

Lappeenranta University of Technology
Faculty of Technology
Department of Mechanical Engineering

Veli-Joel Paananen

**DUAL LIGHT BARRIER METHOD FOR SIX DEGREES OF FREEDOM TOOL
CENTER POINT CALIBRATION OF AN INDUSTRIAL ROBOT**

Examiners: Prof. Heikki Handroos
D.Sc. Yongbo Wang

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology
Department of Mechanical Engineering

Veli-Joel Paananen

Dual Light Barrier Method for Six Degrees of Freedom Tool Center Point Calibration of an Industrial Robot

Master's Thesis

2014

62 pages, 23 figures, 4 tables

Examiners: Professor Heikki Handroos
D.Sc. Yongbo Wang

Keywords: robot, calibration, tool center point, tcp, light barrier

Tool center point calibration is a known problem in industrial robotics. The major focus of academic research is to enhance the accuracy and repeatability of next generation robots. However, operators of currently available robots are working within the limits of the robot's repeatability and require calibration methods suitable for these basic applications.

This study was conducted in association with Stresstech Oy, which provides solutions for manufacturing quality control. Their sensor, based on the Barkhausen noise effect, requires accurate positioning. The accuracy requirement admits a tool center point calibration problem if measurements are executed with an industrial robot. Multiple possibilities are available in the market for automatic tool center point calibration. Manufacturers provide customized calibrators to most robot types and tools. With the handmade sensors and multiple robot types that Stresstech uses, this would require great deal of labor.

This thesis introduces a calibration method that is suitable for all robots which have two digital input ports free. It functions with the traditional method of using a light barrier to detect the tool in the robot coordinate system. However, this method utilizes two parallel light barriers to simultaneously measure and detect the center axis of the tool. Rotations about two axes are defined with the center axis. The last rotation about the Z-axis is calculated for tools that have different width of X- and Y-axes.

The results indicate that this method is suitable for calibrating the geometric tool center point of a Barkhausen noise sensor. In the repeatability tests, a standard deviation inside robot repeatability was acquired. The Barkhausen noise signal was also evaluated after recalibration and the results indicate correct calibration. However, future studies should be conducted using a more accurate manipulator, since the method employs the robot itself as a measuring device.

TIIVISTELMÄ

Lappeenrannan Teknillinen Yliopisto
Teknillinen tiedekunta
Konetekniikka

Veli-Joel Paananen

Dual Light Barrier Method for Six Degrees of Freedom Tool Center Point Calibration of an Industrial Robot

Diplomityö

2014

62 pages, 23 figures, 4 tables

Tarkastajat: Professori Heikki Handroos
TkT Yongbo Wang

Hakusanat: robotti, kalibrointi, työkalukeskipiste

Työkalukeskipisteen määrittäminen on tunnettu onglema teollisuusrobotiikassa. Akateeminen tutkimus on pääasiassa keskittynyt uusien robottisukupolvien tarkkuuden kehittämiseen liittyviin ongelmiin, mutta robottien käyttäjät tarvitsevat uusia kalibrointimenetelmiä nykyisten mallien tarkkuuden parantamiseksi.

Tämä työ toteutettiin yhteistyössä laadunvalvontasovelluksia tuottavan Stresstech Oy:n kanssa. Stresstechin Barkhausenin kohinaan perustuva mitta-anturi vaatii huolellista paikoitusta, joten robottisovelluksissa vaaditaan tarkkaa työkalukeskipisteen määrittämistä. Eri työkaluille suunniteltuja kaupallisia sovelluksia on saatavilla, mutta Stresstechin käyttäessä useiden eri valmistajien robotteja jonka lisäksi anturit ovat käsintehtyjä, olisi kalibrointilaitteiston määrittäminen työlästä.

Diplomityössä tutkittiin työkalukeskipisteen kalibrointimenetelmää joka sopisi useisiin eri teollisuusroboteihin. Perinteisestä valosädettä käyttävästä menetelmästä poiketen, tässä menetelmässä käytetään kahta samansuuntaista sädettä määrittämään työkalun reunan sijaintia robotin koordinaatistossa. Reunasijaintien perusteella lasketaan keskipisteet. Pisteden muodostaman keskiakselin orientaatio määrittää työkalupisteen kiertymät. Z-akselin kiertymä ratkaistaan työkalun X- ja Y-akselien leveyksien suhteesta.

Tulokset osoittavat, että menetelmä on soveltuva Barkhausen –anturin geometrisen työkalukeskipisteen määrittämiseen. Kokeiden perusteella menetelmän toistotarkkuus on robotin toistotarkkuuden mukainen. Barkhausen signaalia arvioitiin luomalla anturille virheasento ja kalibroinnin jälkeen mittaustulokset olivat vastaavat kuin ennen kalibrointia. Jatkotutkimuksia tulisi kuitenkin suorittaa tarkemmalla robotilla, sillä menetelmä käyttää robottia itsessään mittalaitteena, jolloin laitteiston virhettä ei voi erottaa robotin virheestä.

CONTENTS

1. INTRODUCTION	8
1.1. Background	8
1.2. Research problem and focus	9
1.3. Implementation and structure.....	10
2. MAGNETIC BARKHAUSEN NOISE	12
2.1. Ferromagnetic domain theory	12
2.2. Domain Wall Movement.....	13
3. MATHEMATICAL MODELING IN ROBOTICS.....	15
3.1. Position and orientation in task space	15
3.2. Three angle representations.....	17
3.3. Homogeneous transformation	20
3.4. Geometric model of the manipulator	21
3.4.1. Denavit-Hartenberg parameters.....	22
3.4.2. Kinematic equations	24
4. ROBOT CALIBRATION.....	25
4.1. Accuracy and repeatability.....	25
4.2. Calibration of the manipulator	27
4.3. Tool center point calibration	28
4.3.1. Manual calibration of the tool center point.....	29
4.3.2. Automatic calibration of the tool center point	29
5. THE STUDIED CALIBRATION METHOD	32
5.1. Calibration setup	33
5.2. Universal robots	33
5.3. Barkhausen noise sensor	36

5.4.	Calibrator equipment.....	37
5.5.	Installation in the robot work envelope.....	38
5.6.	Calibration procedure.....	40
6.	QUALITATIVE MEASUREMENTS.....	46
7.	RESULTS.....	49
7.1.	Maximum acceleration.....	49
7.2.	Repeatability of the calibration method.....	50
8.	ANALYSIS.....	55
8.1.	Acceleration.....	55
8.2.	Repeatability.....	56
8.3.	Barkhausen noise signal quality.....	57
9.	CONCLUSIONS AND DISCUSSION.....	59
10.	SUMMARY.....	61
	REFERENCES.....	62

LIST OF SYMBOLS

a_i	[m]	link length
d_i	[m]	link offset
E_i	[J]	theoretical energy
$e_{i,zj}$	-	measured edge axis, number and height
i_{zj}	-	calculated center point, number and height
\mathbf{P}	-	vector of pose
\mathbf{R}	-	rotation matrix
\mathbf{S}	-	vector of position
\mathbf{T}	-	homogeneous transformation matrix
$w_{i,zj}$	-	measured width, number and height
α_i	[rad]	link twist
β_i	[rad]	rotation about y-1
Δ	[m]	distance between light barriers
θ_i	[rad]	link angle
η	-	vector of geometrical parameters

LIST OF ABBREVIATIONS

BN	Barkhausen noise
DH-parameters	Denavit-Hartenberg parameters
GMAW	Gas Metal Arc Welding
GUI	Graphical User Interface
I/O	Input / Output bus
ISO	International Organization for Standardization
PNP	Positive-Negative-Positive Layered Semiconductor
SCBU	Single Beam Calibration Unit
TCP	Tool Center Point
URscript	Universal Robots scripting language

1. INTRODUCTION

1.1. Background

A robot controller solves a series of mathematical equations when a manipulator is requested to perform a task. These equations are mainly based on the model of the robot's kinematics. It is obvious that the accuracy of this model describing the physical structure directly affects the success of the requested task. When considering early pick and place robots, which were the first steps in industrial robotics, the mathematical model was rather simple and was directly transferrable between similar units. However, when industrial robots became more complex and their tasks required high precision, the mathematical modeling methods were no longer sufficient; even a completely equivalent unit could not repeat the task exactly as the previous one did (Mooring et al. 1991). This increased the labor required for programming and reduced efficiency when increasing capacity by multiplying robot cells.

The position of the robot's end-effector that actually performs the task can be several millimeters off from that requested if the robot or its components are replaced. This is due to differences in the structure and unique characteristics of components of robots. It was realized that each robot should be considered as unique, therefore the mathematical model needs to be defined individually. This is easier than trying to create a completely equivalent physical structure (Roth et al. 1987, p.377). Robots are calibrated by alternating parameters in the embedded software of a controller. To fully calibrate the manipulator, the pose that describes the position and orientation of the end-effector is defined mathematically, then the robot is requested to perform this pose. If simplified, the difference between the mathematical pose and the acquired pose leads to calibration parameters, since the difference is the error caused by the inaccuracy of the mathematical model.

Another important aspect in addition to manipulator calibration is the basis for this study. The mathematical model embedded into the controller also takes into account the end-effector. However, the default equations are considered to control only the tool flange to which the end-effector is attached. Since the end-effector is selected by task, for example

one robot might use a welding gun for welding tasks and a completely different kind of grasp for lifting tasks, the controller also needs to be programmed with the parameters for the current end-effector. These parameters are the translations and rotations of a coordinate frame whose origin is defined to be at the most important point of the tool. This origin is referred as the tool center point, commonly abbreviated as TCP. After this has been defined, the robot controls the pose of this frame. For example, for gas metal arc welding the origin of the frame should be defined to the tip of the welding wire that is protruding from the gun. In this case the robot is programmed to move the tool center point, this being the tip of the wire, in a precise path over the groove. However, if the welding gun accidentally hits an obstacle and becomes bent, the robot controller is unaware of this. The pre-defined translations and rotations will be incorrect and the welding wire will now be off from the groove, which causes direction errors and shapes an imperfect weld. When the robot operator notices this, they must bend the gun back into its original shape or recalibrate the tool center point to the wire tip's new position.

Manual tool center point calibration takes time and requires a skilled operator. Briefly, the robot arm is moved into different poses and the TCP is programmed by holding it in a stationary position during the movements. Automatic calibration units were designed to reduce down time and to improve calibration accuracy. This study was conducted in association with Stresstech Oy to research suitable automatic calibration methods for less traditional end-effectors.

1.2. Research problem and focus

Stresstech Oy provides quality control solutions for various manufacturing processes. Their applications are able to monitor residual stresses, retained austenite contents, grinding burns, hardness changes and welding stresses. These quality control applications are mainly based on Barkhausen noise effect and Röntgen radiation diffraction. This study is focused on the Barkhausen noise sensor which is particularly used to evaluate the microstructural properties of ferromagnetic materials. The research problem arises from the fact that the sensors are handmade and therefore have small deviations in shape. This requires each sensor to be calibrated individually for the robot. The general research

problem is stated as follows: how to calibrate a tool center point on a sensor that measures abrupt changes in magnetization?

The magnetic Barkhausen phenomena introduce a difficult problem in terms of calibration. Barkhausen noise is excited by a varying magnetic field produced by a coil. The noise is captured by a secondary pick-up coil which reveals the abrupt changes in the magnetization of the specimen. The voltage induced to the secondary coil, and therefore the signal quality, is dependent on the position of the magnetization coil in relation to the specimen and to the secondary coil. It is considered that the actual or *optimal* tool center point would be in the position of the sensor where these relations produce the strongest and clearest signal. It is not possible to uniquely define this position from the geometric properties of the sensor; it is dependent on inner properties. However, considering that the sensor's geometric shape and especially the end point of the sensor reflects the position from where a constant translation would lead to this optimal tool center point, the calibration can be conducted using traditional methods.

The focus of this study was to find a method for defining the constant translation from the robot tool flange to the tip of the sensor, which would eventually increase the probability that the optimal TCP was being controlled properly. Because of the sensor properties, the calibration method was limited to rotationally symmetrical tools which have the end point at the center axis of the tool. The method is general and able to be modified for different robot units and different tools.

1.3. Implementation and structure

The study began by exploring TCP calibrators currently available on the market. A lack of academic papers and the existence of only a few commercial manufacturers indicated that the optimal solution would be to study an alternative and customized method for the calibration. In this way, the calibrator could be designed especially for Barkhausen noise sensors. The proposed calibration method is an incremental development of a single beam calibration unit and is designed to be simple, transferrable between robots and low-cost.

The theoretical background for the study is mainly introduced in Chapters 2 and 3. Chapter 2 defines the phenomena of magnetic Barkhausen noise, a crucial aspect of the beginning of the study, since the most important target of the calibration method is to increase the signal quality when measuring Barkhausen noise with an industrial robot. Chapter 3 is focused on mathematical modeling in robotics. This chapter presents general mathematical equations and conventions used in robot kinematics. Even though TCP calibration requires only a small portion of these equations, they are the foundation of the entire thesis. One cannot truly understand the cooperative behavior of the robot controller and the calibrator without being familiar with coordinate transformations in robot kinematics.

Chapter 4 is focused on robot calibration. Commonly used terms such as repeatability, accuracy and resolution are defined and linked with the importance of calibration. This chapter also introduces the calibration of the entire manipulator and the geometric and non-geometric errors that are the primary reason for the need for manipulator calibration. Tool center point calibration is introduced in both manual and automatic methods. The end of the chapter briefly discusses some major manufacturers' automatic TCP calibration units.

The fifth chapter is the result of the study based on the previous chapters and introduces the calibration method that is proposed as an answer to the research question. Since the method is universal, the chapter begins by describing its principle. Then the components, design and equipment used in this study are introduced. Finally the entire calibration method that binds the method and equipment together is explained using flowcharts and thorough explanations.

Chapter 6 is focused on describing the quantitative methods used in the evaluation of this calibration method. Each measurement set is explained, including the number of samples used in data acquisition, key parameters and variables. The results of these measurements are shown in Chapter 7, with explanations, graphs and tables. The results are analyzed in Chapter 8 and discussed in Chapter 9.

2. MAGNETIC BARKHAUSEN NOISE

2.1. Ferromagnetic domain theory

Magnetic domains are considered as spontaneously magnetized regions inside ferromagnetic material. Principally, the domains are established because of the relations of atomic dipole moments (Sorsa 2013, p.19.) Instead of one magnetic area through the entire piece, ferromagnetic material favors segregation into several diminutive domains to reach a state of minimum internal energy. Five basic energies affect domain appearance according to Formula (1) (Stefanita 1999, p.10).

$$E = E_{exchange} + E_{magnetostatic} + E_{magnetocrystalline} + E_{magnetoelastic} + E_{wall} \quad (1)$$

From the Barkhausen noise point of view, energy related to the domain wall is the most interesting. The behavior of the wall is however dependent on the equilibrium of all magnetic energies as they tend to reach minimum state.

For two domains having opposite magnetization, the transition between magnetization moments spawns an area referred to as the domain wall. In the domain wall, the magnetization moments gradually change direction, as seen in Figure 1. In iron, two types of domain walls are formed due to the easy axes; a 180° domain wall for opposite magnetic moments and a 90° domain wall for perpendicular magnetic moments. (Stefanita 1999, p.13; Santa-aho 2012, p.9). Without an external magnetic field, the domains are randomly oriented and the overall magnetization of the material is zero (Sorsa 2013, p.19).

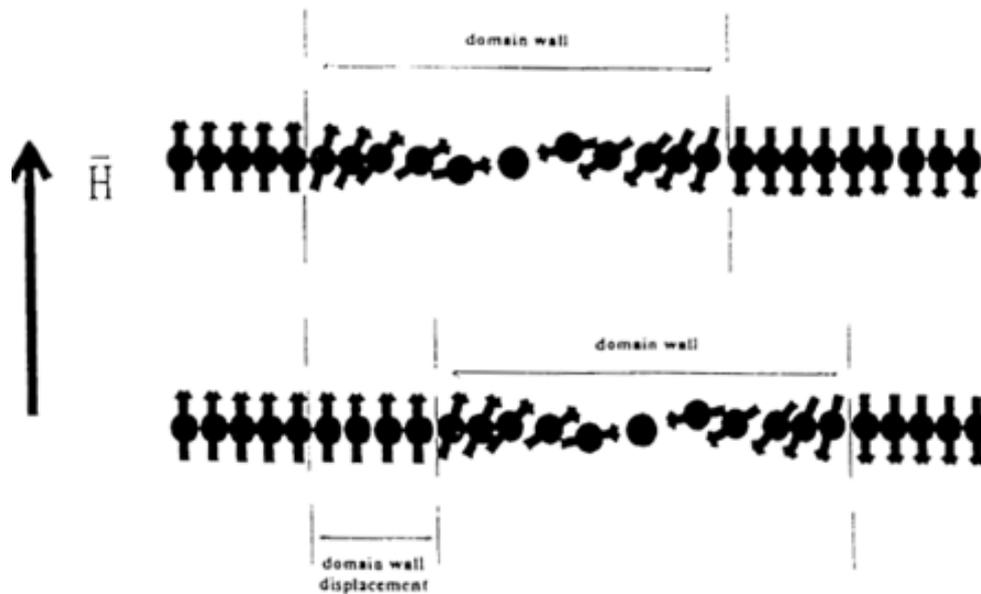


Figure 1. Domain wall movement. H is the applied external magnetic field. (Stefanita 1999)

2.2. Domain Wall Movement

The external magnetic field changes the relations of energy components in Formula (1). Consequently, the energies readjust to reach minimum again. The external field attempts to realign individual magnetic moments. However, the crystallographic energy forces the domain magnetizations parallel to the axis of easy magnetization. Therefore the effect of the external field is concentrated on domain wall behavior; at the domain wall the magnetic moments towards the alignment of the external field are realigned. The domain wall displaces as much as the individual moments are aligned with the external field (Stefanita 1999, p.16-17; Sorsa 2013, p.19). In a larger view, domains that have magnetic moment adjacent to the alignment of the external field will increase in volume and domains with smaller or opposite alignment lose volume.

Nonetheless, domains are generally unable to freely change volume. Defects, stress states and microstructural properties, such as grain boundaries, inclusions and dislocations pin the domain wall (Lindgren 2003, p.33). A static external magnetic field is incapable of forcing domain walls over all of the pinning sites. When the intensity of the external field

is increased to overcome pinning sites, the domain wall suddenly displaces irreversibly. This abrupt domain volume change is the basis for the magnetic hysteresis of ferromagnetic materials and the source of Barkhausen noise (Stefanita 1999, p.17).

As concluded, magnetic hysteresis of ferromagnetic material is due to the irreversible movements of domain walls when excited with an external magnetic field. Increasing the external magnetic field eventually leads to maximum magnetization of the material, where each domain has magnetization forced over the easy axes and only one magnetic domain exists with parallel alignment to an external field. Decreasing the intensity of the external field does not demagnetize the piece within the same curve as it was magnetized. When the external field is reduced to zero, the material still has remanence magnetization. Remanence is left over because the zero external field does not force the domain walls to overcome the last pinning state that was jumped in the magnetization phase. To obtain zero magnetization of the material again, an opposite external field must be applied. The amount of opposite external field needed to zero magnetize the piece again is termed coercivity (Sorsa 2013, p.19; Lindgren 2003, p.26).

When the piece is gradually magnetized, the irreversible discontinuous movements of the domain walls under pinning sites create sudden changes in the magnetization. These changes produce a signal that was amplified and listened to as a crackling noise in the first experiments by Professor Barkhausen (Lindgren 2003, p.25.) The Barkhausen signal is acquired by magnetizing the specimen using a magnetizing coil and analyzing the data recorded by a secondary or pick-up coil. Alternating magnetization is performed by either a sinusoidal or triangular wave function (Sorsa 2013, p.21), and the magnetization frequency affects the measurement depth (Santa-aho 2012, p.12). Sudden changes during the magnetization due to domain wall movements induce voltage pulses to the pick-up coil. This noise-like signal is amplified and filtered for analysis. Analysis of Barkhausen noise signals allows evaluation of the stresses, defects and microstructural properties of the specimen.

3. MATHEMATICAL MODELING IN ROBOTICS

Definition of an object's position, orientation and velocity in a reference coordinate system is essential in robot kinematics. The position and orientation of an end-effector is described in geometrical terms in a coordinate frame and the relations are acquired from coordinate transformations between frames. These coordinates are usually expressed in a Cartesian frame. However, spherical or cylindrical coordinates are also applicable. This chapter provides familiarization with the basic mathematics in robot kinematics.

3.1. Position and orientation in task space

At least six parameters are necessary to fully describe the position and orientation of an object in a Euclidean space. The position of an object is described with a coordinate vector between the origins of a reference frame and a frame attached to the object. Rotations describe the orientation of an object frame about the reference frame axes. These six parameters are referred as the *pose* of the object (Lenarčič et al. 2013, p.8.) In the simplest notation in robotics, the pose of an end-effector is described in a task space where the constraints and joints of the manipulator structure are neglected and only the pose required for the task in hand is expressed as a vector (Mooring et al. 1991, p.2; Siciliano & Kahtib, p.10).

Consider point S which is declared in Cartesian coordinate frame $\{B\}$ with vector \mathbf{S}_B . As seen in Figure 2a, frame $\{B\}$ is rotated in respect to frame $\{A\}$. The position of frame $\{B\}$ in the frame $\{A\}$ is expressed with another vector ${}^A\mathbf{P}_B$ between the origins. Vector \mathbf{S}_B introduces a displacement in addition to the frame position. If the axes of frame $\{B\}$ was aligned with the axes of frame $\{A\}$, the vector \mathbf{S}_A is the sum of vectors ${}^A\mathbf{P}_B + \mathbf{S}_B$. In this case, frame $\{B\}$ would be purely translated in respect to frame $\{A\}$. However, frame $\{B\}$ has a different orientation due to the rotations. In order to solve vector \mathbf{S}_A , a rotation matrix \mathbf{R} is required to compensate the rotations of the frame $\{B\}$. Applying the rotation matrix yields the general solution of Formula (2).

$$\mathbf{S}_A = {}^A\mathbf{R}_B \mathbf{S}_B + {}^A\mathbf{P}_B \quad (2)$$

To illustrate only the rotations, the origin of frame {B} is transferred to the origin of frame {A} as seen in Figure 2b. It should be noted that the both frames are spanned by unit vectors. The unit vectors of {B} are projected at the frame {A} using a dot product. For example, vector \mathbf{X}_B is described in frame {A} using a dot product:

$${}^A\mathbf{X}_B = \begin{bmatrix} \mathbf{X}_B \cdot \mathbf{X}_A \\ \mathbf{X}_B \cdot \mathbf{Y}_A \\ \mathbf{X}_B \cdot \mathbf{Z}_A \end{bmatrix}$$

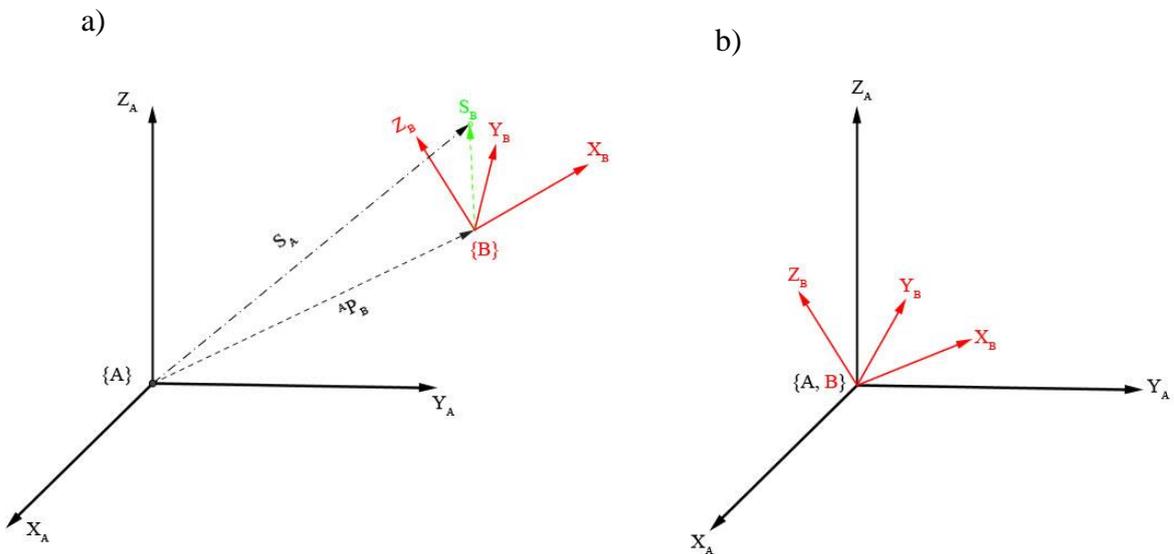


Figure 2. a) Translation of the frame. b) Rotation of the frame

The rotation matrices' columns are the unit vectors of {B} represented at {A}. Therefore, the matrices' elements are direction cosines of the angles between the axes of the frames. (Bajd 2013, p.18; Lenarčič et al. 2013. p.21). The direction cosines from {B} to {A} are the principal functions of the general rotation matrix:

$${}^A\mathbf{R}_B = \begin{bmatrix} \cos \theta_{X_B X_A} & \cos \theta_{X_B Y_A} & \cos \theta_{X_B Z_A} \\ \cos \theta_{Y_B X_A} & \cos \theta_{Y_B Y_A} & \cos \theta_{Y_B Z_A} \\ \cos \theta_{Z_B X_A} & \cos \theta_{Z_B Y_A} & \cos \theta_{Z_B Z_A} \end{bmatrix} \quad (3)$$

The rotation matrix of direction cosines (3) is orthogonal, implying that the inverse of the rotation matrix is also its transpose. This leads to the conclusion that rotation from {A} to {B} is the transpose of rotation from {B} to {A}: ${}^B\mathbf{R}_A = {}^A\mathbf{R}_B^T$. An orthogonal matrix also has the feature that its elements are related to each other. This is seen as nine parameters (Lenarčič 2013, p.20-21; Siciliano & Khatib 2008, p.11).

3.2. Three angle representations

It should be possible to describe the orientation of an object in space using only three parameters instead of nine. The orthogonal rotation matrix of direction cosines is therefore redundant. Redundancy comes from the fact that the direction cosines express the relations of unit-sized vectors that are always perpendicular to each other. To reduce the redundant parameters, the orientation can be described with various representations; Euler angles, Fixed Angles, Angle–Axis and Unit Quaternions (Siciliano & Khatib 2008, p.11.) The first two are referred as three angle representations and are commonly used in robotics.

With a fixed angle representation, the orientation of an object frame is determined with three successive rotations about fixed axes, for example X-Y-Z in angles (ψ, θ, φ) , which is commonly used convention. However, it should be remembered that twelve different possibilities for rearranging the rotation sequence are applicable to both fixed and Euler angles. Fixed angle representations are generally known as Tait-Bryan angles. In aviation and maritime applications they are referred to as RPY-angles (Bajd et al. 2013, p.22-26). In these applications it is beneficial to fix the coordinate frame to the hull of the craft and introduce rotations termed roll, pitch and yaw to describe the axis about which the angle is occurring, as seen in Figure 3. In robotics, Tait-Bryan angles are useful when describing the rotations of the end-effector in a robot coordinate system. In this way, the user can easily determine the pose in the task space, since the robot coordinate system is fixed to the base.

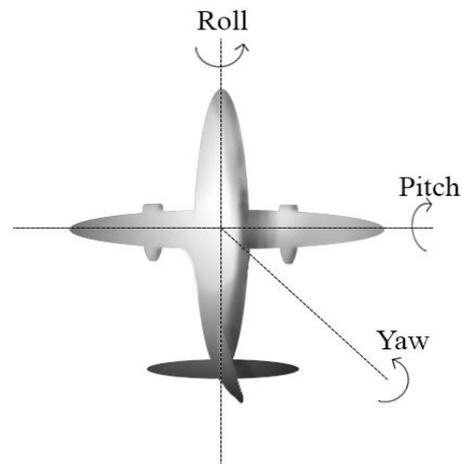


Figure 3. Fixed angle rotations in aircraft

The generally applied method for relative three angle rotations is known as Euler angles. Relative angles are combinations of three successive rotations in such a way that the next rotation occurs in the previously rotated frame. Figure 4 illustrates the procedure of three successive rotations about the Z - Y - Z axes with angles (α, β, γ) . The frame is first rotated about the Z -axis with angle α , which causes the X - and Y -axis to rotate in their plane. The second rotation is done about the new Y' -axis with angle β and the last rotation is done about the Z'' -axis with angle γ (Bajd et al. 2013. p.25.)

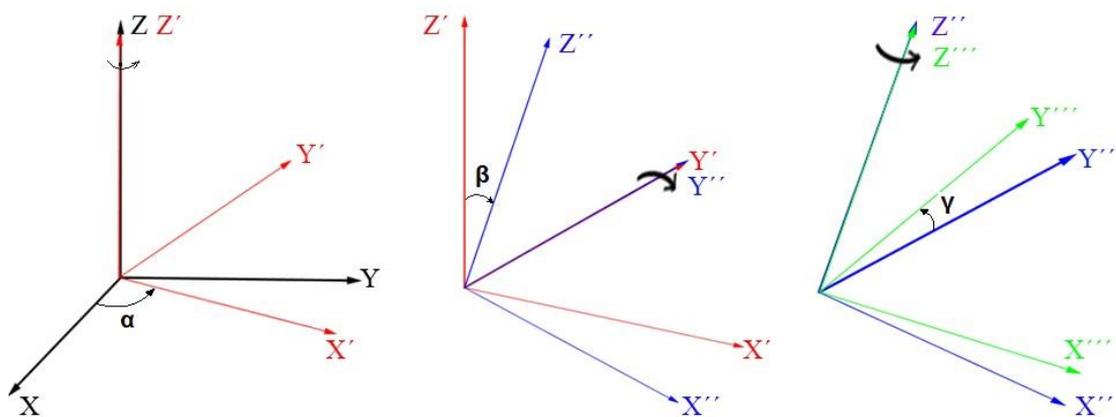


Figure 4. Three successive rotations: Z - Y - Z sequence (α - β - γ rotations)

Each rotation produces an individual rotation matrix shown below (4). The cosine and sine functions are abbreviated “c” and “s” in this notation.

$$\mathbf{R}(\alpha, Z) = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}(\beta, Y') = \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \quad (4)$$

$$\mathbf{R}(\gamma, Z'') = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These rotation matrices are considered as intermediate steps for a complete rotation matrix. Intermediate rotation matrices are multiplied to form a combined matrix describing all three angles (5).

$$\mathbf{R}(\alpha, Z) \mathbf{R}(\beta, Y') \mathbf{R}(\gamma, Z'') = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -s\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\alpha s\gamma & c\beta \end{bmatrix} \quad (5)$$

It is important to note that both Euler and fixed angle representations suffer from singularities in all rotation sequences if the first and last rotations are about same axis. For example, the Z - Y - Z Euler angles reaches singularity, where α and γ are not uniquely defined, when the rotation about Y is either 0° or 180° (Siciliano & Khatib 2008, p.12.) In robotics, the Euler angles are applicable with coordinate transformations between links as described in the next chapter.

3.3. Homogeneous transformation

General transformation composed of rotation and position becomes complex due to the sum term if multiplying several frames in a chain. Therefore an operator referred to as a homogeneous transformation is formed from the general transformation (Bajd et al. 2013, p.40. p.51). Adding a fourth row of nulls and one to the general transformation does not affect the result. However, the homogeneous transformation matrices can now be multiplied according to matrix calculus while carrying the position vector along. Formula (1) is rearranged:

$$\begin{bmatrix} {}^A\mathbf{S} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} {}^A\mathbf{R}_B & {}^A\mathbf{P}_B \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} {}^B\mathbf{S} \\ \mathbf{1} \end{bmatrix} \quad (6)$$

The position vector is 3x1 and rotation matrix is 3x3. Therefore the 4x4 homogeneous transformation matrix becomes as in Formula (7).

$$\mathbf{T} = \begin{bmatrix} {}^A\mathbf{R}_B & {}^A\mathbf{P}_B \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (7)$$

The added row is not just to create a quadratic matrix: in 3D-graphics or computer vision systems, the 1x3 vector below rotation matrix is used to produce perspective transformations between frames and the scalar below position vector to define scaling. However, these are neglected in basic manipulator geometry and kinematics (Bajd et al. 2013, p.52-55).

The homogeneous transformation matrix allows multiplication in order to solve the position and orientation of the last frame through the entire chain. As previously concluded, vector \mathbf{S}_C in Figure 5 could be expressed at frame {B} with homogenous transformation ${}^B\mathbf{T}_C \times \mathbf{S}_C$ and further forward the \mathbf{S}_B expressed at frame {A} with ${}^A\mathbf{T}_B \times \mathbf{S}_B$. Combining homogeneous transformation matrices yields a direct solution from {C} to {A}:

$$S_A = {}^A_C T S_C \quad (8)$$

$${}^A_C T = {}^B_C T {}^A_B T \quad (9)$$

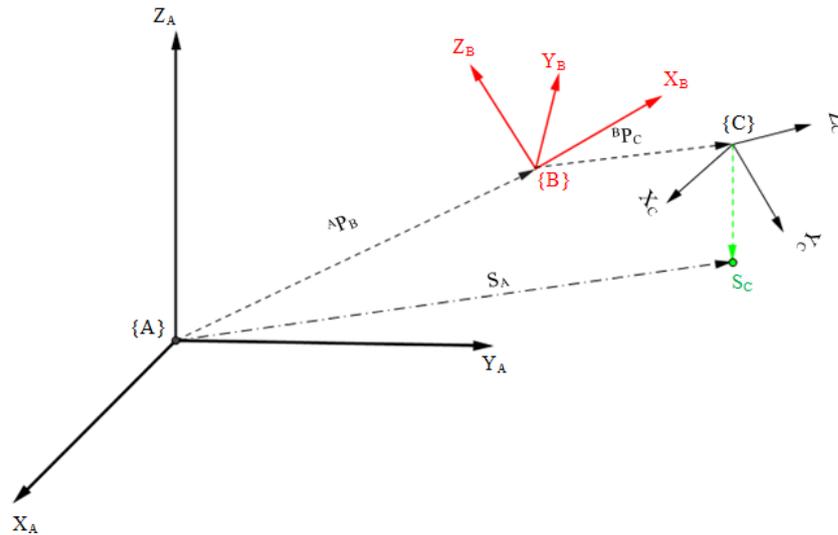


Figure 5. Chain of frames

3.4. Geometric model of the manipulator

The previously introduced methods are applied to robot manipulators. The manipulator is studied as a set of rigid body links connected with revolute or prismatic joints. Links and joints introduce geometric constraints to the relative motion between two links to set up only one degree of freedom. A revolute joint can be imagined as a hinge in real-life applications. It introduces angles variable about a fixed axis. A prismatic joint introduces displacement variables about a fixed axis and is thereby a sliding joint like a hydraulic cylinder (Siciliano & Khatib 2008, p.23.) Other joint types are possible in real systems; helical, cylindrical, spherical or planar. However, these can be modeled as a combination of revolution and prismatic joint properties. For example, a spherical joint is modeled as three perpendicular revolute joints with the same origin.

A system of links and joints can be expressed as a kinematic chain. It can be serial or parallel or a combination of both. The origin of the base frame is fixed in space and is held as a reference coordinate system (Lenarčič et al. 2013 p.61.) Rigid body links and joints introduce a number of constraints between the frames. Six parameters are needed to describe the pose of each rigid body in the chain. Nonetheless, these parameters have relationships between each other, and therefore a set of independent coordinates is created. Independent coordinates for each frame comply with the functions of the rigid body constraints. They are referred to as generalized coordinates that consist of the constraints (Siciliano & Khatib 2008, p.23). A homogeneous transformation matrix is applied to describe frame transformation according to the geometry and the connection between each link. This geometric representation is built using the previous rules and described in the next chapter.

3.4.1. Denavit-Hartenberg parameters

The minimal set of parameters needed to define a manipulator's geometric model is acquired with a convention that Denavit and Hartenberg introduced in 1955. This functions with homogenous transformations with known parameters between each frame. The pose of a rigid body connected by a joint to another rigid body can be defined using four parameters (Bajd et al. 2013, p.60; Siciliano & Khatib 2008, p.23.) The original method was introduced for purely mechanical systems and over time has been modified with numerous conventions to improve the parameters' suitability for robotics and clarify notation. However the original method is still valid in principle and is therefore introduced here.

Modeling is begun by positioning a coordinate frame for each link as illustrated in Figure 5. The pose of the frame attached to the link could be chosen randomly. However, the common convention for obtaining DH-parameters is to position the Z-axis of the frame along the axis of the joint as seen in Figure 6. Link length a_i is then described with a common normal drawn between z_{i-1} and z_i . The X-axis is positioned along this common normal, and therefore the origin of the frame is at the intersection between the joint axis and common normal. The Y-axis is defined using the right hand rule.

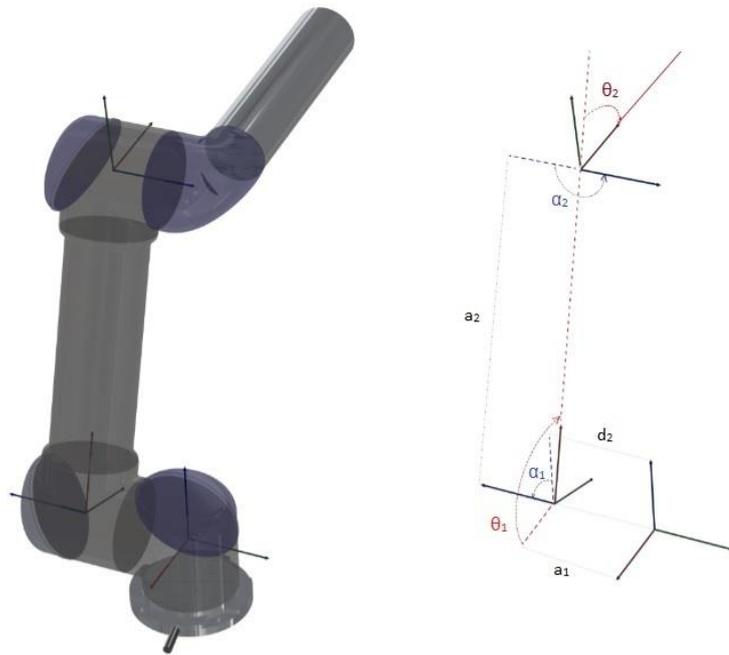


Figure 6. Frame allocation for DH-parameters: CAD model: Courtesy of Universal Robots

Parameter d_i describes the offset between two links and is obtained from the distance between the origins of two successive common normals along z_i . In the case of parallel Z -axes, the common normal is not uniquely defined. The number of non-zero parameters is therefore reduced by positioning the common normal with $d_i = 0$.

In the case of intersecting Z -axes, the X -axis is positioned to the intersection point as perpendicular to the plane that the axes span, and the link length becomes zero. Parameter α_i is the link twist. This is the angle between z_{i-1} and z_i , measured as a counter clockwise positive direction about x_{i-1} . The last parameter, θ_i is the angle between two common normals.

Parameters α_i and a_i describe the link properties and are therefore constants, while d_i and θ_i describe the link connection and one of them is variable depending on the joint type. With a revolute joint, the variable is θ_i and d_i is constant, and vice versa with a prismatic joint (Bajd et al. 2013, p.60). The transformation matrix over each link is the multiplication of intermediate frames acquired from the parameters:

$${}^{i-1}\mathbf{T}_i = \mathbf{Rot}(x, \alpha_i) \mathbf{Trans}(x, a_i) \mathbf{Rot}(z, \theta_i) \mathbf{Trans}(z, d_i) \quad (10)$$

Conventional DH-parameters have a problem with poorly defined common normals between consecutive joint axes if they are nearly parallel. Therefore Hayati and Mirmirani suggested an additional parameter β . This is a rotation about the y_{i-1} axis. The modified method yields link transformation according to Formula (11) (Siciliano & Khatib 2008, p.324).

$${}^{i-1}\mathbf{T}_i = \mathbf{Rot}(y, \beta_i) \mathbf{Rot}(x, \alpha_i) \mathbf{Trans}(x, a_i) \mathbf{Trans}(z, d_i) \mathbf{Rot}(z, \theta_i) \quad (11)$$

Complete transformation through the entire kinematic chain is obtained by multiplying all of the individual link transformations.

3.4.2. Kinematic equations

The chain of rigid body links and joints is considered to end at the tool flange. The actual tool that operates with the workpiece is attached to the tool flange and is termed the *end-effector*. To control the tool pose in the robot work envelope, a tool center point is defined for the tool. A tool center point is an origin of a frame that the operator introduces as a constant translation and rotation from the tool flange.

Kinematic equations are implemented in order to reach the required pose of the end-effector tool frame. The solution to acquire the end-effector pose in relation to the fixed frame is known as the forward kinematics of the manipulator. Principally, the forward kinematics solution is obtained by solving the transformation between the fixed base frame and the end-effector frame using a homogenous transformation over the entire serial-chain manipulator. Inverse kinematics is more complicated: the end-effector pose in the reference frame is known and the joint displacements are solved as a function of the constraints. Forward and inverse kinematics with 4x4 matrix formalism is well suited for basic design and modeling. However, modern robot controllers rely on more computationally efficient algorithms.

4. ROBOT CALIBRATION

The mathematical methods introduced in the previous chapter are composed for the robot controller to enable it to perform the tasks that the user has defined. Modern industrial robots perform complex tasks and the user has to rely on the robot acting as it was programmed. This leads to two common problems. Firstly, the mathematical model used by the controller has to be accurate enough to reach the required end-effector pose. Therefore the model has to be modified to take errors into account. If the solution routine for kinematics is sophisticated enough, the required modifications are acquired with manipulator calibration. Secondly, since the end-effector design is fully task-dependent, the controller has to be programmed with the current tool center point. A tool center point is an origin of a frame that is translated from the tool flange frame origin. When correctly introduced to the robot controller, the pose of this point is manipulated with the best accuracy of the robot. Therefore the tool center point also has to be calibrated.

4.1. Accuracy and repeatability

As concluded, the geometric model of the manipulator leads to a mathematical representation of its kinematics. This mathematical model is embedded into the robot controller. When the operator requests a tool pose in the task space, the controller solves inverse kinematics to actuate each joint respectively at the joint space. This solution is a transformation from task space to joint space. In other words, the pose \mathbf{P} is described in the task space with a vector of position and orientation, and the controller solves the vector of joint displacements $\boldsymbol{\theta}$ as a function of pose \mathbf{P} and geometrical parameters $\boldsymbol{\eta} = (\mathbf{a}, \mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{\beta})$

$$\boldsymbol{\theta} = f(\boldsymbol{\eta}, \mathbf{P}) \quad (12)$$

The parameter vector $\boldsymbol{\eta}$ of the manipulator geometry is not exactly known. For example, if the link length in the real manipulator is different than what was originally designed, because of the incorrect frame positioning, the mathematical model is inaccurate. Small deviations between the model and real system create an error in the end-effector pose. The capability of the manipulator to avoid pose errors is commonly expressed by the repeatability and accuracy of the robot. Repeatability is determined by a robot's ability to

return to a previously recorded pose from another location, while accuracy describes a robot's ability to move to a pose that has not been previously recorded (Mooring et al. 1991, p.5.)

Repeatability is indicated as a radius of a sphere where the origin of the tool frame repeatedly returns with one program and static load conditions (Siciliano & Khatib 2008, p.83). It is expressed more often than accuracy, since it is considerably better. This difference is explainable with Formula (12). When the robot is moved to a completely new pose in the task space, the controller must solve joint displacements in the joint space. Due to the errors in mathematical model, the joint displacements that finally create the pose are not exactly correct. If each link has few small errors in the DH-parameters, the final pose can be several millimeters off from that required. However, if this pose is now saved to the controller as a vector of joint displacements and the robot is moved away and back, it will return to the same position and orientation much more precisely, because the required joint displacement vector is known and the controller does not have to use a mathematical model. The smallest incremental step that the manipulator is able to perform is expressed as resolution, and at even with an ideal model, this is the hardware limit for repeatability.

In the literature, parameter errors are divided into geometric and non-geometric errors. Geometric errors are mostly due to constructional defects, such as an offset in link length and alignment or joint axis orientation errors. These are mostly the result of manufacturing tolerances. Reducing the manufacturing tolerances yields less error; however it would increase the manufacturing costs greatly (Mooring et al. 1991, p.15).

Non-geometric errors are introduced by the dynamic behavior of the manipulator. Flexibility of rigid body links and gear backlash or compliance of the joints are non-geometric errors. Joint angle transducer gain and computer vision camera distortion are parameters that create non-geometric errors. Nevertheless, most manipulator errors are known to be geometric, although what errors will occur and how significant they are is always dependent on the manipulator design (Mooring et al. 1991, p.17).

Accuracy and repeatability affect the programming method most suitable for the robot and task. Traditionally the operator defines each waypoint by jogging the end-effector to the required pose. In this way the operator can visually verify the required poses and trajectories: it is known as the teaching method. The robot program will act within the limits of the robot's repeatability. However, this method is time-consuming and requires a skilled user. The offline programming method was developed to speed up and outsource robot programming. One operator can manage multiple robot cells and does not have to be located next to the robot. In the offline method, the robot program is constructed using computer software and the complete program is uploaded to the robot controller afterwards. Obviously, this method is highly dependent on the accuracy of the robot, as the trajectory waypoints are introduced as task space coordinates. Since the kinematic model is unique for each manipulator, the offline-generated robot program is not directly transferrable between even identical robot cells, and the advance gained from fast programming is lost without proper parameters.

The modern method is to equip the manipulator with sensors that will improve the task performance with feedback. This method allows the program to be constructed offline, since eventually the controller adjusts the final pose with help, for example from computer vision feedback. Ideally the robot functions with loose trajectories and high velocities in areas where accuracy is irrelevant and the feedback improves the accuracy only when it is relevant.

4.2. Calibration of the manipulator

Kinematic calibration of the manipulator can be divided into two categories. In open-loop calibration, the error is defined by external measurement equipment, whereas closed-loop calibration uses physical constraints to seek the deviation between actual and requested end-effector poses. With open-loop calibration, the accuracy of the measurement is dependent on the measurement equipment. One component of the pose vector is measurable with a simple distance sensor; however a stereo camera system is necessary to obtain all six components simultaneously (Siciliano & Khatib 2008. p.325).

Closed-loop calibration utilizes a constraint in the robot work envelope. The end-effector's point contact with an object will define a known position which is then compared to the calculated position. For full pose measurement, the end-effector is rigidly gripped to the ground, thereby fully constrained, and the pose is defined by the joint displacements. Commonly, each manipulator is individually calibrated at the factory before being shipped to the customer. Nevertheless, manipulator calibration is required during use if the manipulator is damaged due to a crash or if the controller, actuators or transducers are replaced. Since each manipulator has typical errors related to the mathematical model in the controller, the robot manufacturers provide technical support for performing suitable calibrations. However, equipment is available on the market allowing the end user to perform manipulator calibrations.

4.3. Tool center point calibration

The controller of a serial manipulator solves equations to move the last frame of the kinematic chain to a requested pose. By default, the last frame is the frame at the tool flange. When the operator attaches a custom tool required for the specific task, the controller must be updated according to the tool properties in order to actuate the tool pose properly. This is done by defining a tool frame as a constant translation and rotation from the tool flange frame. The tool center point, referred to as the TCP, is the point whose relation to the base frame is then solved by the kinematic equations of the controller. The TCP depends on the tool and task type and the operator must decide which point is the most important. Several tool center points or user frames can be defined to the controller. Obviously, only one can be active at any time.

In some cases, the tool center point can be neglected and all the movements will be conducted as if the tool does not exist. In this case, the operator defines all the waypoints and trajectories for the tool flange. Thereby, adjusting the pose of the tool flange produces the correct pose of the tool. This yields problems if the tool is changed, since changing the TCP will not change the trajectories correctly. This is possible, but not recommended, with the teaching method and visual confirmation for smaller tasks; after any modifications the whole path must be taught again.

For simple tools and robot programs, i.e. when the tool is in line with the tool flange and each pose is manually taught to the controller, the entire pose of the tool frame is not relevant and just translation along Z-axis of tool flange frame will yield the proper accuracy for the task. For complex tools and offline programmed trajectories, the offset along the Y- and X-axis and the orientations of the tool frame are also important.

4.3.1. Manual calibration of the tool center point

Tool center point calibration is traditionally done manually by the user. It requires a skilled operator with advanced knowledge about robotics. If the tool geometry is precisely known, the operator is able to input constant translations and rotations directly to the controller. However, in most cases the operator has to define an unknown tool center point by jogging the manipulator into multiple poses in such a way that tool tip contacts a reference point in the manipulator work envelope. The tool center point is fully defined by jogging the manipulator to at least three different poses and calculating or visually determining the translations and rotations. Some robots are supported with built-in TCP calibration procedures, where the user only defines the requested three poses and confirms the point contact. Nevertheless, this requires time, even from a skilled operator. Ideally the tool center point calibration procedure is required only when the tool is mounted to the robot for the first time. However, the tool center point is changed from its initial measure if the tool wears or is unintentionally driven into an obstacle during the program.

4.3.2. Automatic calibration of the tool center point

Various factors can affect the changes of tool center point and these changes can lead to expensive failures. For example, one millimeter offset of the tool center point in a welding gun could ruin the entire weld. Frequently repeated manual calibration of the tool center point increases the down time of the entire factory line dependent on the robot cell. To reduce down time and risk of failure, the robot cell can be equipped with an automatic tool center point calibrator unit that checks and corrects the tool center point frequently, or if the user initiates the calibration sequence after a crash.

The tool center point can be automatically defined using a computer vision system, laser tracking, contact surfaces, distance measurements or various other measurements. As can be concluded, numerous possibilities for conducting calibration are available and the appropriate method should be chosen considering the requirements of the tool itself. If only translations of the tool center point for a simple tool are required, computer vision or a laser tracking system become rather expensive investments for calibration. The common factor in all these methods is that they are used to seek a reference point on the tool and calculate the translations and rotations for that point.

The importance of automatic tool center point calibration was noticed when welding robot cells become popular in industry. In the early 1990s, Henry F. Thorne invented a fast and accurate way to calibrate tool center point translations suitable especially for a GMAW welding gun (Thorne 1993). A similar method was quickly adopted by ABB Robotics, who brought a calibrator unit known as the BullsEye to the market. Generally this device is known as an SCBU, single beam calibrator unit. It uses a light barrier to locate tool edges in a robot coordinate system when the tool is moved across the beam. In this way, the translations of the tool are easy to define for rotation symmetric tools that have a tool center point in the centerline. The method uses the robot itself as a measuring device and therefore the best accuracy is dependent on robot repeatability. However, the method requires only a light barrier as a measurement device and can be fitted to any robot with one free digital input. The commercial model is equipped with a control unit or support from the robot controller. Figure 7 illustrates the ABB BullsEye Calibrator.



Figure 7. Automatic TCP calibration Units. Left: BullsEye (ABB). Right: RotoLab (Wiest AG, 2014)

Wiest AG's RotoLab calibrator is also shown in Figure 7: this unit uses 2D coordinate measuring by the bisection method (Wiest AG 2014). Leoni AG provides a similar automatic calibrator unit known as the Advintec, shown in Figure 8. In comparison to Bullseye, Advintec uses two light barriers that cross each other. In this way the orientation of the tool center point is also acquired for rotational symmetrical tools and the entire procedure is marketed as faster (Leoni Protec Cable Systems GmbH 2014).



Figure 8. Advintec TCP-6D calibrator (Leoni Protec Cable Systems GmbH 2014)

5. THE STUDIED CALIBRATION METHOD

The tool center point for Barkhausen noise sensor can be defined at two different positions. The optimal tool center point that is in contact with the specimen to obtain an optimal measurement value is dependent on the inner properties of the sensor. These inner properties are affected by the size, shape and positioning of the electronics that excite and measure the Barkhausen noise. Magnetization frequency, current, voltage, etc., affect the optimal position of the measurement. Therefore, calibration regarding this point would require a very advanced calibration system. However, the geometrical tool center point is considered as a frame at the tip of the sensor, as shown in Figure 9. This study focused on determining the geometrical tool center points for a Barkhausen noise sensor that is attached to the fixture. After the physical tool center point is defined at the tip of the tool, a constant translation can be used to move it closer to the true TCP, which is in the area shown in red in Figure 9.

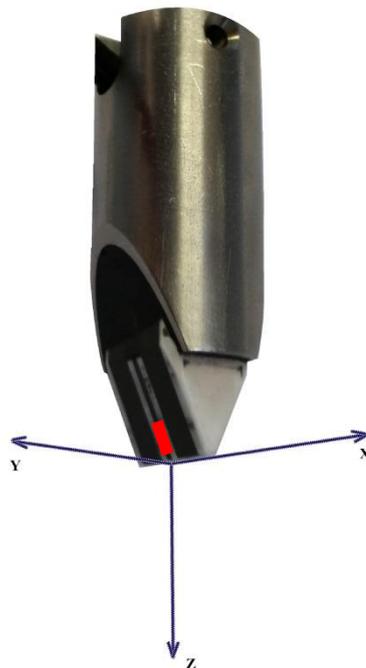


Figure 9. Geometric tool center point frame in a tip of the Barkhausen noise sensor. The red line indicates the true and optimal measuring area

5.1. Calibration setup

The Barkhausen noise sensor is attached to a Universal Robots UR5 manipulator. The Barkhausen sensor is equipped with a special fixture to connect the sensor to the tool flange of a robot in such a way that the sensor head is able to self-align. The calibration unit has two Omron E3T through-beam light barriers that are directly connected to the robot digital inputs. These light barriers are used to define the edges of the tool. The robot controller records the coordinates of the tool flange in the robot coordinate system when either of the light barriers indicates that the sensor is blocking the beam. The translations and rotations of the geometrical TCP are solved with these edge coordinates.

5.2. Universal robots

Universal Robots was the first company to bring lightweight robot arms to the market. Robot researchers from the University of Southern Denmark noticed that industrial robots are heavy and complex to use since they are designed by engineers for engineers. The programming and even supervision of the robots always required an experienced operator. Heavy robots working with high inertias also require safety shields to protect the operator. The cost of the safety equipment increases the total investment considerably. Their conclusion was that no suitable robot models are available for small and medium sized companies. Universal Robots A/S was founded in 2003 and in 2009 the first models were sold to distributors (Universal Robots 2014). Market growth has been remarkable, since the design of the Universal Robots allows small and medium sized companies to increase their automation level at moderately low costs. The reasonable expenses are thanks to the innovative structure of the manipulator and controller.

For small companies, the major benefit is the visual and easy to use graphical user interface that does not require an advanced robotics engineer for programming. Basic tasks can be taught to the robot in minutes and the user can create simple programs without writing even a single line of conventional programming script.

The lightweight structure of aluminum links and force-controlled motors gives low inertia with fast built-in safety procedures. Overall investment costs are reduced, since the robot is

able to work in interaction with a human without excessive protective safety equipment. This is due to the control system that conforms to the standard for machinery safety, ISO 13849-1 performance level D. The Danish Technological Institute has also certified the robot complies with Article 5.10.5 of ISO 10218-1:2006. According to this standard, the robot is suitable to operate as a collaborative robot and therefore does not require safety guards between the robot and the operator. Controller software monitors the motor's dynamic load at a high kernel level to prevent forces over 150 N and torques that significantly deviate from those expected (Universal Robots, 2013).

Universal Robots uses Kollmorgen's customized frameless direct drive motors to rotate the joints. These motors are directly embedded into the robot's frame and connected to harmonic reduction gear on each axis. The hollow shaft harmonic gears are produced by Harmonic Gear AG. Attaching the motor directly to the frame and gear reduces the required bearings and gearing, thereby minimizing the overall weight and gear backlash. The motors provide feedback to the controller to determine the force for each joint. The force at the tool flange is acquired with a precision of +/- 10 Newtons.

Currently two robot models are available for different payloads. The smaller robot, UR5, is shown in Figure 10 and is able to carry 5 kilogram payloads. The larger, UR10, is for 10 kilogram payloads. This study was implemented on the UR5, since it is enough to operate a Barkhausen noise sensor. The overall weight of the UR5 is 18.4 kilograms and it has a reach of 850mm. Six degrees of freedom are acquired with six rotating joints and the robot is capable of +/- 0.1mm repeatability, which is less than in most robots. Nonetheless this is more than enough for most basic tasks and it can be considered exceptional when evaluating the power density of these robots.



Figure 100. UR5 manipulator and controller. Courtesy of Universal Robots.

The controller unit seen in Figure 10 has a compact industrial computer inside which uses a Linux-based operating system. The robot is mainly programmed using the Polyscope graphical user interface (GUI) operated with a touch-sensitive screen connected to the controller with a 5-meter cable. The Polyscope GUI provides a fast and easy teaching method to set the waypoints for the robot and has built-in task generators to quickly set up pick and place or stacking programs. For complex tasks, it is possible to create programs using the *URscript* programming language which contains all the needed controller commands, basic mathematical functions and data communication commands. In this study, the main program was constructed using the Polyscope GUI and only the advanced tasks were added to the program using URscript commands. The controller provides a built-in 24-Volt power source for external equipment connected to the I/O. The power

source also supports safeguarding connections and for all devices the maximum current is 1.2 Amperes. In addition to multiple emergency and safeguard inputs, earlier models provided 8/8 digital and 2/2 analog I/O connections. The third generation model released in 2014 is equipped with twice the connections.

An interesting property of Universal Robots is the built-in possibility to use the robot in force control mode. In this mode, the tool center point is not exactly controlled, however the forces along the tool center point axes or tool forces in respect to the base frame axes are controlled. In addition, the operator is able to define user coordinate frames to the work envelope and control the force at the TCP in relation to these frames.

5.3. Barkhausen noise sensor

As described in Chapter 2, Barkhausen noise is measured by a coil which is induced by abrupt changes in the magnetization of the specimen. Therefore, the sensor head itself consists of two coils: one to create the changing magnetic field for the magnetization and a second one to measure the Barkhausen noise. The coils are inside the measuring head, which can be shaped to match the specimen. Figure 9 illustrates a sensor that is suitable for measuring flat surfaces. For complex surfaces, a sensor with curved head can be selected.

The sensor head is attached to a fixture that has rotational joints, since successful measurement requires the sensor to be properly aligned with the surface of the specimen. Rotational joints in the fixture allow the tool to self-align with the surface and the spring load provides constant contact even though the robot trajectory varies. Figure 11 illustrates the fixture used in this study. The fixture is connected to the robot tool flange with adaptors. In addition to the Barkhausen sensor, an inductive sensor is also rigidly attached to the fixture as seen in Figure 11. This sensor is used to define the start position of the measurement. For example, the position of a tooth in a gear is found using the inductive sensor.

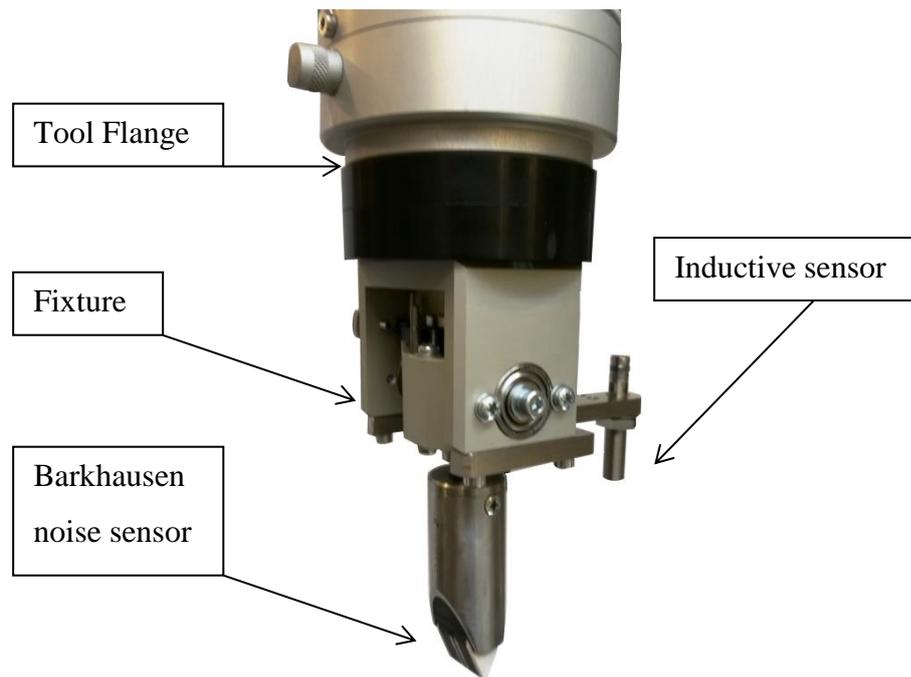


Figure 111. The fixture and the sensor attached to the tool flange.

5.4. Calibrator equipment

Photoelectric switches or *light barriers* are used to detect the edge of the tool. Light barriers should be selected as precisely as possible since the resolution of the light barrier affects the accuracy of the calibration. Omron E3T -photoelectric switches were selected for this study. E3T-CT14 is a PNP switching, through-beam sensor that operates with a 630nm wavelength red light emitting diode. The response time is less than 0.5milliseconds and <0.1mm edge detection accuracy is gained with 0.5mm slits that suppress the spot of the beam. This accuracy is suitable considering that the robot repeatability is +/- 0.1mm.

The photoelectric switches are mounted on a compact steel frame shown in Figure 12. The frame is designed to set the beams properly in respect to each other and provide stable support to position the entire calibrator to the robot work envelope. The model for the frame was designed according to the requirements of UR5's tool flange and a specific Barkhausen sensor shown in Figure 11. The shape of the frame was cut with a laser at the University of Lappeenranta. The distance between the transmitter and receiver is designed to fit the tool flange of the UR5 between them. This is important because the beam position in the robot coordinate frame's Z-axis is determined at the first installation by lowering the

tool flange across the beams. For larger robots, this distance should be increased to fit the flange. In addition, the distance between the beams is uniquely defined by the height of the measuring head in the specific Barkhausen sensor. Only the sensor head is calibrated, and due to manufacturing tolerances, the orientation of the head might deviate from the orientation of the sensor frame. Geometric tolerances for the frame are set tight, since the beams should line up parallel with X - Y -plane of the robot coordinate system and the alignment of the transmitter and receiver is crucial for positioning the centerline of the beam in the middle of the spot. Incorrect alignment would introduce an error into the measuring accuracy.



Figure 12. Assembled calibrator

5.5. Installation in the robot work envelope

The calibrator is mounted in the robot work envelope according to Figure 13. The surface where the calibrator is mounted should be parallel with the surface of the robot base plate. Thereby the beams will align with the robot coordinate system. If the beam is in a different orientation than the plane of the robot coordinate system, the positioning error along the X -axis due to bad robot repeatability would create minor errors in the measurement. These are avoided by careful positioning, and the optimal alignment for beams is parallel with the robot coordinate frame's X - Y -plane and perpendicular to its Y -axis.

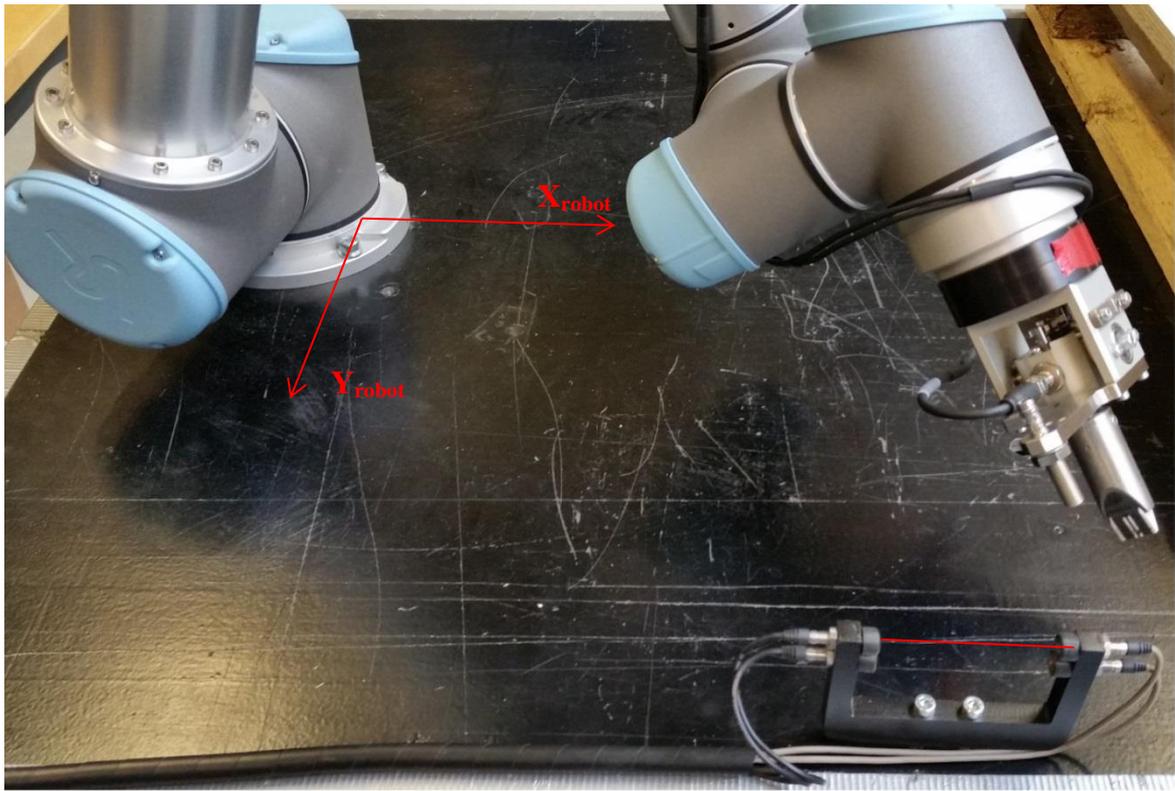


Figure 123. Positioning of the calibrator unit in respect to the robot coordinate frame

The calibration procedure requires that the position of both beams is known in the robot coordinate frame's Z -axis. Therefore, when the calibrator is installed in the robot work envelope, the user has to dismantle the tool from the tool flange, position the flange between the frame and run the initialization program. This program is a set of movement commands to evaluate the beam positioning and orientation with the tool flange. The flowchart in Figure 14 illustrates the procedure of the initialization program.

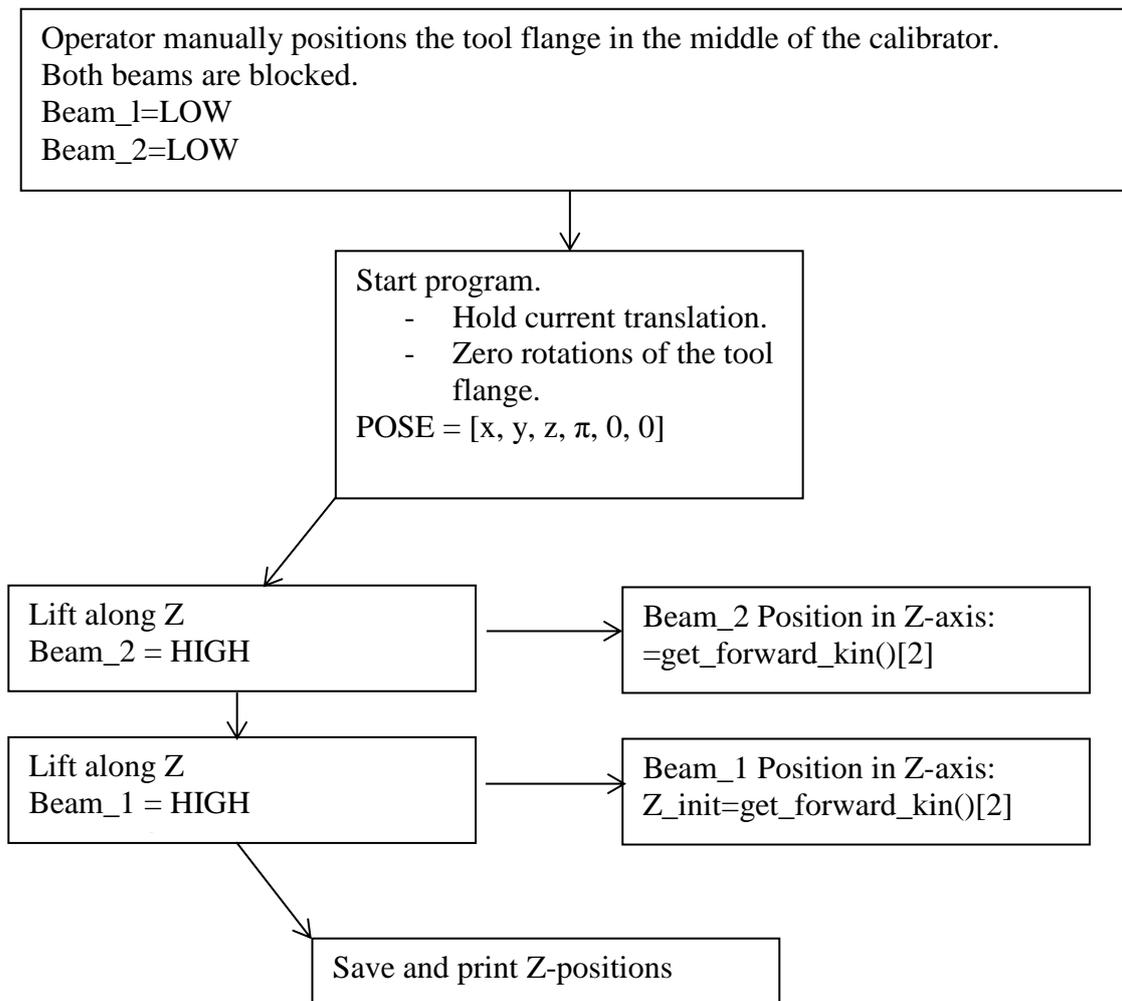


Figure 14. Flowchart of the initialization procedure

5.6. Calibration procedure

TCP calibration is conducted by measuring the Barkhausen sensor in different poses. These measurements are acquired by detecting the edges of the sensor when it is moved across the light barriers. Figure 15 illustrates the procedure of calibration for 5 degrees of freedom as a flow chart.

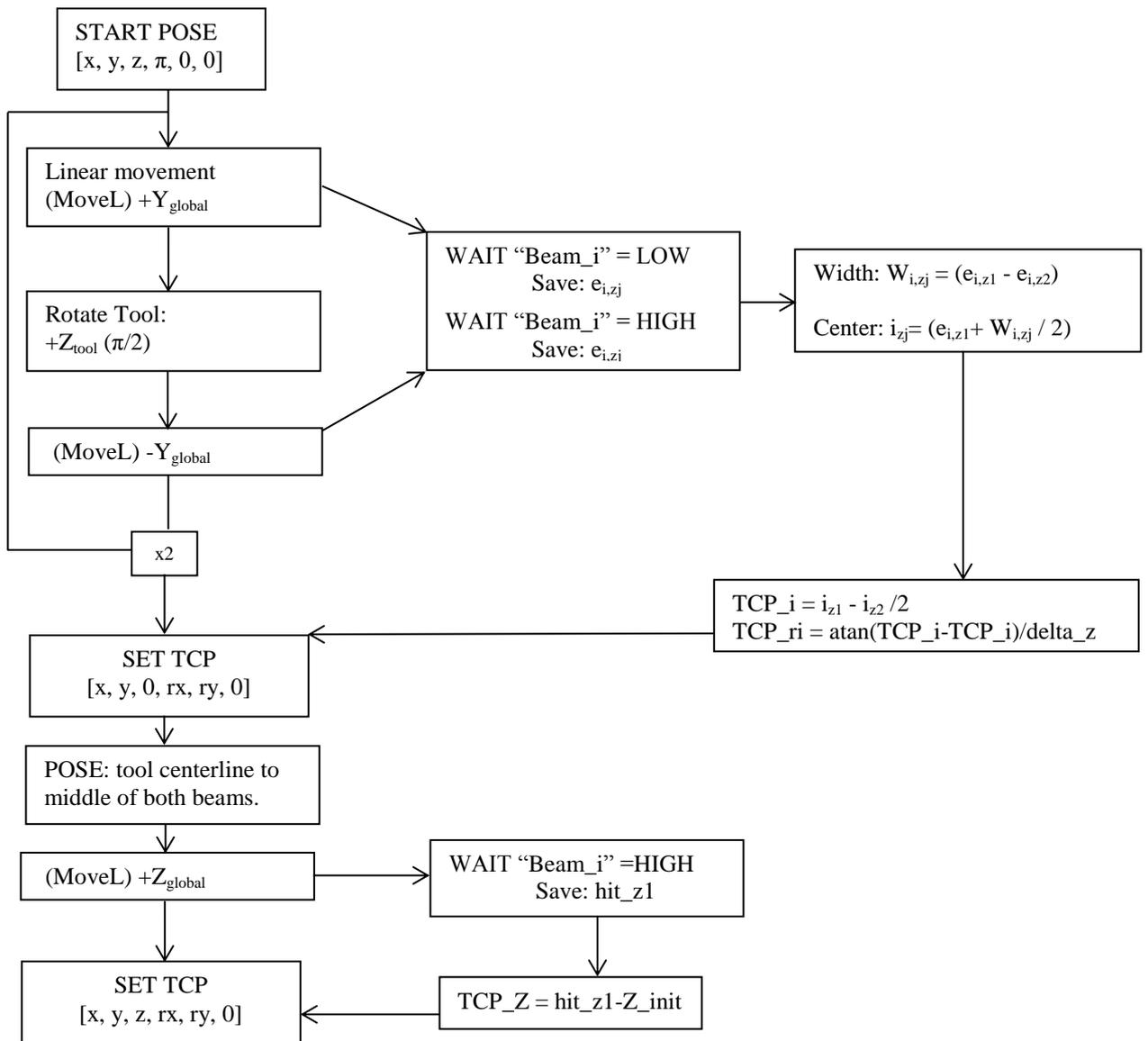


Figure 15. Flowchart of the calibration procedure in 5 degrees of freedom

The robot controller digital input receives a *high*-signal from the photoelectric switch when the tool blocks the beam. This is the trigger to save the edge position at the robot coordinate frame to a variable $e_{i,zj}$. In this notation, i is the measured axis, z is the beam position (upper or lower beam) and j is the tool edge number (1 or 2). When the sensor is moved further, eventually the *low*-signal triggers the save for the second edge. Pose $e_{i,zj}$ is saved with the URscript command `get_forward_kin()` which creates a spatial pose vector describing the current TCP pose in the robot coordinate frame.

$$e_{i,zj} = [x \ y \ z \ rx \ ry \ rz]^T$$

If the calibrator is mounted perpendicular to the robot coordinate frame's Y-axis as shown in Figure 13, the movement of the entire calibration procedure occurs along the Y-axis. Measurements are conducted along this axis and the measured axis in the tool frame is changed by rotating wrist joint six through 90 degrees as described in the flowchart in Figure 15. The position of the edge in the Y-axis is element 1, since the numbering of the vector's elements in URscript begins at zero. Therefore the variable is defined in URscript as $e_{i,zj}[1]$.

At the beginning of the calibration procedure, the robot is programmed to *zero* the tool flange orientations with the *start pose*. This means that the initial TCP is considered to be only a translation along the tool flange's Z-axis and the tool flange is rotated to be parallel with the robot coordinate frame's X-Y-plane. As seen in Figure 15, the rotation about the X-axis is π , while the others are zero; this indicates that the tool is pointed against the X-Y-plane from up. Following this, when the tool is moved along one axis of the robot coordinate frame, the width $W_{i,zj}$ of the tool is calculated with the edge position variables.

$$W_{i,zj} = \mathit{abs}(e_{i,z_1} - e_{i,z_2})$$

For example, as shown in Figure 16, the width of the tool in its both planar axes is measured by the upper beam and calculated as:

$$W_{X,up1} = \mathit{abs}(e_{X,up_1} - e_{X,up_2})$$

$$W_{Y,up1} = \mathit{abs}(e_{Y,up_1} - e_{Y,up_2})$$

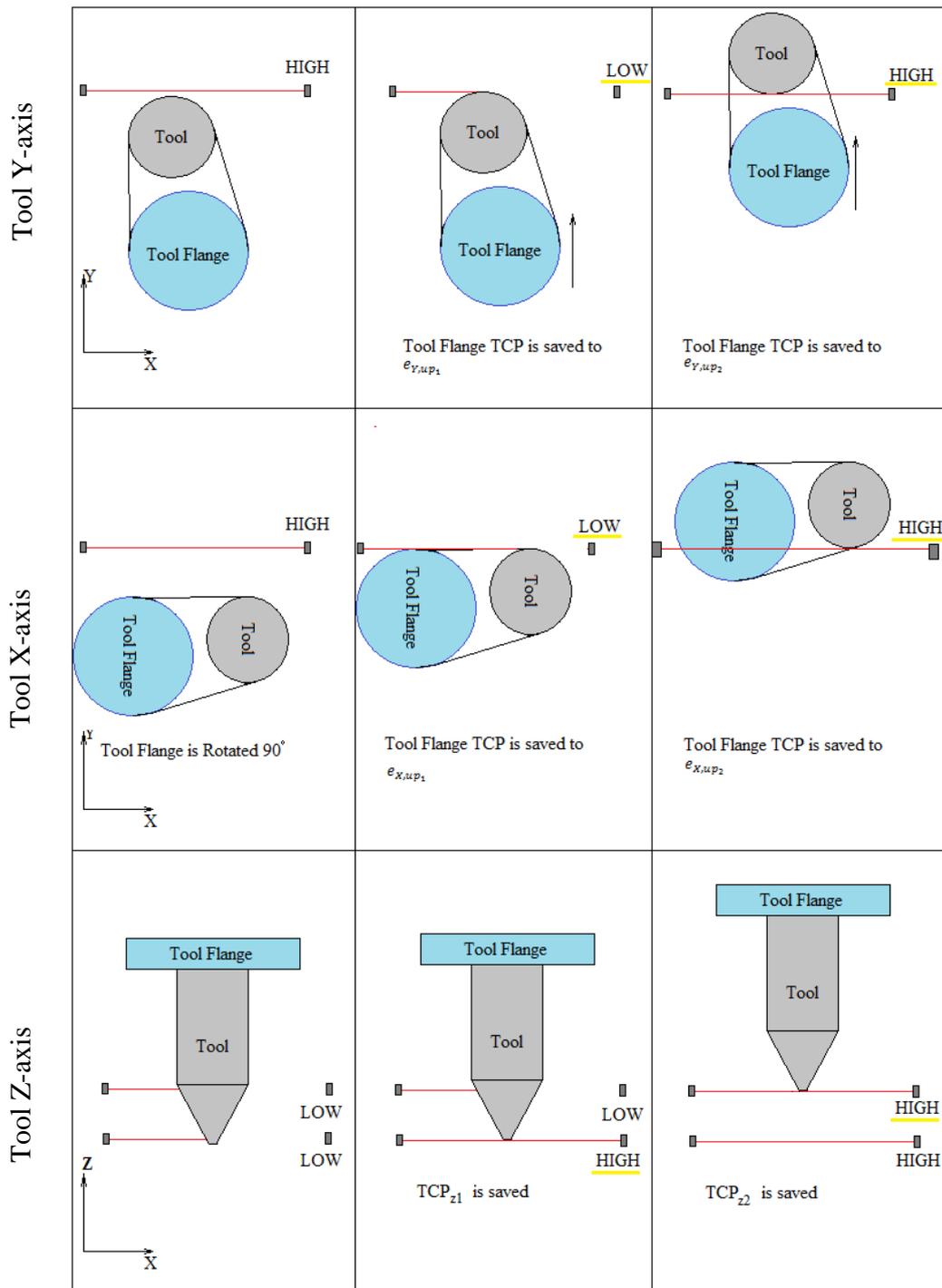


Figure 16. Principle of tool edge detection

The center point position of the tool flange in axis i in this beam position is now determined in the robot coordinate system by adding the half of the width to the first edge position.

$$\mathbf{i}_{z_j} = \mathbf{e}_{i,z_1} + \frac{W_{i,z_j}}{2}$$

Translation of the tool from the flange is over-exaggerated in Figure 16. This is to clarify that the measurement from one direction implies only the center point of the *tool flange* in the robot coordinate system, since the robot controller saves the forward kinematics with zero translations for the tool. Therefore the tool needs to be measured from the opposite direction to define its true offset f .

Therefore, joint six is first rotated 90 degrees to measure the perpendicular axis of the sensor using the same method. Even though the tool is once again moved along the same axis in the robot coordinate system, the tool frame is now realigned and the measured axis in this frame is changed. A 90-degree rotation of the joint six is conducted three times to measure the X - and Y -widths and center points in the full circle of the tool, thereby collecting opposite directions for both axes. Overall, eight center point parameters \mathbf{i}_{z_j} are collected since each axis is measured in two opposite orientations by two beams that are positioned along the robot coordinate system's Z -axis.

$$\mathbf{i}_{z_j} = \{X_{up_1} X_{up_2} X_{down_1} X_{down_2} Y_{up_1} Y_{up_2} Y_{down_1} Y_{down_2}\}$$

The translation of the center point along a specific axis is determined by defining the offset between two center points. Since the center points are defined in the robot coordinate system, the translations are adopted with a correct sign. The tool center point translation in axis i and beam j is then defined using:

$$TCP_{i,j} = (\mathbf{i}_{z_1} - \mathbf{i}_{z_2})/2$$

Using two beams also allows calculation of the orientations (TCP_{ri}) of the tool. Orientation about one axis is defined as an arctangent of the offset for the second axis and the distance between the beams Δ .

$$TCP_{ri} = \text{atan}\left(\frac{TCP_{i,j}}{\Delta}\right)$$

Since the translations and orientations about the X - and Y -axes are defined, they are introduced as a new tool center point for the robot controller. The exact translation of the TCP Z -axis is still unknown. Therefore the tool is controlled to an upright position with a new TCP and the center line of the tool is now therefore parallel with the robot coordinate system's Z -axis. The center line is controlled exactly to the middle of both beams. The tool is lifted along the Z -axis of the robot coordinate system, the low-signal from the lower beam reveals the actual translation and the TCP is defined for 5 degrees of freedom.

Since the Barkhausen noise sensor's geometry is not ideal, the procedure is repeated with the new 5D TCP in the robot controller. During the second sequence, the previously defined TCP is controlled closer than 1mm from the lower beam. Therefore the measurements occur in the vicinity of the actual geometrical endpoint. After the second sequence, the orientation about the Z -axis is also defined. The orientation is acquired by measuring the width W of the sensor's rectangular head in two known rotations about the Z -axis by angle β . The method is introduced in figure 17 and is applicable for defining rotations if the tool has different widths in the X - and Y -axes.

$$TCP_{rz} = \frac{\beta}{W_1 + W_2} (W_1 - W_2)$$

Calibration for rotation about the Z -axis is conducted last, since this measurement requires that all the other translations and rotations have already been introduced to the controller.

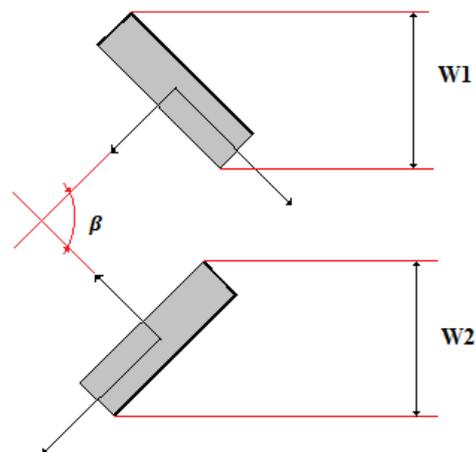


Figure 17. Principle of definition for rotation about the Z -axis

6. QUALITATIVE MEASUREMENTS

Three different data acquisition methods were used to study the suitability of the previously introduced calibration procedure with the Barkhausen noise measurements. The first two measurements were conducted to evaluate the repeatability of the calibration method and seek optimal parameters for velocity and acceleration. The third measurement was initiated to study the actual Barkhausen noise signal analysis with a calibrated tool center point.

In the first set of measurements, the velocity and acceleration were gradually increased to study the maximum values that would affect the calibration accuracy. Calibration for the Barkhausen noise sensor was performed repeatedly and after each calibration round, the measured TCP was exported to a spreadsheet. The velocity and acceleration of the robot was increased in 11 steps from 25mm/s and 120mm/s² to 155mm/s and 744mm/s². The deviation of the TCP compared to the previous result was the key factor to determine maximum acceleration.

The second set of measurements were to evaluate the repeatability with previously defined maximum acceleration. The sensor was calibrated 12 times without changing any parameters. Velocity was kept at 65mm/s and acceleration was 300mm/s². To compare the results, another sample of 20 measurements was conducted with a velocity of 50mm/s and acceleration of 150mm/s. After each calibration, the measured TCP was exported to a spreadsheet and the deviation of the results calculated individually for each degree of freedom.

Table 1. Measurement samples for repeatability

Sample number:	1.1	1.2	1.3
Sample size:	11	12	20
Description of measurement:	Velocity and acceleration increase in steps	Velocity: 65mm/s Acceleration: 300mm/s	Velocity: 50mm/s Acceleration: 150mm/s

The third evaluation was done with actual Barkhausen noise measurements and the procedure is illustrated in Figure 19. The robot was programmed to move the sensor head in contact with a helical gear tooth as in normal usage. The results of this measurement were acquired using the Stresstech Rollscan 300 Barkhausen Analyzer and transferred to ViewScan software version 4.0.0. In this measurement, the TCP's path along the tooth's face was defined by the geometrical tool center point waypoints. Reference values were obtained by moving the sensor head correctly through the waypoints with a calibrated TCP referred as $TCP_{initial}$. After these measurements were repeated three times, an error was introduced. The TCP was translated and rotated by adjusting the joints of the fixture to another position, creating a TCP_{error} . Three different error orientations were introduced:

- Sample 2.1: Translation along Y- and rotation about X-axis as shown in Figure 18.
- Sample 2.2: Translation along Y-axis and minor rotation about X-axis and Z-axis.
- Sample 2.3: Translation along Y-axis, minor rotation about X-axis and large rotation about Z-axis.

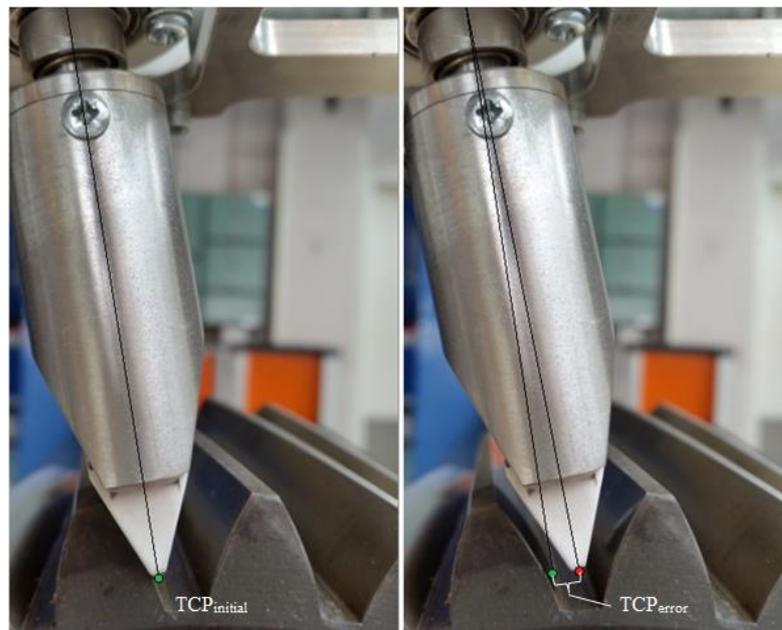


Figure 18. Created error in Sample 2.1: Translation about Y-axis and rotation about X-axis

Barkhausen measurements were repeated three times with the incorrect TCP to validate the data about the wrong path. After these measurements, the TCP was calibrated again to move it back to the path and measurements were repeated three times. All measurements were transferred to the PC to evaluate the differences in the signal quality between the first and second calibrations.

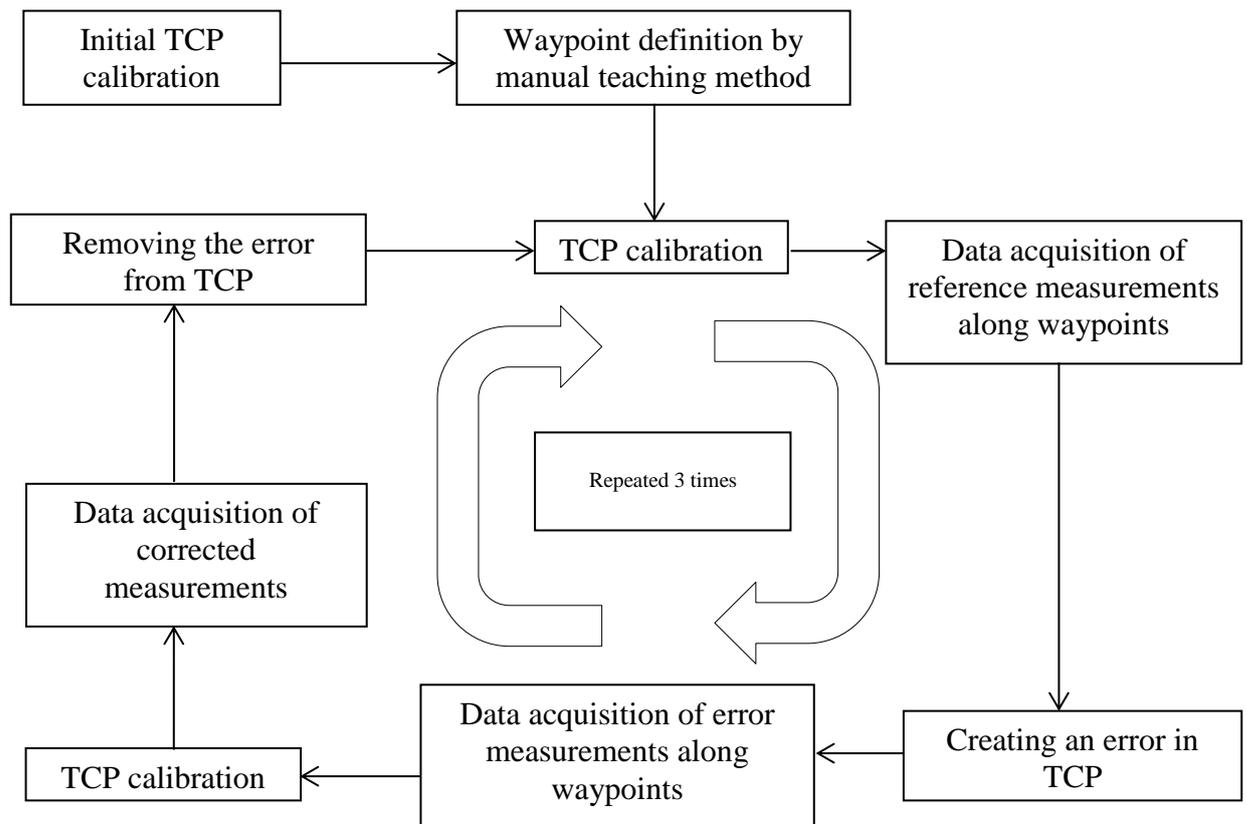


Figure 19. Procedure for the evaluation of recalibration

7. RESULTS

7.1. Maximum acceleration

As shown in Figure 20, the increase in acceleration rapidly affects the calibration accuracy, especially at the *Y*-axis of the tool frame. Deviation in the chart is calculated as the difference between initial TCP translation and measured TCP translation. Initial TCP translation is calibrated with a velocity of 25mm/s and acceleration of 120mm/s². These values are 10% from the default values that the robot uses in order to comply with safety regulations. According to the manufacturer, the robot's repeatability is +/- 0.1mm. Therefore deviations over 0.2 are not tolerable and are considered to be the result of the calibration method itself instead of robot repeatability.

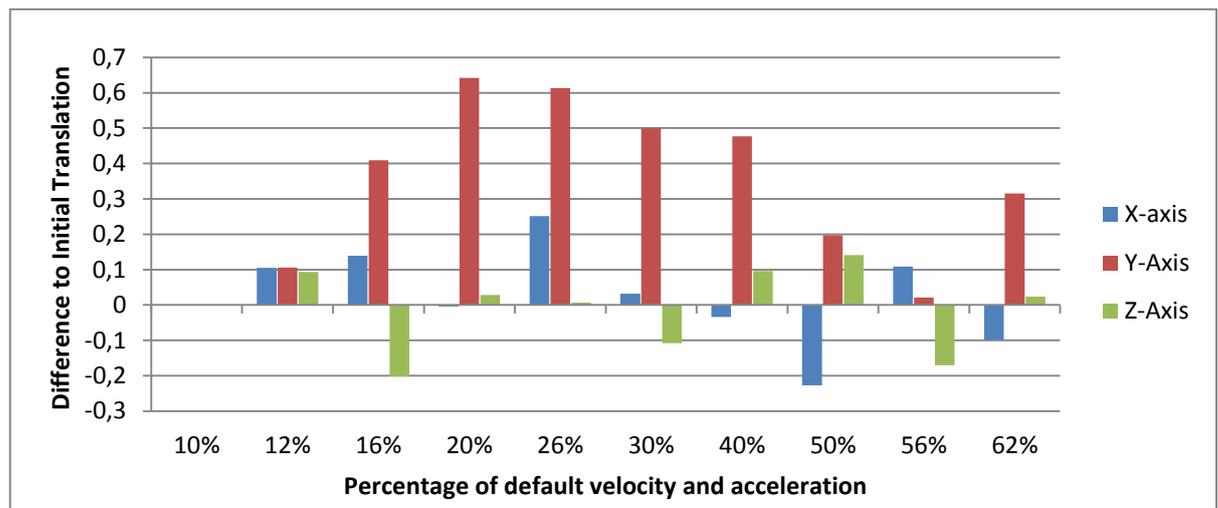


Figure 20. A translation of the TCP as acceleration is increased in Sample 1.1

When the acceleration is 16% from the default, or 192mm/s², the translation of the tool center point along the *Y*-axis of the tool is displaced 0.41mm in comparison to the initial calibration value. Figure 20 also shows that the displacement is high until the acceleration is increased and over 40% acceleration is initiated. Although the *Y*-axis displacement is reduced to 0.2mm after 40%, the *X*-axis displacement is increased to the negative side >-0.2mm. Displacements of translations along *X*- and *Z*-axes are inside the robot's repeatability when operating with acceleration below 20% of the default.

7.2. Repeatability of the calibration method

Sample 1.2. was conducted with constant acceleration and velocity in order to study the repeatability of the calibration method. Table 2 introduces the key parameters for the translations of the TCP frame along each axis after twelve repeated measurements. In this sample, the velocity was 65mm/s and acceleration was 300mm/s². Values were selected from previous measurements.

Table 2. Key parameters from the repeatability measurements in Sample 1.2.

Translation	Max-Min (mm)	Mean (mm)	Standard Deviation (mm)
TCP_x	0.51	-0.396	0.172
TCP_y	0.39	0.819	0.144
TCP_z	0.72	139.894	0.264

The Max-Min values describe the largest difference between individual measurements in the sample. For the worst case scenario, it is shown that the difference between maximum and minimum values in the Z-axis is 0.72mm. The histogram in Figure 21 illustrates the cause of the large standard deviation in TCP_z translation. Dense frequency is focused on the area of 140.0mm translation. However, three measurements are 139.5mm, which is not explainable with robot repeatability. Also the X- and Y-axes have a Max-Min difference over 0.39mm.

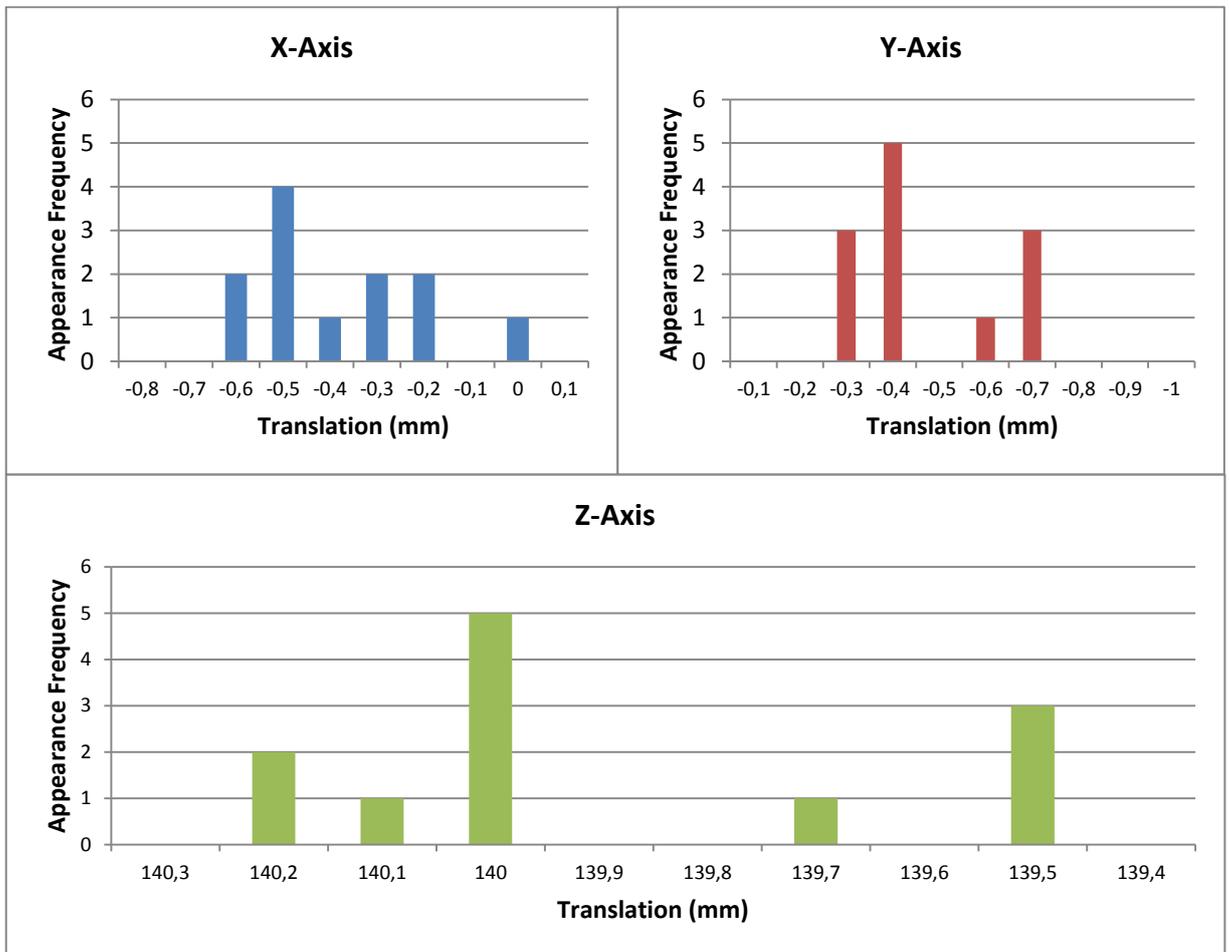


Figure 21. Histograms of the TCP translations in Sample 1.2 (X-axis: Measured Translation, Y-Axis: Number of measures in the sample)

A second sample was conducted to evaluate the repeatability with lower velocity and acceleration. For this sample, the velocity was set to 50mm/s and the acceleration to 150mm/s². The sample size is 20.

Table 3 verifies that reduction of acceleration increases the repeatability for this calibration method. The Max-Min difference between individual measurements is reduced significantly and becomes closer to the robot repeatability. The standard deviation for each axis is ~0.1mm, which is acceptable.

Table 3. Key parameters from the repeatability measurements in Sample 1.3

Translation	Max-Min (mm)	Mean (mm)	Standard Deviation
TCP _x	0.31	-0.031	0.083
TCP _y	0.30	1.104	0.116
TCP _z	0.31	140.034	0.092

The histograms for the second sample are shown in Figure 22. The X- and Z-axis measurements are highly concentrated in the area inside of 0.2mm, although the measurements of translation along Y-axis are spread wider and unevenly without a clear distribution curve.

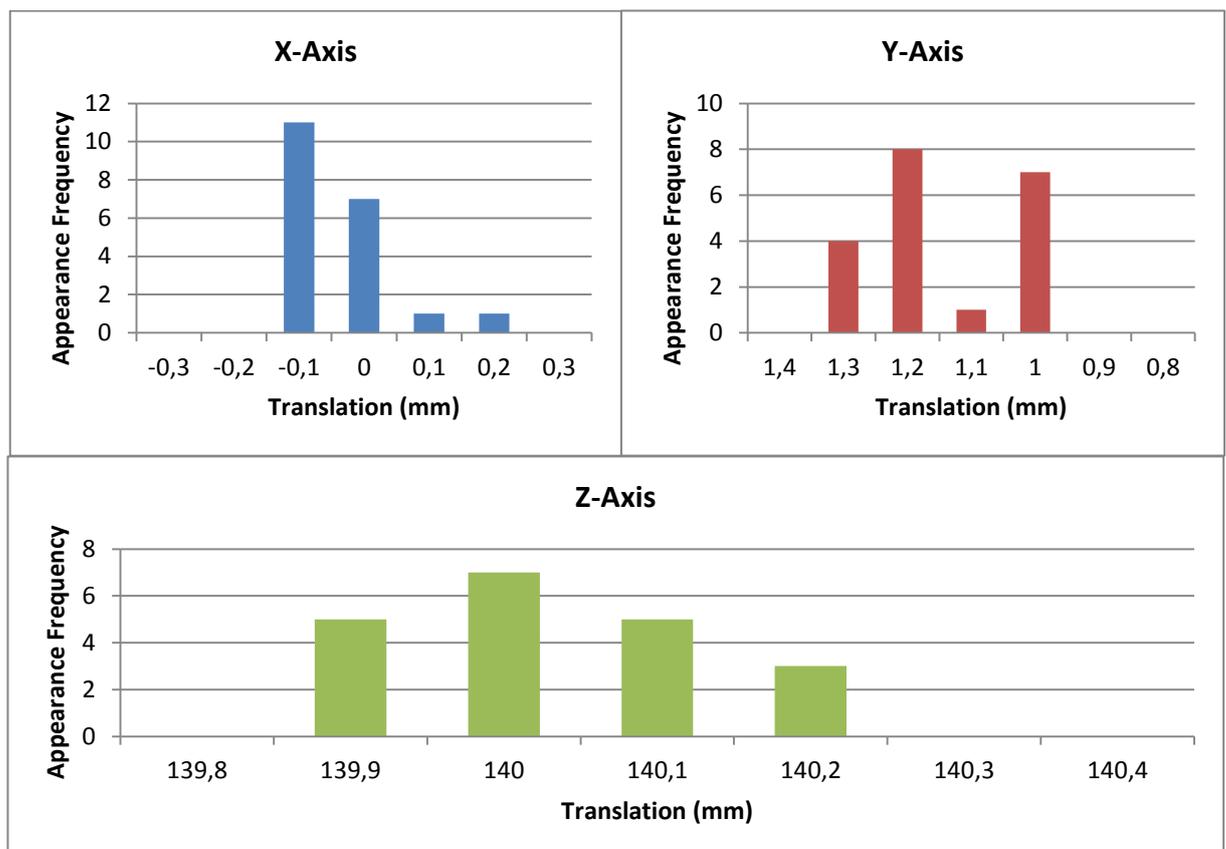


Figure 22. Histograms of the TCP translations in Sample 1.3 (X-axis: Measured Translation, Y-axis: Number of measures in the sample)

7.3. Barkhausen measurements after calibration

The next measurement set was implemented to evaluate the calibration accuracy with an actual Barkhausen noise sensor. In this set, the waypoints along the face of the gear's tooth were taught to the controller with calibrated TCP. Barkhausen noise measurements were repeated three times with this initial TCP and then the sensor was misaligned to simulate an error due to a crash. After recalibration, the Barkhausen noise measurements were repeated again three times to study the difference between the signals acquired with the initial TCP and recalibrated TCP. Table 4 introduces the Barkhausen noise measurement results for each sample and each set of three repeated measurements. Values are obtained from a Rollscan 300 Barkhausen noise analyzer and Viewscan software.

The average value analyzed from the signal for TCP_{initial} and $TCP_{\text{recalibrated}}$ in all samples has a deviation of less than 0.07. Table 4 also introduces the average and deviation of the $TCP_{\text{misaligned}}$. Results with the misaligned TCP are only introduced to prove that the sensor was not properly located at the waypoints during the sample. Values near 10 are actually from the background noise in the laboratory and therefore indicate that the sensor is not acquiring any signal from the specimen.

The combined results for measurements with TCP_{initial} and $TCP_{\text{recalibrated}}$ are listed in the last row of Table 4. This row holds the key parameters to evaluate the accuracy of the calibration method. In the first sample, the average value of the analyzed Barkhausen noise increases 0.17 units from the TCP_{initial} after recalibration. The combined deviation with these six measurements is 0.20. In Sample 2, the average value is reduced 0.43 units after recalibration and the deviation is 0.48. In the third sample, the average value after calibration is very close to the initial value, and the combined deviation is only 0.1 units.

Table 4. Average and standard deviation of values from the measured Barkhausen noise (unitless quantity).

	Sample 2.1		Sample 2.2		Sample 2.3	
	<i>Average</i>	<i>Deviation</i>	<i>Average</i>	<i>Deviation</i>	<i>Average</i>	<i>Deviation</i>
TCP_{initial}	23.88	0.06	23.99	0.04	23.59	0.07
TCP_{misaligned}	12.14	0.16	10.46	0.15	10.67	0.01
TCP_{recalibrated}	24.22	0.07	23.12	0.05	23.74	0.01
Combined	24.05	0.20	23.56	0.48	23.66	0.10

In Sample 2.1, the average of TCP_{misaligned} measurements is 12.14. This is explainable with Figure 23: the TCP is close to the specimen at the beginning of the measurements. Nevertheless, the signal is lost when the sensor has moved towards the second waypoint. The plot is obtained from the Viewscan software.

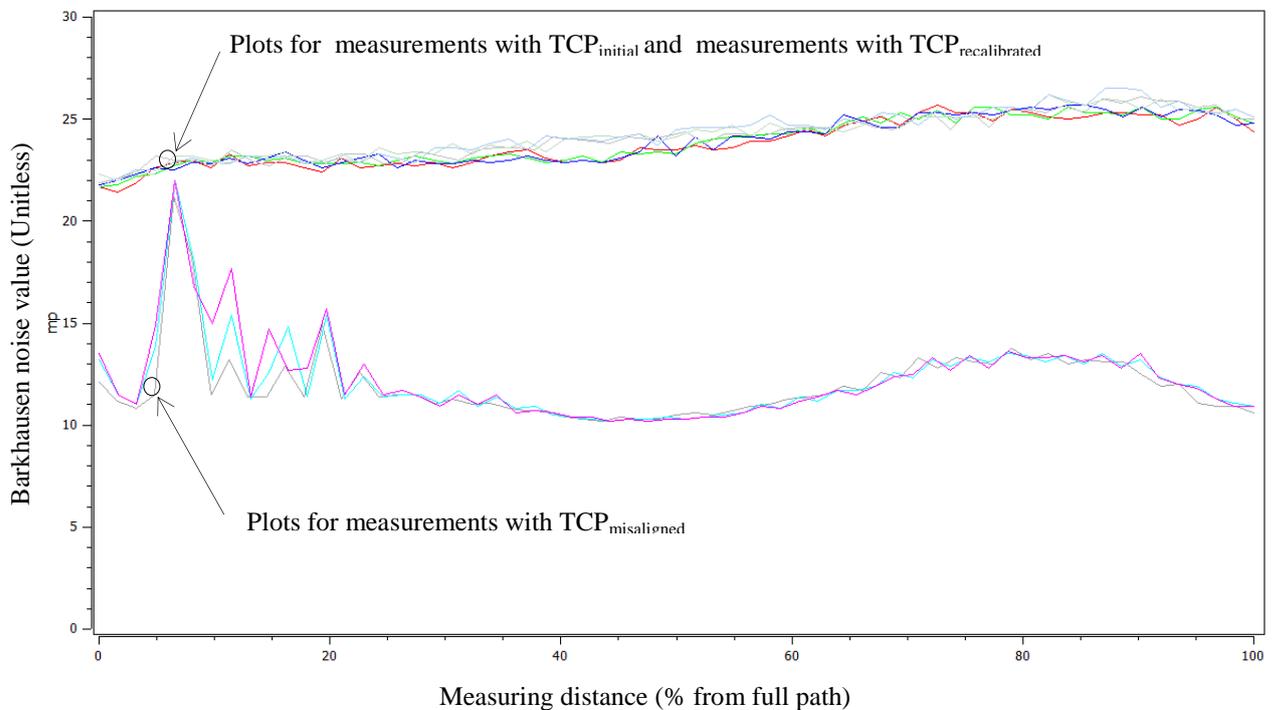


Figure 23. Plot of signals acquired from Sample 2 (X-axis: Measurement distance (%), Y-axis: Barkhausen noise signal value (Unitless))

8. ANALYSIS

8.1. Acceleration

The robot cell is off from production during the calibration sequence. All of the manufacturers of automatic calibration units are introducing calibration times for their customers. The procedure for the proposed TCP calibration in six degrees of freedom is constant, meaning that all of the movements are required. Therefore maximum acceleration was determined to reduce the time consumed. A deviation for each translation was calculated individually. Evaluation of the results is focused on the translations, since the calibration procedure for five degrees of freedom relies on the definition of translations in two different positions in the robot's coordinate frame's Z -axis. The tool center point is measured in respect to the tool's X - and Y -axes with two different heights, and rotations are acquired by defining a line through these center points. Rotations about the X - and Y -axes are therefore dependent on the accuracy of the translations on all axes. Rotation about the Z -axis is measured individually and therefore is not dependent on translations.

Figure 20 shows that the translation along the Y -axis grows when the acceleration is increased. The highest difference in comparison to the initial translation is 0.64mm. This is achieved when the acceleration is 240mm/s^2 . However, the translations about the X - and Z -axes seem to be constant in the entire sample and therefore are independent of the acceleration. This is explainable by the design of the fixture: the sensor is attached to a fixture that has two joints, allowing the sensor to self-align with a measured surface. The first joint allows rotation about the Z -axis of the tool and the second joint rotates about an axis parallel to the X -axis of the tool frame. The first rotation can have a minor effect on translation in both the X - and Y -axes. However, the later rotation directly affects the translation of the TCP in the Y -axis. In conclusion, if the acceleration is high, the inertia of the sensor head affects the calibration in the Y -axis. This is confirmed by noting that the fixture does not have joints that allow rotation about the Y -axis or translation along the Z -axis. Therefore the sensor is behaving as a rigid body and calibrated translations about the X - or Z -axes are not similarly affected by an increase of acceleration.

Spring-loaded joints in the fixture return the sensor head to an initial position after the force is removed. However, it is justifiable to presume that the manufacturing tolerances in the fixture joints do not allow complete return to the initial pose. Even a small angle at the joint would create an error in the translation of the sensor head. The results of the acceleration measurements introduce an increase in the translation along the Y-axis. This is considered to be caused by the inertia of the sensor head. Therefore the accelerations of the sensor head should be kept low so that the joint movement is not excited. With a rigid body tool, the acceleration and velocity can be increased. Limits for the acceleration and velocity are also exceeded when the light barrier cannot notice the tool properly or the controller digital input's frequency is exceeded. These values were not measured in this study, since calibration with such a high acceleration would introduce other problems, especially with this flexible and complex fixture.

8.2. Repeatability

In order to confirm the effect of acceleration and to evaluate the repeatability of this calibration method, two samples with constant accelerations and velocities were conducted. In the first sample the acceleration was set to be 300mm/s^2 . With this acceleration, the standard deviation of the results for X- and Y-axes was inside the robot's repeatability. Nevertheless, the spread of the occurrence in the histogram shown in Figure 21 highlights the problem: the difference between the maximum and minimum translations is unacceptable. This is also described in Table 2: for the X-axis the Max-Min difference is 0.51mm and for the Z-axis it is over 0.7mm. A difference of this magnitude between individual calibrations could drastically affect the signal quality of a Barkhausen noise measurement.

The second sample with an acceleration of 150mm/s^2 was initiated and the results obviously improved. The Max-Min difference in all translations was 0.3mm, which is acceptable for this robot. Considering that the largest positioning error due to robot repeatability is 0.2mm, the error from the calibration was now 0.1mm. The histograms in Figure 22 introduce the shape of normal distribution, and the parameter uncertainty caused by too high acceleration can be considered to be less. With these values, the full calibration sequence takes 90 seconds to complete. It is optimistically presumed that the robot

positioning error is 0.2mm and therefore the calibration error is only 0.1mm. Measurements with a more accurate robot would introduce the true error from the calibration. However, the velocity and acceleration used in the later measurement defend the calibration method repeatability and therefore it is suggested to use these values. In comparison to the calibrators available on the market, 90 seconds is not too much for six degrees of freedom. Manufacturers of automatic calibration units claim calibration times of less than 15 seconds in their brochures. However, calibration method proposed in this study is *full calibration*, so the only initial value needed is just a rough estimate of the translation in the Z-axis, which is the length of the tool. A quick TCP check can be conducted in the same 15 seconds, and if the TCP is translated, the full calibration would be conducted. This is the principal procedure for all automatic calibration units based on light barriers, since calibration for six degrees of freedom takes time.

8.3. Barkhausen noise signal quality

An acceleration of 150mm/s^2 and velocity of 50mm/s was selected for the last measurement. These measurements were conducted to analyze the signal quality acquired from Barkhausen measurements after the tool center point had been misaligned and recalibrated. The standard deviation of the signal acquired with the initial tool center points was 0.17 units. This states that the calibration of the initial TCP yields to proper alignment at the waypoint. In other words, when the calibration is conducted without significant change in the TCP, the signal quality is kept good between calibrations.

However, with the recalibrated TCP, the deviation of the average value for all samples was 0.45 units. This confirms that calibrating the geometric TCP with rather large errors increases the uncertainty of the calibration method. This is seen clearly in the results for Sample 2.2 in Table 4. The average value for three repeated measurements with initial TCP was 23.99. After these measurements, the sensor was translated along the Y-axis and rotated about the X- and Z-axes. As seen in Table 4, the misaligned TCP did not contact the specimen and the average value for all three measurements was 10.46 units, almost the value for the background noise in the laboratory. After recalibration, the average value for the three measurements was reduced to 23.12 and the combined deviation for initial TCP

and recalibrated TCP was 0.48. This indicates that the recalibrated TCP does not align perfectly similarly at the waypoint as the initial TCP does.

More positively, in Sample 2.1 the combined deviation of average values was 0.2 and in Sample 2.3 it was 0.1. In these samples, the sensor alignment is much more precise when compared to Sample 2. Two possible reasons could explain the large deviation in Sample 2; the calibration failed or the error was such that calibration of the geometric TCP did not yield a proper alignment for the entire tool. Since the repeatability measurements indicated that the geometric TCP was calibrated with a standard deviation of 0.1mm, the second reason is more probable and interesting.

If only the geometric tool center point is calibrated, the Barkhausen measurement can fail, since all of the waypoints were defined for the geometric tool center point. As explained previously, the actual TCP for the Barkhausen noise sensor is located at a position where the magnetic properties, for example the flux density, are optimal. Since this study was conducted for geometric calibration and a constant translation to the actual TCP was not introduced, it is possible that the deviations in Sample 2 are caused by incorrect alignment of the point that should be contacting the specimen.

9. CONCLUSIONS AND DISCUSSION

In comparison to other automatic calibration units, the studied method is simple. It is demonstrated that the calibrator does not require expensive hardware and can be attached directly to the robot controller without an additional control unit with embedded systems. The method itself consists of simple movements to allow the calculation of the tool center point in six degrees of freedom. These movements and calculations based on digital input signals are easy to transfer between robot types. It requires only that the operator is able to copy the movement procedure and trigger the digital inputs correctly. The method is faster than the original single beam calibration unit, since with two beams the center point is simultaneously defined for two points at different heights in the Z-axis. These two center points lead directly to the definition of the center axis, which allows the determination of the orientations in respect to the tool flange frame.

Considering that the accuracy of this calibration method is mainly a combination of the robot repeatability and the light barrier resolution, the acquired 0.3mm difference between the maximum and minimum values for translations is a success. Also the standard deviation of 0.1mm for each translation is what was expected from this study. However, the performance of the method itself is not easy to evaluate since the robot repeatability was poor. Therefore future research should be focused on the evaluation of this method with a robot that has better repeatability.

When evaluating the performance of the calibration method, some subjects were neglected. Light barrier resolution is a parameter that affects the accuracy. However, this is not so significant if the tool that is calibrated is relatively big. For example, in edge detection, if the light barrier resolution is 0.05mm, it can be considered that the tool edge is blocking 0.05mm from the light until the edge is detected and the high-signal is sent to the controller. Therefore it is 0.05mm too large. This is also repeated at the next edge; the tool is still blocking 0.05mm of the light when the low-signal has already been sent to the controller. This yields an error of 0.1mm when calculating the *width* of the tool. Nevertheless, if the light-barrier centerline is ideally aligned and the error due to the resolution is thereby symmetrical at both edges, the *center point* is correctly defined in the robot coordinate system since the error is symmetrically repeated at each edge.

Therefore, it is obvious that the accuracy is also dependent on the manufacturing tolerances of the calibrator body which leads to the alignment of the light barriers in respect to each other. A small deviation in the alignment between the light barriers is almost inevitable and the overall accuracy is reduced by this deviation. All of this was considered ideal in this study, since the focus is on the functionality of the method. Increasing the manufacturing tolerances will enhance the accuracy, and the error due to the manufacturing tolerances can be calculated without testing. Enhancing the resolution or the manufacturing tolerances would probably not give any noticeable results if calibrating with UR5. The robot's repeatability will hide the true results of the calibration accuracy. For future research, this method should be transferred to a more precise robot.

Measurements with Barkhausen signal were also a success for two reasons. Firstly, the results from Samples 2.1 and 2.3 indicate that the misaligned TCP can be corrected properly. Secondly, Sample 2.2 demonstrated that calibration according to the geometric TCP is probably not enough. The sensor head should be defined in such a way that a constant translation from the calibrated geometric TCP will lead to a TCP which is optimal for the measurements. When the waypoints are defined for this TCP, the orientation in respect to the specimen after recalibration will deviate less.

This method is restricted to rotationally symmetrical tools. In addition, the tool needs to have different width X - and Y -axes in order to calculate the tool center point rotation about the Z -axis. With the Barkhausen sensor it should be noted that the accuracy of the calibration is also dependent on the manufacturing tolerances of the sensor. With a handmade Barkhausen sensor the centerline might be a little different than that measured.

10. SUMMARY

The method was chosen by studying low-cost possibilities that utilize the robot itself as measuring equipment. This method defines the tool center point translations and rotations by detecting the edges of the tool in a robot coordinate system. Assuming a rotationally symmetrical shape, the center line is calculated from opposite center points between the detected edges. It is based on an original single beam calibration method that is simple and accurate. The advantage of using two beams is that this simultaneously defines two different center points. These center points directly reveal two rotations for the tool frame if it is considered to be at the end of the centerline. The last rotation about the Z-axis is finally defined by measuring tool width in different orientations.

The result confirms that the TCP can be calibrated only as accurately as the robot is. It is concluded that the method has an error of at least 0.1mm. However, the result also clarifies that the proposed method is fully functional for calibration of six degrees of freedom and errors can be minimized using more precise equipment. Calibration of the Barkhausen noise sensor revealed that recalibration of a large TCP error does not provide exactly similar orientation in all cases. When considering small deviations, the results in the Barkhausen measurements show that the TCP was defined correctly and the signal deviation was low. Repeatability tests confirm this.

The requirements for the calibration method were that it should define all six degrees of freedom, it should be simple and easy to transfer between robot units, and therefore it should require as little hardware as possible. The proposed calibration method complies with all of these requirements, although transferring the method to another robot type requires that the entire program is programmed again in the controller. Units on the market do not suffer from this issue. Their major advantage is that the calibration procedure is programmed into the embedded systems of the control unit and manufacturers only provide the communication setup between the control unit and the robot. The method in this study requires more labor. Nevertheless it is easier to customize to special tools such as handmade Barkhausen sensors.

REFERENCES

- Bajd, T., Mihelj, M., Munih, M. 2013. Introduction to Robotics. London: Springer.
- Lenarčič, J., Bajd, T., Stanišić, M.M. 2013. Robot Mechanisms. London: Springer.
- Leoni Protec Cable Systems GmbH, 2014. Advintec TCP 6D datasheet [web document]. [Referred 31.8.2014]. Available at:
http://www.leoni-industrial-solutions.com/fileadmin/leoni.com/downloads/pdf/produkte/en_advintec_TCP_6D.pdf
- Lindgren, M. 2003. Barkhausen Noise: Application to Residual Stress Measurements and Fatigue Damage Estimation. Ph.D. Thesis. Tampere: Tampereen Teknillinen Yliopisto. Publications 412.
- Mooring, B.W., Roth, Z.S., Driels. M.R. 1991. Fundamentals of Manipulator Calibration. New York: John Wiley & Sons Inc.
- Roth Z.S., Mooring, B.W., Ravani B. 1987. An Overview of Robot Calibration. *IEEE Journal of Robotics and Automation*. Vol. RA-3, pp. 377-385.
- Santa-aho, S. 2012. Barkhausen Noise Method for Hardened Steel Surface Characterization – The effect of Heat Treatments, Thermal Damages and stresses. Tampere: Tampereen Teknillinen yliopisto. Publications 1080.
- Siciliano, B. & Khatib, O. 2008. Springer Handbook of Robotics. Berlin: Springer.
- Sorsa, A. 2013. Prediction of Material Properties Based on Non-Destructive Barkhausen Noise Measurement. Oulu: University of Oulu. Faculty of Technology. Publications C442.
- Stefanita, C.-G. 1999. Surface Magnetic Barkhausen Response to Plastic Yield of Steel. Ph.D. Thesis. Queens University at Kingston. Ontario. Canada.
- Thorne, H.F. 1993. Tool Center Point Calibration Apparatus and Method. US Patent: 5,457,367. 1995.
- Universal Robots. 2013. UR5 - User Manual.
- Universal Robots. 2014. Press Kit – Company History & Profile [web document]. [Referred 22.7.2014]. Available at :
http://media1.limitless.dk/UR_Press_Kit/GB_United_Kingdom/UR_Profile_GB.pdf
- Wiest AG, 2014. Rotolab datasheet [web document]. [Referred 31.8.2014]. Available at:
<http://www.wiest-ag.com/products/rotolab>