oriëntatie) tussen deze coordinatensystemen en dat van de robot-flens moeten nauwkeurig worden gecalibreerd. Hiervoor zijn drie tool calibratie-procedures ontwikkeld. In twee van deze procedures worden de transformatie tussen de robot-flens en het laser, respectievelijk sensor coördinatensysteem verkregen door de robot een aantal bewegingen te laten maken boven speciaal ontwikkelde calibratie-objecten. Zowel experimenten als simulaties laten zien dat de nauwkeurigheid van deze procedures in belangrijke mate wordt beïnvloed door geometrische fouten in het robot model. Om de nauwkeurigheid te verbeteren, is er een derde calibratie-procedure ontwikkeld waarin een directe meting van de transformatie tussen sensor en laser coördinatensysteem wordt uitgevoerd. Deze procedure is onafhankelijk van robot bewegingen.

Voor het automatisch inleren van lasnaden zonder voorkennis, zijn twee inleer-algoritmes beschreven die gebruik maken van point-to-point bewegingen. Met deze algoritmes kunnen 3D trajecten worden gelast in twee stappen: eerst sensorgestuurd inleren, daarna blind lassen. Er is een simulatie-omgeving ontwikkeld, waarin de invloed van diverse fouten die optreden in een sensorgestuurd laserlas systeem kan worden gesimuleerd. Zowel experimenten als simulaties laten zien dat de inleer-algoritmes geschikt zijn voor het automatisch inleren van lasnaden.

De naadvolgsensor meet enige afstand voor het laser-focus. Om deze metingen te gebruiken om real-time correcties uit te voeren is een nieuwe zogenaamde

'trajectory-based control'-aanpak gepresenteerd. Hierbij worden sensormetingen gerelateerd aan de robot positie om een geometrisch lastraject op te bouwen, dat wordt gevolgd door het laser focus. Om deze 'trajectory-based control'-aanpak te kunnen gebruiken zijn een real-time Setpoint Generator, gebaseerd op cubische interpolatie, en een methode om de robot-jointmetingen en de sensormetingen te synchroniseren geïmplementeerd. Er is een real-time naadvolg algoritme geïmplementeerd, waarbij de orientatie van een gemeten naadlocatie wordt gecorrigeerd om de sensor binnen het meetbereik van een vooraf gedefineerd lastraject te houden op het moment dat de laser deze locatie bereikt. De gecorrigeerde locaties worden gefilterd in het positie-domein. De laser beweegt met een constante snelheid door deze gefilterde locaties, terwijl de sensor tegelijkertijd nieuwe locaties meet. Experimenten laten zien dat de 'trajectory-based control'-aanpak erg geschikt is voor real-time naadvolgen.

Bij het real-time naadvolgalgoritme hangt de positioneernauwkeurigheid van de laser-spot ten opzichte van de lasnaad af van de kromming van de lasnaad en van de voorloopafstand van de sensor als gevolg van geometrische robot fouten. Laskoppen met geintegreerde sensoren kunnen vanwege hun kleine of niet aanwezige voorloopafstand niet worden gebruikt voor real-time naadvolgen. Daarom is een real-time naadleeralgoritme ontwikkeld dat binnen het 'trajectory-based control' framework past. De nauwkeurigheid wordt met deze methode aanzienlijk verhoogd, in het bijzonder voor sterk gekromde lasnaden.