

A critical investigation into the accuracy of a parallel mechanism coordinate measuring machine

by

Peter Labuschagne



*Thesis presented in partial fulfilment of the requirements for
the degree of Master of Engineering (Mechatronic) in the
Faculty of Engineering at Stellenbosch University*

Supervisor: Prof. K. Schreve

April 2019

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2019

Copyright © 2019 Stellenbosch University
All rights reserved.

Plagiarism Declaration

1. Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.
2. I agree that plagiarism is a punishable offence because it constitutes theft.
3. I also understand that direct translations are plagiarism.
4. Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.
5. I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Initials and surname: Peter Labuschagne	Date: April 2019

Abstract

A critical investigation into the accuracy of a parallel mechanism coordinate measuring machine

P. Labuschagne

*Department of Mechanical and Mechatronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.*

Thesis: MEng (Mech)

April 2019

The increasing demand for manufacturing high quality and high precision products implies that special methods are required to ensure the quality and precision of these manufacturing processes and techniques (Lin and Chen, 2001; Küng *et al.*, 2007). As a result, coordinate measuring machines (CMM's) are widely used as a means of precision measurement and verification of quality control. Ordinary CMM's have matured in their technology, yet are still lacking in their ability to measure at a sub-micron level (10^{-9} m). Furthermore, existing systems with sub-micron accuracy have a limited measurement range and are extremely costly (Sudatham *et al.*, 2016).

The process of designing and manufacturing micro-coordinate measuring machines (μ CMM's) is kept as a "trade secret" by existing market suppliers (Bommakanti, 2012). This has a negative effect on micro-manufacturing companies, since quality verification through the use of costly speciality CMM's, proves to be disadvantageous (Kruth *et al.*, 2002). Introducing a CMM that can achieve a measurement accuracy in the sub-micron level has shown evidence to be beneficial for the market of micro and nanomanufacturing systems (Wu *et al.*, 2012). However, the lack of theory supporting the challenging de-

ABSTRACT

sign and research of μ CMM's present great challenges in manufacturing and commercializing these systems.

This thesis endeavoured to investigate the accuracy of the current system developed by Rugbani (2014) to fashion improvements. Rugbani's (2014) system is unverified and does not yet achieve the desired accuracy. Therefore, investigating the accuracy of the μ CMM designed and manufactured by Rugbani (2014) provided useful insight into the current state of the machine. Furthermore, the Rugbani (2014) system illuminated accuracy influencing factors of the μ CMM such as the design, layout, and measurement equipment. The initial hypothesis envisioned improvements through temperature compensation. Therefore, the study endeavoured to improve measurement accuracy, perform baseline accuracy study on a calibrated artefact, and to incorporate and test thermal error compensation. However, in the course of the study, various fundamental design problems of the system were identified.

This thesis details the measurement test that was performed to examine the accuracy state of the μ CMM. Post-processing of the measurement showed unfavourable results with the original calibrated parameters. These parameters were tweaked, however, the results did not improve. Further tests were initiated to simplify methods of calibration to sanction improved regulation of these parameters. The mentioned tests evidenced that: the machine required large displacements for the laser measurement to change significantly; the ratio of probe movement to laser measurement is 20:1 and the tetrahedron has a position settling error. The study recommends an investigation into design improvement of the μ CMM to address fundamental design problems.

Uittreksel

'N Kritiese ondersoek na die akkuraatheid van 'n parallel meganisme koördinaat meet masjien

("A critical investigation into the accuracy of a parallel mechanism coordinate measuring machine")

P. Labuschagne

*Departement Meganiese en Megatroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: MIng (Meg)

April 2019

Die toenemende vraag na die vervaardiging van hoë gehalte en hoë akkuraatheid produkte impliseer dat spesiale metodes nodig is om die gehalte en presisie van hierdie vervaardigingsprosesse en tegnieke te verseker (Lin en Chen, 2001; Küng *et al.*, 2007). As gevolg hiervan, word Koördinaat Meetmasjiene (KMM'e) wyd gebruik vir die meting van akkuraatheid en die verifikasie van gehaltebeheer. Tradisionele KMM'e het in hul tegnologie gevorder, maar ontbreek die vermoë om op 'n sub-mikronvlak (10^{-9} m) te meet. Verder, bestaande stelsels met sub-mikron akkuraatheid het 'n beperkte meting reeks en is uiters duur (Sudatham *et al.*, 2016).

Die proses van die ontwerp en vervaardiging van Mikro-koördinaat Meet Masjiene (μ KMM'e) word as 'n "handelsgeheim" deur bestaande verskaffers gehou (Bommakanti, 2012). Dit het 'n negatiewe uitwerking op mikrovervaardigingsmaatskappye, omdat kwaliteit verifikasie deur die gebruik van duur spesialis KMM'e, blyk nadelig te wees (Kruth *et al.*, 2002). Daar is getuienis dat die gebruik van 'n KMM met 'n meetings akkuraatheid op 'n sub-mikron

UITTREKSEL

vlak voordelig is in die mikro- en nanovervaardiging stelsel mark (Wu *et al.*, 2012). Die gebrek aan onderliggende teorie vir die uitdagende ontwerp en navorsing van μ KMM'e verteenwoordig groot uitdagings vir vervaardiging en kommersialisering van hierdie stelsels.

Hierdie tesis poog om die akkuraatheid van die huidige stelsel wat deur Rugbani (2014) ontwikkel is te ondersoek met die doel op verbeterings. Rugbani (2014) se stelsel is ongeverifieer en bereik nog nie die gewenste akkuraatheid nie. Die ondersoek na die akkuraatheid van die μ KMM ontwerp en vervaardig deur Rugbani (2014) het nuttige insig verskaf na die huidige stand van die masjien. Verder, het Rugbani (2014) die bydraende faktore tot die μ KMM akkuraatheid verlig, soos die ontwerp, uitleg en sensore. Die aanvanklike hipotese het verbeterings deur temperatuur kompensasie voorgestel. Daarom, het die studie gepoog om die metings akkuraatheid te verbeter, basislyn akkuraatheidstudie op 'n gekalibreerde artefak uit te voer, en inkorporeer en toets termiese-foutkompensasie. Deur die loop van die studie was verskeie fundamentele ontwerpprobleme van die stelsel egter geïdentifiseer.

Daarom, word hier basislyntoetse beskryf wat ontwerp was om die akkuraatheid van die μ KMM vas te stel. Naverwerking van die metings het ongunstige resultate getoon met die oorspronklike gekalibreerde waardes. Hierdie waardes was verander, maar die resultate het nie verbeter nie. Verdere toetse is onderneem om vereenvoudigde metodes vir die kalibrasie van hierdie parameters te ontwikkel. Die genoemde toetse het die volgende bewys: die masjien benodig groot verplasinge vir die lasermeting om aansienlik te verander; die verhouding van aftasterbeweging tot laserverplasing is 20:1 en die tetraëder het 'n posisie-vestigingsfout. Die studie beveel 'n ondersoek aan in die ontwerpverbetering van die μ KMM om fundamentele ontwerp probleme aan te spreek.

Contents

Abstract	iii
Uittreksel	v
Contents	vii
List of Figures	ix
List of Tables	xi
Nomenclature	xii
1 Introduction	1
1.1 Background	1
1.2 Objectives of the study	3
1.3 Motivation for the study	4
1.4 Overview of thesis	4
2 Literature review	5
2.1 Measurement standards	5
2.2 Calibration	7
2.3 Optimization algorithms	11
2.4 Temperature measurement	12
2.5 Error measurement	17
2.6 The μ CMM	20
2.7 Existing parallel CMM	25
2.8 Literature conclusion	26
3 The μCMM, user interface and software	27
3.1 Machine description	27
3.2 Machine characterization	30
3.3 Joysticks	30
3.4 Arduino [®] and LabVIEW [™] interface	32
3.5 Machine settling time	34
3.6 Machine measurement	34

CONTENTS

4	Thermal error model	36
4.1	Simple model derivation	36
4.2	Error model results	43
5	Accuracy test	44
5.1	Calibrated measurement specimen	44
5.2	Experimental procedure	44
5.3	Measurement results	47
5.4	Measurement conclusion	48
6	μCMM investigation	49
6.1	Manual parameter tweaking	49
6.2	Numerical calibration	50
6.3	Error investigation	58
6.4	Investigation conclusion	61
7	μCMM calibration	63
7.1	Calibration method	64
7.2	Calibration data processing	64
7.3	Calibration results	67
8	Post-calibration	69
8.1	Micrometer stage calibration test	69
8.2	Micrometer stage test results	71
9	Conclusion	73
	Appendices	76
A	Operation manual	77
B	Optimization code	81
C	Post calibration processing	83
D	Post-processing code	89
E	Thermal error model code	100
F	Brute force calibration code	107
	List of References	111

List of Figures

1.1	Parallel mechanism coordinate measuring machine.	2
2.1	Possible artefacts for use in CMM calibration verification (ISO 10360-2:2009).	6
2.2	Laser tracker used for calibration (Jeon <i>et al.</i> , 2010).	8
2.3	Laser interferometer used for calibration of a sphere (Jansen <i>et al.</i> , 2010).	8
2.4	Laser interferometer setup used for measuring displacement (Wu <i>et al.</i> , 2012).	9
2.5	A surface-mounted thermocouple with insulated wires (Kerlin Johnson, 2012).	13
2.6	Thermocouple point measurement (Kerlin Johnson, 2012).	14
2.7	Typical layout of an RTD (Hughes, 1995).	14
2.8	Three-wire RTD Wheatstone bridge configuration (Hughes, 2015).	15
2.9	Temperature sensor curve shape comparison (Hughes, 2015).	16
2.10	Leadscrew bearing housing showing temperature concentrations and gradients (Wu <i>et al.</i> , 2012).	18
2.11	Gap sensor measurement jig and setup (Pahk and Lee, 2002).	19
2.12	(a) Parallel mechanism (Oiwa, 2000) and (b) a serial mechanism (Salsbury, 2016).	20
2.13	The deformation of a serial mechanism shown by the dotted lines (Barakat <i>et al.</i> , 2000).	21
2.14	μ CMM coordinate system (a) top view and (b) side view (Rugbani, 2014).	22
2.15	Showing the reachable workspace (a) top view, (b) front view, and (c) side view of the μ CMM (Rugbani, 2014).	22
2.16	Change in pivot point height as probe is displaced (Rugbani, 2014).	23
2.17	The Equator TM 300 parallel mechanism CMM with working volume and dimensions (Renishaw [®] , 2016).	25
3.1	(a) The μ CMM frame structure and (b) tetrahedron.	28
3.2	Spherical joint attached to a bracket of the linear slider.	29
3.3	Probe used for the μ CMM.	29
3.4	Hardware interaction on the μ CMM.	30

LIST OF FIGURES

3.5	Serial device identification and selection.	31
3.6	LabVIEW™ block diagram for joystick index identification, setting and operation.	31
3.7	(a) Original and (b) new LabVIEW™ block diagram for Arduino® communication.	33
3.8	Motor encouder counts vs. settling time.	34
4.1	Machine top view with boundary conditions on (a) original and (b) deformed μ CMM.	38
4.2	μ CMM sideview with arbitrary probe position.	40
4.3	Diagram of the displaced μ CMM.	41
4.4	Arm l_k forming a triangle to calculate new x- and y-points.	42
5.1	Specimen measurement process.	45
5.2	Example of probing a gauge block with only two points.	45
5.3	Post-processing method followed after a specimen measurement.	46
5.4	Calculated top- and side-plane from accuracy test.	47
6.1	Process outline for numerical calibration procedure.	51
6.2	Height gauge measuring the change in pivot point height on the lead screw bracket.	51
6.3	Error data for three measurements performed with the GA algorithm.	53
6.4	Dead distances needed to calculate each pivot point z-height.	54
6.5	A plot of the combined error of each arm.	55
6.6	Individual and combined brute force calibration error results.	57
6.7	Measurement errors for calibration using a large dataset and large displacements.	60
7.1	The tetrahedron on CMM ₁ 's platform for calibration.	63
7.2	A visualisation of slider paths, maximum slider movement, and probe point.	65
7.3	Illustration for a probe point not on the linear path of the slider.	66
7.4	Laser distance value plot as a function of data points recorded.	68
8.1	3-Axis micrometer stage.	70
A.1	USB hub for connecting lasers and motors.	77
A.2	Serial connection between motors.	77
A.3	Power supply.	78
A.4	Connection between Arduino® and probe reader.	78
A.5	Motor height bracket.	78
A.6	Structure spacer bracket.	79

List of Tables

2.1	Classification of Calibration Parameters. Adapted from: Takamasu <i>et al.</i> (2002).	10
2.2	Four main types of CMM errors.	18
2.3	Equator TM 300 specifications. Adapted from: Renishaw [®] (2016).	26
3.1	Original calibrated parameters of Rugbani (2014).	29
3.2	Format of saved data from μ CMM measurements.	35
4.1	Fixed and variable coordinate points for the μ CMM.	38
4.2	Thermal error model probe coordinate calculation.	43
5.1	Post-processing results.	48
6.1	Summarised results from before and after tuning for different measurement data.	49
6.2	Numerical optimization results for one arm.	52
6.3	Three arm calibration results for 15 measurement points per arm.	54
6.4	Calibrated parameters.	55
6.5	Individual arm calibration error results.	56
6.6	Start of measurement laser data for test 1, 2 and 3.	58
6.7	Laser distances for starting point.	59
6.8	Repeated ratios of laser versus height gauge displacement.	61
7.1	Calibrated parameter values.	67
8.1	3MS measurement experiment results.	71
8.2	3MS results with variable machine parameters.	72

Nomenclature

Constants

$$g = 9.81 \text{ m/s}^2$$

$$\alpha = 23.1 \times 10^{-6} \text{ K}^{-1}$$

Variables

a	Pivot point	[]
b	Pivot point	[]
b_{in}	Distance between pivot points	[mm]
c	Pivot point	[]
d	Dead distance	[mm]
err	Error	[]
h	Length	[mm]
L	Vector length	[mm]
l	Length	[mm]
R	Resistance	[Ω]
T	Temperature	[$^{\circ}\text{C}$]

NOMENCLATURE

β	Angle	[mm]
Δ	Change in	[]
δ	Distance	[mm]
ψ	Angle	[°]
θ	Angle	[°]

Vectors and Tensors

\vec{n}	Normal vector
\vec{P}	Point vector
\vec{r}	Parametric vector
\vec{S}	Position vector
\vec{v}	Measurement line vector
σ	Standard deviation

Subscripts

A	Structure column point
a	Arm
B	Structure column point
b	Arm
C	Structure column point
c	Arm
err	Error

NOMENCLATURE

<i>f</i>	Finish
<i>i</i>	Current arm
<i>in</i>	Current arm to next arm
<i>ip</i>	Current arm to previous arm
<i>S</i>	Side
<i>s</i>	Start
<i>T</i>	Top
<i>x</i>	Coordinate
<i>y</i>	Coordinate
<i>z</i>	Coordinate
0	Original

Superscripts

'	New
---	-----

Chapter 1

Introduction

1.1 Background

The increasing demand for manufacturing high quality and high precision products implies that special methods are required to ensure the quality and precision of these manufacturing processes and techniques (Lin and Chen, 2001; Küng *et al.*, 2007). As a result, coordinate measuring machines (CMM's) are widely used as a means of precision measurement and verification of quality control. Ordinary CMM's have matured in their technology, yet are still lacking in the ability to measure at a sub-micron level (10^{-9} m). Furthermore, existing systems—with sub-micron accuracy—have a limited measurement range and are costly (Sudatham *et al.*, 2016).

The process of designing and manufacturing micro-coordinate measuring machines (μ CMM) is kept as a "trade secret" by existing market suppliers (Bommakanti, 2012). This has a negative effect on micro-manufacturing companies, since quality verification through the use of costly "speciality" CMM's, proves to be disadvantageous (Kruth *et al.*, 2002). Introducing a CMM that can achieve a measurement accuracy in the sub-micron level has shown evidence to be beneficial for the market of micro and nanomanufacturing systems (Wu *et al.*, 2012). However, the lack of theory supporting the challenging design and research of μ CMM's present great challenges in manufacturing and commercialising these systems.

Stellenbosch University's Mechanical and Mechatronic Engineering Department set a goal to design and manufacture a μ CMM that would help micro-manufacturing companies realize the quality control necessary. The novel μ CMM machine was designed and manufactured by a former PhD student, Rugbani (2014). Figure 1.1 shows the μ CMM of Rugbani (2014), which is discussed in detail in a later chapter. The accuracy aimed for

by Rugbani (2014) was in the order of 10 μm for a measurement range of 100 mm x 100 mm x 100 mm. While Rugbani's (2014) model and prototype showed promise of achieving this accuracy, it has not yet been achieved. Therefore, further investigation of the influencing factors is still needed.

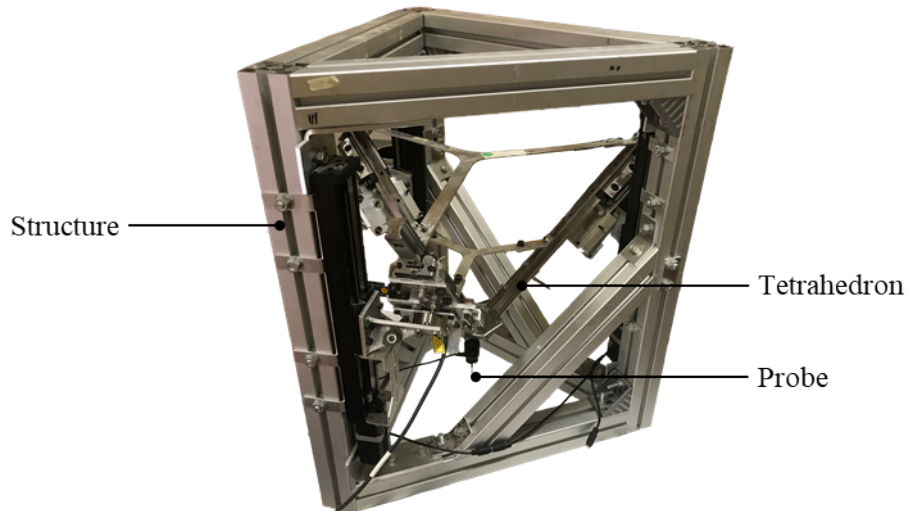


Figure 1.1: Parallel mechanism coordinate measuring machine.

Before the influencing factors could be investigated, the accuracy of the μCMM needed to be determined. This was achieved through various methods, one of which was to measure a calibrated artefact where the exact dimensions were known. As a result, the measurement value provided by the μCMM could be compared to the calibrated artefact value, with the comparison result providing the accuracy of the μCMM . With the accuracy known, a better idea was given to illuminate the steps necessary for the investigation of potential accuracy influencing factors. These factors may include the design of the machine, assembly of the machine, the actuation equipment, and the measurement equipment used.

Barakat *et al.* (2000) state that, in addition to the high cost, increasing the measurement accuracy of a CMM through the use of higher precision hardware does not overcome system design errors or limitations. Consequently, CMM manufacturers have placed emphasis on the design phase in order to increase measurement accuracy. Vermeulen *et al.* (1998) postulates that the design of a μCMM should aim at minimising factors such as vibration and machine deformation from load and temperature factors. However, physical limitations are inherent to the technologies and material available today. Other post-processing factors, such as calibration or error compensation, can aid in machine accuracy (Küng *et al.*, 2007).

Jeon *et al.* (2010) emphasises that in addition to the unknown CMM parameters, manufacturing and assembly inaccuracies present a further need for calibration. These manufacturing inaccuracies affect aspects of the μ CMM's geometry, such as: the position of the probe, joints, lasers, and motors. Therefore, the theoretical geometry of the μ CMM is not the same as the geometry of the physical machine. For this reason, the model that describes the μ CMM and its parameters can be adjusted as a result of calibration. The calibrated parameters are then used for the equations that calculate the μ CMM probe pose. Once these factors were completely characterized and understood, an investigation of thermal errors, measurement equipment, and actuation equipment was made.

A previous study evidenced that, when measuring to extremely high precisions (in the order of 10^{-9} m), a small change in temperature affects the structure to the extent that the measurement data of the CMM's can no longer be considered completely reliable (Schmidt, 2014). Heat is generated by moving parts such as motors, friction on leadscrews, and can also be generated by measurement lasers and other components. Up to 70 % of measurement errors are thermal related, called quasi-static errors (Barakat *et al.*, 2000). Quasi static errors create a need to compensate for the negative effects that temperature changes have on CMM's.

The aim of this project was to investigate the accuracy of the current system and then improve it. The initial hypothesis was that this could be done through temperature compensation. However, in the course of the study, various fundamental design problems of the system were identified. Thus, the focus of the study changed to understand the fundamental design problems of the mentioned system.

1.2 Objectives of the study

The overall objective was to improve the accuracy of the μ CMM. To do this, the project endeavoured to investigate the current accuracy by performing a baseline test which measured calibrated artefacts. Once the accuracy was verified, a further objective was to increase the measurement accuracy of the existing μ CMM, through the implementation and testing of temperature compensation.

The study endeavoured to:

- improve measurement accuracy,
- perform a baseline accuracy study on a calibrated artefact, and

- incorporate and test thermal error compensation.

The baseline study revealed that an objective should be added for calibration of the system parameters. This, in turn, placed less emphasis on thermal error compensation.

1.3 Motivation for the study

Currently, Rugbani's (2014) system is unverified and does not yet achieve the desired accuracy. Therefore, investigating the accuracy of the μ CMM designed and manufactured by Rugbani (2014) will provide useful insight into the current state of the machine. Furthermore, this will illuminate accuracy influencing factors of the μ CMM, such as the design, layout, and measurement equipment. This will, in turn, uncover advantages and disadvantages of the μ CMM in question.

As a result, a μ CMM that achieves a high level of accuracy will not only be beneficial to Stellenbosch University but to the field of metrology as well. The design and manufacture of high accuracy CMM's will be explored. Furthermore, methods of achieving high levels of accuracy and possible accuracy influencing factors will be provided. This will aid CMM—and even computer numeric control (CNC) machine—developers to design and develop more competitive products and push the metrology industry to reach new heights. In turn, CMM machines could be used for higher precision manufacturing processes as the world tends toward nano-technology (Patnaik and Raxit, 2014).

1.4 Overview of thesis

The report documents the methods used to investigate the μ CMM's accuracy, calibrate it, and develop a thermal error compensation model.

Chapter 2 provides a literature overview on topics relevant to the project to support the project objectives. Chapter 3 discusses the μ CMM used, in detail, which includes software and hardware aspects of the system. Chapter 4 develops a thermal error compensation model for the μ CMM. The baseline test of determining the μ CMM accuracy is presented in Chapter 5 and includes the experimental procedure and results. Chapter 6 investigates the μ CMM and possible accuracy influencing factors. Chapter 7 aims to increase the accuracy of the μ CMM by performing various types of calibration and providing the results. Chapter 8 provides the details of the results for the most successful calibration performed, along with an accuracy test for the newly calibrated parameters. Chapter 9 concludes with the findings of the project.

Chapter 2

Literature review

2.1 Measurement standards

Attempts at achieving the project objective of increasing a current CMM's measurement accuracy to 100 nm can be rendered void if metrology standards or methods are not adhered to. The standards used in this document are from the International Organization for Standardisation (ISO) since their standards are globally recognized. This section provides a brief description of standards that need to be considered before carrying out any experiments and should be strictly adhered to during experiments.

2.1.1 Geometrical measurement of a specimen

The global CMM market consists of various machines, each with unique features and capabilities. As a result, the ISO standard for Geometrical Product Specifications (GPS) has provided several standards that cater for a large portion of CMM's in the field of metrology. These standards form part of the ISO 10360 group. ISO 10360-1, -2 and -6 are the only standards applicable to the type of CMM used in this research project.

2.1.1.1 ISO 10360-1:2000

This standard forms part one of the GPS and provides the general vocabulary used for metrology in the ISO 10360 standards. The vocabulary related to ISO 10360, commonly used in this document, will be provided where necessary. However, if further knowledge of the standard's vocabulary is desired by the reader, ISO 10360-1:2000 should be studied.

2.1.1.2 ISO 10360-2:2009

This forms part two of the metrology standard and is concerned with CMM's used for measuring linear dimensions. Specified in part two are "acceptance tests for verifying the performance of a CMM used for measuring linear dimensions as stated by the manufacturer" (ISO 10360-2:2009). The standard is only applicable to Cartesian CMM's that make use of a contacting probe system that performs measurements in a discrete-point probing mode.

The standard provides environmental and metrological requirements. These cover condition requirements of the environment, where the environment chosen must remain within specified limits for temperature, humidity, etc. Also included are operating conditions required for the CMM, limits for length measurement error, repeatability range, and workpiece loading effects. The manufacturer is given the freedom to choose a specimen that represents the calibrated test length. Annexe B and Annexe D of the standard provide allowed specimen choices. Examples of possible specimens are provided in Figure 2.1. The calibrated test length is a linear measurement length that the CMM is calibrated for, which is specified by the manufacturer. A set of specified positions on these specimens are to be measured and compared to actual values. This process is repeated to evaluate measurement repeatability.

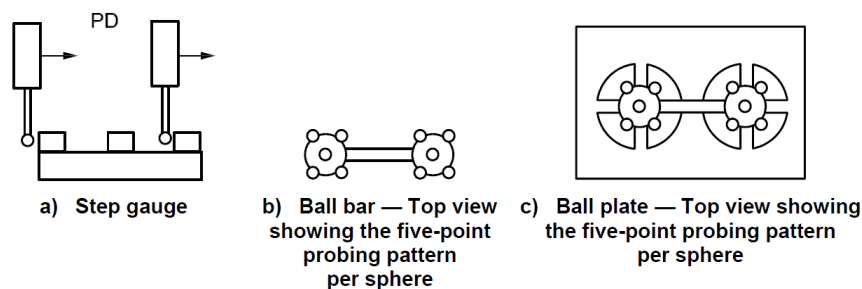


Figure 2.1: Possible artefacts for use in CMM calibration verification (ISO 10360-2:2009).

2.1.1.3 ISO 10360-6:2014

Part 6 of the ISO 10360 standard evaluates software features of the CMM. In particular, CMM software features that support the measurement of various shapes and features. These include "the line (in two and three dimensions), the plane, circle (in two and three dimensions), the sphere, the cylinder, the cone and the torus" ISO 10360-6:2014. In particular, part six of ISO 10360 provides the appropriate methods to test each software feature. The standard requires that one or more tests be performed on each software feature of the available CMM-features specified by the manufacturer.

The basic test requirements section provides general guidelines for performing tests on a CMM. The ISO 10360-6:2014 document provides more meticulous requirements that will be used, but not discussed. The basic requirements of the software should provide features to allow:

- Recording of raw measurement data prior to post-processing, such as error correction.
- Automatic computation of the feature type under test, when the respective coordinates have been entered into the software.

2.2 Calibration

Calibration is a necessary procedure for a machine with a complex kinematic structure as it creates the link between the actual probe positions and the computer calculated positions (Jeon *et al.*, 2010). This is a crucial step for the accuracy of the μ CMM, yet the precision of calibration can very often be overlooked. As a result, investigating and applying various methods for calibration could improve the accuracy of the μ CMM.

It is important to understand that if every part fitted together perfectly and there was no play or tolerances between joints and there were no machine deformations and everything was exactly according to the theoretical model of the system, then calibration would not be necessary. However, this is not a realistic expectation, thus calibration is a crucial step for the accuracy of the CMM.

2.2.1 Pose measurement

The literature on the calibration of CMM's provides various methods used to measure the probe position during calibration. These methods are discussed in this section.

2.2.1.1 Laser tracker

Verl *et al.* (2008) and Jeon *et al.* (2010) use a laser tracker to measure the probe coordinates during calibration (Figure 2.2). The measurement is made possible by using laser vectors to determine the distance of the target measured from the laser tracker, which outputs 3D-coordinates. As the pose of the CMM changes, the laser tracks the probe from a reference point, which allows the output of the necessary x-, y-, and z-coordinates.

A laser interferometer could also be used in place of a standard laser tracker. Figure 2.3 shows a laser interferometer being used for calibration of a sphere.



Figure 2.2: Laser tracker used for calibration (Jeon *et al.*, 2010).

Additionally, Figure 2.4 provides an illustration of a laser interferometer in use with a CNC machine. A laser interferometer can only track the displacement of a particle in one dimension, which is the distance between the laser interferometer and the particle. However, the laser tracker uses encoders that measure the rotation of the laser interferometer in two planes. Therefore, the data can be used to determine the x-, y- and z-coordinates of the particle as it displaces. In this case, the particle of concern is the CMM probe. Okafor and Ertekin (2000) use a laser interferometer that shows an accuracy of 0.1 pm.

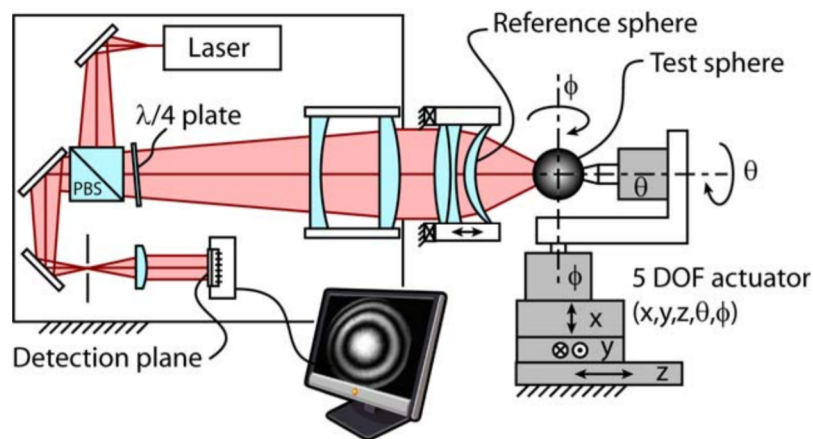


Figure 2.3: Laser interferometer used for calibration of a sphere (Jansen *et al.*, 2010).

The laser interferometer is a very high accuracy displacement sensor with high reliability its accuracy. However, these sensors are very costly and they require expensive software to interpret sensor data as well.

2.2.1.2 Capacitive sensor

A capacitive sensor (often referred to as a gap sensor) can also be used as a means of providing feedback for the probe position (Park *et al.*, 2006). The more cost-effective capacitive sensors are generally uni-directional, hence, one capacitive sensor would be required for each axis of movement.

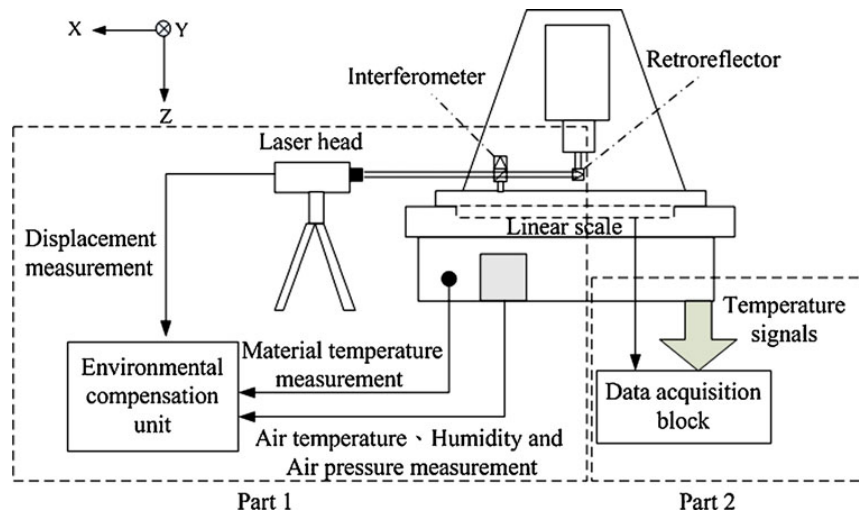


Figure 2.4: Laser interferometer setup used for measuring displacement (Wu *et al.*, 2012).

Typical capacitive sensors have the ability to measure to precisions of up to 0.05 mm (50 μm), with special nano-positioning capacitive sensors being able to measure up to 0.04 nm (Micro-Epsilon, 2017). However, these sensors only have a limited range and should, therefore, be placed close to the subject being measured. Moreover, capacitive sensors only work on a selection of materials, such as stainless steel, etc., that have capacitance related properties.

2.2.1.3 Measurement instruments

In certain cases, Takamasu *et al.* (2002) states that "*some parameters can still be identified directly with a high accuracy using measuring instruments*". If parameters need to be measured, but an extremely high level of accuracy is not required, then systems such as verniers can be used for these measurements. Additionally, the errors that are experienced when measuring these parameters with conventional manual measuring instruments, can be accounted for during calibration as well. The advantage of measuring some parameters with conventional instruments is to reduce the calibration time and complexity.

2.2.1.4 CMM measurement

Takamasu *et al.* (2002) use another—already calibrated—CMM (denoted by CMM_1) to measure the pose of the CMM to be calibrated (denoted by CMM_2). In addition to using conventional measuring instruments, mentioned in Section 2.2.1.3, CMM_1 probes the probe of CMM_2 while in multiple poses. This data is recorded and provides the coordinates for each probe position related to each pose of CMM_2 . Since the probe being measured is spherical, a set of five points need to be measured on the probe tip of CMM_2 . In turn, the roundness, radius, and centre coordinates of the probe tip are measured.

Table 2.1: Classification of Calibration Parameters. Adapted from: Takamasu *et al.* (2002).

Classification		Parameters
geometric calibration	kinematic (size) calibration	size dimensions, positions, angular dimensions, etc.
	form-deviation calibration	scale, straightness, pitching, yawing, rolling, etc.
non-geometric calibration		deformation, clearance, dynamic effect, etc.

Dai *et al.* (2009) achieve a measurement repeatability of 1.5 nm on their nano-CMM. This proves the availability of CMM's to measure and calibrate a μ CMM. As a result, this is a common method used for calibration, but can be expensive if an additional CMM is not accessible for calibration (Saputra *et al.*, 2014; Takamasu *et al.*, 2002; Verl *et al.*, 2008).

2.2.2 Calibration types

Takamasu *et al.* (2002) states that there are two types of calibration: (1) geometric calibration, and (2) non-geometric calibration, which are further described in Table 2.1. Kinematic calibration allows for crucial characterization of a machine, especially if the probe pose is calculated based on size dimensions, positions, and angular dimensions. Form-deviation is better suited to further increase the accuracy of an existing calibration where parameters, such as straightness of upright columns may have an effect on machine accuracy. Non-geometric calibration covers aspects such as dynamic effect, where moving parts could generate heat and, in turn, invoke a thermal error. However, Verl *et al.* (2008), Jeon *et al.* (2010), Saputra *et al.* (2014), Wampler *et al.* (1995), and Takamasu *et al.* (2002) place the highest importance on kinematic calibration above other calibrations. There are various ways of kinematic calibration and Verl *et al.* (2008), and Jeon *et al.* (2010) provide one such alternative, which is numerical calibration. Others physically measure the kinematic parameters of the CMM for calibration.

2.2.2.1 Numerical calibration

Numerical calibration is the process of using numerical optimization to calibrate a machine so that measurement errors are at a minimum (Verl *et al.*, 2008). When applied to CMM's, a physical measurement is made on parts of the machine that move and form part of the independent variables used to calculate the final probe pose, such as actuator angles or actuator displace-

ment. Measurement methods mentioned in Section 2.2.1 can also be used for these. The results of the algorithm used to calculate the machine parameters, which in turn calculate the probe pose, are compared to the measured results. The numerical optimization process is then used to calibrate the kinematic parameters of a CMM by reducing the error between measured and calculated variables.

Verl *et al.* (2008) place emphasis on finding a set of well conditioned measurement poses for numerical calibration. The following steps are provided:

1. Find a min ($K_0(H(Y_i))$) for the minimal set of needed poses;
2. Find a min ($K_0(H(Y_{i+1}))$) by varying the y_{i+1} pose;
3. Find a min ($K_0(H(Y_{i+1}))$) by varying the y_i, \dots, y_{i+1} pose;
4. If the i th pose has not changed ($(K_0(H(Y_{i+1})))$) for r iterations, fix it and do not vary anymore;
5. Go to step 2

Where K_0 is the condition number of the identification matrix $H(Y_m)$, with lowest achievable condition number being 1. $H(Y_m)$ is the matrix of combined Jacobians of different poses. Y_m is a matrix of real-world coordinates y_m . In the case that a derivation of the above steps is required, the paper of Verl *et al.* (2008) can be studied.

2.2.2.2 Parameter measurement

Parameter calibration is the process of identifying the unknown machine parameters that relate the machine movement to the probe pose (Takamasu *et al.*, 2002). The parameters are typically measured using methods mentioned in Section 2.2.1 and can achieve very high levels of accuracy for calibration.

2.3 Optimization algorithms

Generally, in order to achieve numerical calibration, an optimization algorithm needs to be used (Chanal *et al.*, 2007). Therefore, this section summarizes methods applicable to the use of numerical calibration for a CMM. Since MATLAB[®] is the platform used for numerical calibration, all algorithms investigated and mentioned will be for use in MATLAB[®] and are unconstrained. Importantly, the details of these algorithms will not be described in this section. Instead, their application for the numerical calibration along with their advantages and disadvantages will be described.

2.3.1 Linear descent algorithm

The function "fminsolve" is a linear descent algorithm used for finding the minimum of an objective function (Snyman, 2005). This method requires minimal computing power, however, can often require a large amount of computational time as opposed to more developed algorithms. Nonetheless, this algorithm can still be applied to the numerical calibration process of Rugbani's (2014) system, since only nine variables need to be calibrated for, as will be shown later (Chapter 6). A drawback of this method is that it can converge on a local minimum, which is largely dependent on the starting point of iterations. Getting a good starting point for the algorithm is often a very difficult task.

2.3.2 Genetic algorithm

Genetic Algorithms (GA's) are entirely different and modern optimization methods forming part of a zero order method (Snyman, 2005). This means that the algorithm does not use the conventional approaches of line or gradient descent for optimization. GA's can be very computationally expensive if an objective function with a large set of data is to be optimised. The advantage is that this algorithm is not limited to a local minimum based on a starting point. The stochastic nature of the GA means that, with a few iterations, finding the global minimum is more likely. When the objective function and variables do not require much computing power, the computational cost of a GA is negligible.

2.4 Temperature measurement

The large impact that temperature has on the CMM system means that a large amount of emphasis and effort should be placed on correctly measuring CMM and ambient temperatures. There are several different methods that can be used for measuring the CMM structure temperature and ambient temperatures. This section will discuss different types of equipment and configurations used for measuring temperature.

2.4.1 Thermocouples

Thermocouples are arguably the simplest type of device used for measuring temperature (Kerlin and Johnson, 2012). The thermocouple uses a basic concept in which two wires of different material type—with dissimilar characteristics—are joined at one end. When these wires experience a change in heat, a change in voltage potential can be measured. An increase in temperature produces an increase in voltage. The measured voltage can be translated into a temperature value, which is calibrated by the manufacturer. Surface-

mounted thermocouples (Figure 2.5) can be used in one of three main configurations.

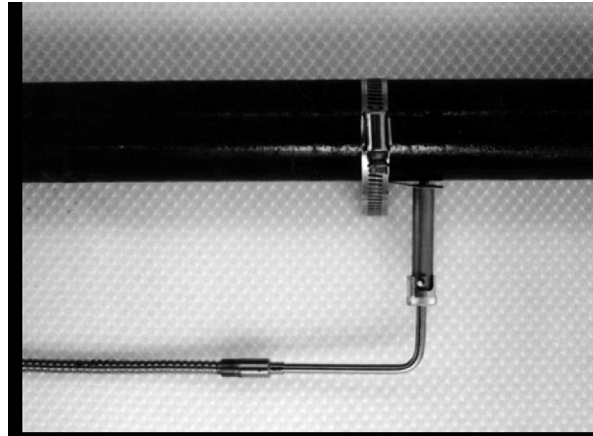


Figure 2.5: A surface-mounted thermocouple with insulated wires (Kerlin Johnson, 2012).

The first configuration is point measurement with the surface insulated from the thermocouple wires. Figure 2.5 shows a sheathed thermocouple, which is the most common type of surface-mounted thermocouple since they can be used in conditions that would normally be unsuitable for bare wires. Thermocouple surface contact is maintained by means of either spring loading, clamping, or welding. The second configuration is point measurement with the surface integral to the thermocouple circuit (Figure 2.6). In this configuration, bare thermocouple wires are attached to a surface by either bolting or riveting. Although an extra material has been introduced at the junction of the thermocouple, the measurement integrity remains intact as the additional material does not affect it.

The third configuration is the temperature difference measurement. This method measures a change in temperature between two surface points. This suggests that two thermocouples can be used to measure the temperature difference. However, differential thermocouples are better suited for this application since they are specifically calibrated and designed for this configuration.

2.4.2 Resistance temperature detector

Resistance Temperature Detectors (RTD's) make use of semiconductors or metals. When an RTD experiences an increase or decrease in temperature, its resistance increases or decreases, respectively. The change in resistance can be measured and converted to temperature values by using its calibration data (Hughes, 1995). These temperature sensors are the second most used type of temperature sensor since they have appealing characteristics, such as:



Figure 2.6: Thermocouple point measurement (Kerlin Johnson, 2012).

- Simple to use and manufacture.
- Good accuracy.
- Provides good stability.

Hughes (2015) describes the RTD to be a wire that is wound around a ceramic bobbin, which is then sealed. Typical materials used are platinum for the wire and glass for the seal. Figure 2.7 further illustrates the concept of an RTD.

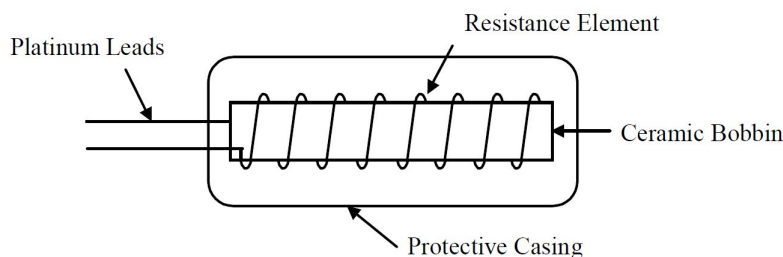


Figure 2.7: Typical layout of an RTD (Hughes, 1995).

A more modern type of RTD is the metal-film RTD. This is a much smaller device that can provide higher accuracy over a linear range. The metal-film RTD have a quicker response time to changes in temperature due to the small size. The metal-film RTD is used in a Wheatstone bridge circuit. This allows a voltage value to be read from the change in resistance from the RTD, which can

then be converted to a temperature. An RTD can be used in two- or three-wire configuration. However, the only configuration of concern in this document is the three-wire configuration as it provides the benefit of eliminating lead wire resistances. As a result, the RTD sensor layout remains accurate, given that all lead wires are the same length. Figure 2.8 illustrates the three-wire Wheatstone bridge circuit.

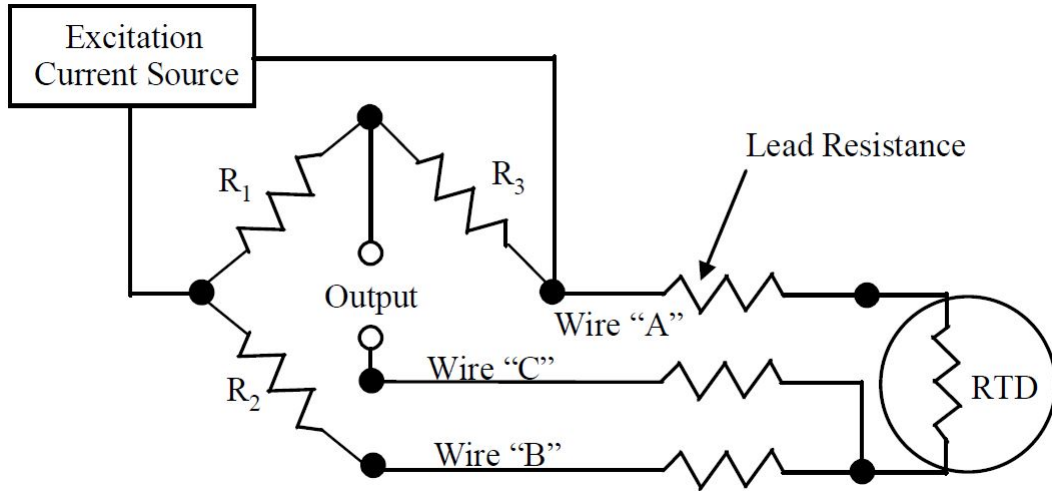


Figure 2.8: Three-wire RTD Wheatstone bridge configuration (Hughes, 2015).

The resistance to temperature relationship is described by the third order polynomial as follows:

$$R = R_0 + \alpha R_0 \left[T - \delta \left(\frac{T}{100} - 1 \right) \left(\frac{T}{100} \right) - \beta \left(\frac{T}{100} - 1 \right) \left(\frac{T^3}{100} \right) \right] \quad (2.1)$$

where

R = resistance at temperature T (Ω)

T = temperature ($^{\circ}\text{C}$)

R_0 = resistance at $T = 0$ ($^{\circ}\text{C}$)

α = constant (dependent on material type used)

$\beta = 0$ if $T \geq 0$, or 0.1 if $T \leq 0$

δ = constant (dependent on material type used)

2.4.3 Thermistor

A thermistor is a temperature-sensitive resistor and gets its name from an abbreviation of "*thermally sensitive resistors*" (Hughes, 1995). Hence, the thermistor's resistance changes as a function of temperature. Hughes (1995) defines the thermistor as being the temperature sensor that exhibits "*the largest value change with temperature of three major categories*".

A high temperature measurement resolution can be achieved by a thermistor since there is a large change in resistance per degree change in temperature. Hughes (2015) further shows that a typical $2000\ \Omega$ thermistor will have a $78\ \Omega$ change in resistance per $^{\circ}\text{C}$ change in temperature, whereas an RTD will only change $7.2\ \Omega$ per $^{\circ}\text{C}$ change in temperature. A thermistor that is connected to an adequate bridge circuit could provide a temperature resolution as low as $0.0005\ ^{\circ}\text{C}$. Figure 2.9 depicts the resistance-temperature relationship of the three different types of sensors discussed. The plot shows that the high accuracy of the thermistor is at the cost of linearity, whereas the RTD has great linearity but at the cost of lower accuracy. An RTD also has a larger operating range compared to a thermistor.

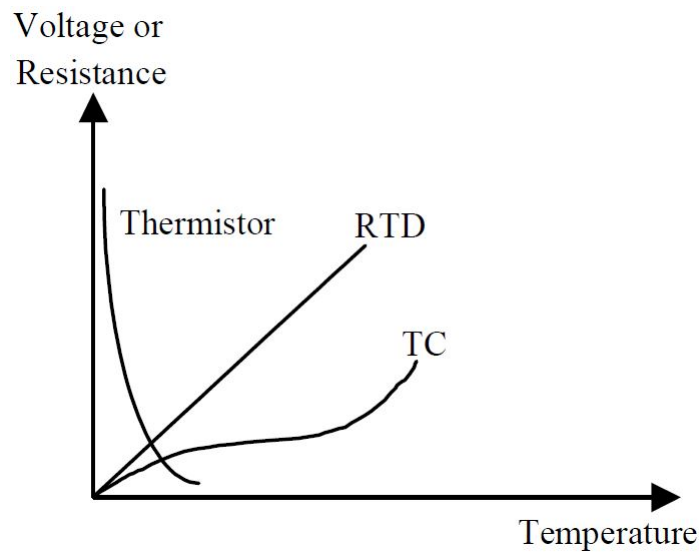


Figure 2.9: Temperature sensor curve shape comparison (Hughes, 2015).

The curve of a thermistor can be closely approximated by the use of the following equation called the Steinhart-Hart equation:

$$\frac{1}{T} = A + B \ln(R) + C(\ln(R))^3 \quad (2.2)$$

where

$$T = \text{temperature (K)}$$
$$R = \text{thermistor resistance } (\Omega)$$

and A , B and C are curve fitting constants. The non-linear curve shape of the thermistor can be overcome by using microprocessors that are widely available in the industry. The semiconductor nature of thermistors means that they are more prone to irreversible calibration errors as a result of excessive temperatures. Therefore, it should be ensured that a thermistor will always be used within its measurement capabilities. A thermistor is typically limited to a few hundred degrees Celsius. Thermistors can be manufactured in small packages, which allows for a quick response to temperature changes. Consequently, this makes thermistors fragile and costly.

2.5 Error measurement

2.5.1 Error classes

Barakat *et al.* (2000) suggest that there are four different types of errors that can be experienced during the use of a CMM. Each error has a different level of contribution, but each error is too significant to be ignored. Table 2.2 lists the various types of errors, each accommodated by a brief description.

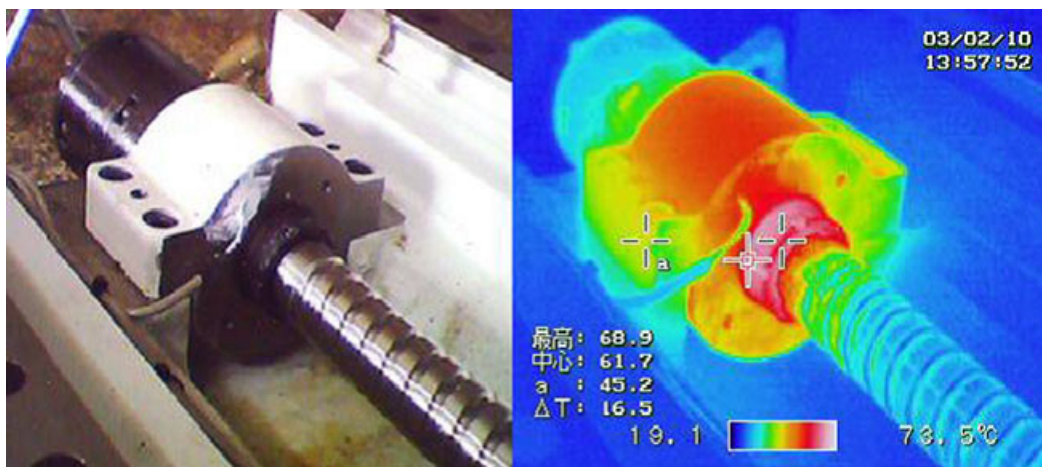
2.5.2 Characterizing the error

Simply measuring the change in ambient and CMM temperature would not provide the necessary information to develop a thermal error model. The thermal error model is required to provide probe displacement as a function of temperature, which will allow for displacement compensation. As a result, a need arose to develop a method of measuring the system displacement as a function of temperature. This section investigates and discusses previous efforts to compensate for thermal error.

Wu *et al.* (2012) proposed a method of measuring thermal expansion on a CNC machine by putting a capacitive sensor at the end of the machining mechanism. The capacitive sensor measures displacement for the varying temperatures and concentrations. Figure 2.10 uses a thermal imaging camera to illustrate the temperature effects that friction components may have. The study of Wu *et al.* (2012) used PT-100 RTD's to measure temperature changes on the structure.

Table 2.2: Four main types of CMM errors.

Geometric error	This type of error is because of the design of the system. These errors typically cannot be compensated for, they can only be corrected if done during the design phase of the CMM.
Kinematic error	The CMM's ability to reach an exact specified position from software commands. The error is the difference between the specified position and the actual position. Sources for this error are usually components such as bearings, gears, motors, leadscrews, etc.
Stiffness error	Assuming a CMM structure is rigid or has negligible deflection leads to stiffness error. Each component in the CMM structure is subject to deflection from its own weight and configuration.
Thermal error	Wu <i>et al.</i> (2012), Harris and Spence (2004), and Barakat <i>et al.</i> (2000) state that thermal errors are the largest contribution and can be between 60 % and 70 % of CMM related measurement errors. Ambient temperatures or structure heating from friction or electronic components cause the system to expand and contract, causing the probe to drift.

Figure 2.10: Leadscrew bearing housing showing temperature concentrations and gradients (Wu *et al.*, 2012).

In the study of Kruth *et al.* (2002), a different approach is used for characterizing structure displacement as a function of temperature. The first difference in this study is the use of a different temperature sensor. Instead of using an RTD, a thermistor was used. The study fails to document the specific make and model of the thermistor used, which lacks information such as the

measurement accuracy and resolution. However, it does reiterate benefits of using a thermistor such as a low cost, compact, long-term stability, and a high measurement range (for the thermistor used in the study) of -20°C to 200°C . This temperature range is sufficient since the reference operating temperature of a CMM is 20°C (ISO 1:2016).

Instead of using a capacitive sensor, Kruth *et al.* (2002) make use of a laser interferometer. Figure 2.4 provides an illustration of a laser interferometer in use with a CNC machine. A laser tracker can track displacement of an object in the x-, y- and z-direction. In this case, the object of concern is the CMM probe.

Pahk and Lee (2002) investigated thermal error compensation on a CNC machine similar to Wu *et al.* (2012), however, uses a set of gap sensors to measure the displacement of the spindle in the x-, y- and z-directions (Figure 2.11). Additionally, eight thermocouples are placed on the frame and near components that generate heat, two thermocouples are used to measure the ambient temperature, and 1 thermocouple is placed around the measurement jig, which contains the gap sensors. The thermocouple is placed on the measurement jig to ensure that measurements taken are not influenced by temperature. In the case that they are, it can also be compensated for.

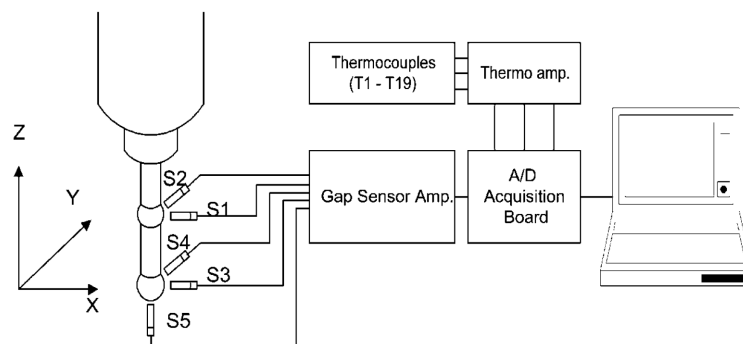


Figure 2.11: Gap sensor measurement jig and setup (Pahk and Lee, 2002).

Since capacitive sensors need to be in close proximity to the object being measured, it becomes difficult to measure the effects that a change in temperature on the structure may have on the probe while in different positions within the μCMM 's measurement range. Furthermore, this means only a limited displacement of the probe can be measured before it is out of range or touches the capacitive sensor. The laser interferometer that can track the movement of an object will allow the the probe to be placed in different positions while observing the effects of a change in temperature on the μCMM structure. Ad-

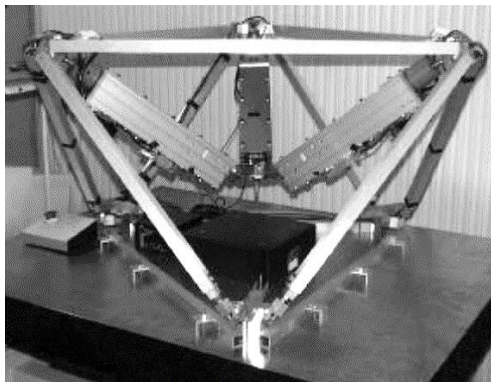
ditionally, the measurement range of a laser interferometer is higher than that of a capacitive or gap sensor.

2.6 The μ CMM

The μ CMM used in this study is provided by the Mechanical and Mechatronic Engineering Department of Stellenbosch University and was designed and built in 2014 by a PhD student, Dr A. Rugbani, as part of his thesis. Further development has been performed on the μ CMM, the most notable was the addition of two joysticks used for direct operation of the μ CMM. This section further discusses the μ CMM to provide a greater understanding of the layout and capabilities of the system. The information presented in this section is from Rugbani (2014).

2.6.1 System design

The μ CMM makes use of a parallel system, which is the alternative to the more common serial mechanism. The parallel system has each of its links connected to the probe, eliminating the accumulation of errors. An example of a serial mechanism and parallel mechanism for a CMM is provided by Figure 2.12 (a) and (b), respectively. Figure 2.13 further illustrates the concept of accumulation of error in a serial mechanism CMM, where material bending occurs in the upright column and the cantilever beam, which adds to the error.



(a)



(b)

Figure 2.12: (a) Parallel mechanism (Oiwa, 2000) and (b) a serial mechanism (Salsbury, 2016).

2.6.2 Coordinate system

The μ CMM uses a set of three motors and lead screws to control the vertical movement of each arm. Each motor can be controlled independently and are

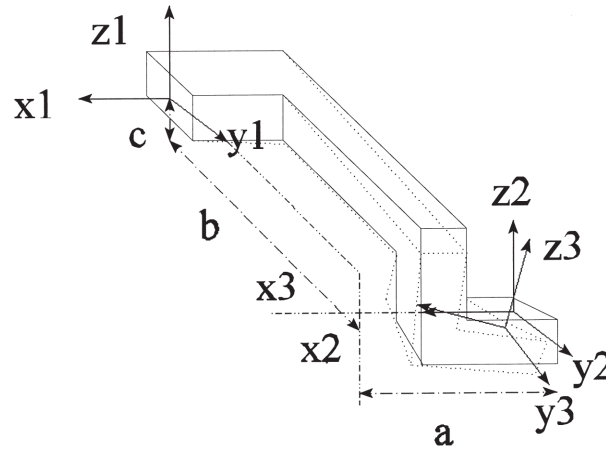


Figure 2.13: The deformation of a serial mechanism shown by the dotted lines (Barakat *et al.*, 2000).

equipped with encoders to measure rotation. This allows the vertical position to be calculated. The arms are connected to the vertical slider bracket by means of a spherical joint, and each arm length is measured by a separate laser. Each point on the μ CMM can be represented as a combination of variables x , y , and z and is further depicted in Figure 2.14. Motor commands, motor and laser data are sent and received using National Instruments LabVIEW, with data post-processing being performed in MATLAB[®] to give the probe position output. Spherical joints p_i where $i = a, b, c$ represent the spherical joints x -, y -, and z -coordinates, and where $i = 0$ represents the probe point.

2.6.3 Measurement abilities

The μ CMM has a specified accuracy of 10 μm with a three-dimensional measurement of range of 100 mm x 100 mm x 100 mm with use of a parallel manipulator. Figure 2.15 illustrates the reachable workspace of μ CMM, with the red area showing the 100 mm x 100 mm x 100 mm working volume desired. The machine uses a force contact probe method.

2.6.4 Kinematic model

The Direct Position Kinematic Model (DPKM) of the μ CMM is used to calculate the probe pose. Changing the probe pose is a matter of moving one motor at a time until the desired probe pose is reached. As a result, the probe pose can be easily calculated after each motor movement. Figure 2.16 illustrates how the probe pose is calculated after movement, and Rugbani (2014) further describes this by the equations below:

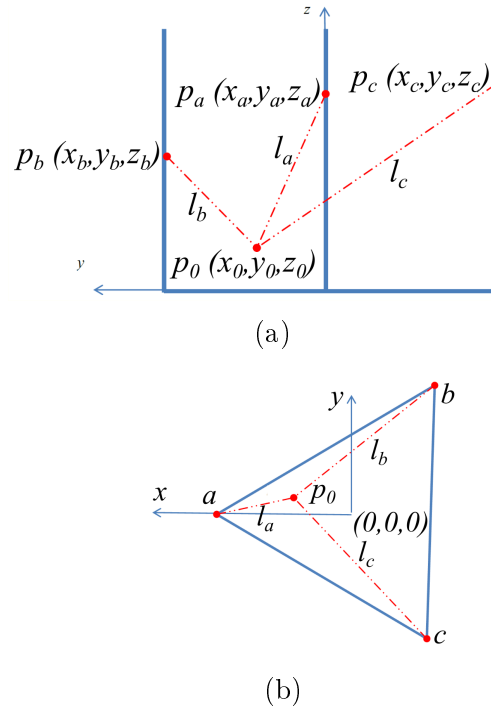


Figure 2.14: μ CMM coordinate system (a) top view and (b) side view (Rugbani, 2014).

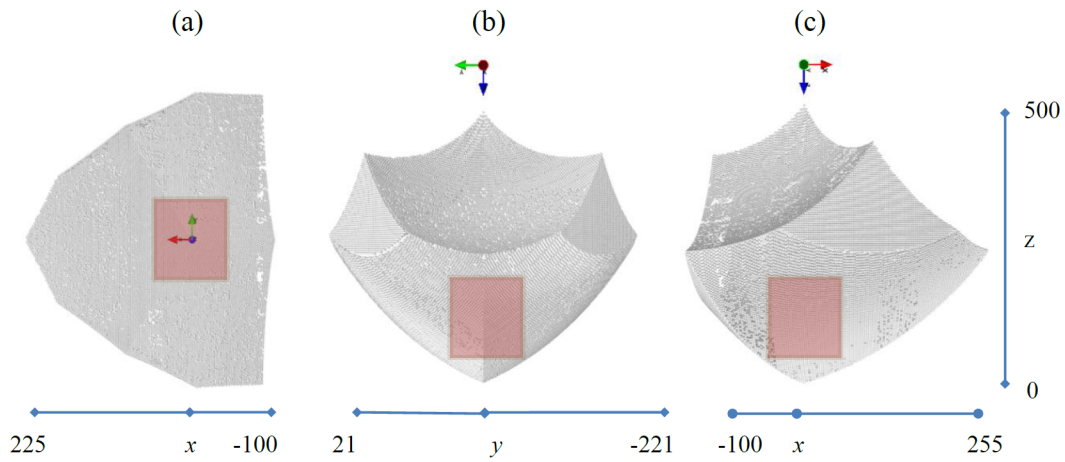


Figure 2.15: Showing the reachable workspace (a) top view, (b) front view, and (c) side view of the μ CMM (Rugbani, 2014).

$$z_{in} = z_i + dz_{in} \tag{2.3}$$

$$z_{ip} = dz_{ip} + z_i \tag{2.4}$$

$$dz_{in}^2 = d_{in}^2 - b_{in}^2 \quad (2.5)$$

$$dz_{ip}^2 = d_{ip}^2 - b_{ip}^2 \quad (2.6)$$

$$d_{in}^2 = l_i^2 + l_{in}^2 - 2l_i l_{in} \cos(\beta_{in}) \quad (2.7)$$

$$d_{ip}^2 = l_i^2 + l_{ip}^2 - 2l_i l_{ip} \cos(\beta_{ip}) \quad (2.8)$$

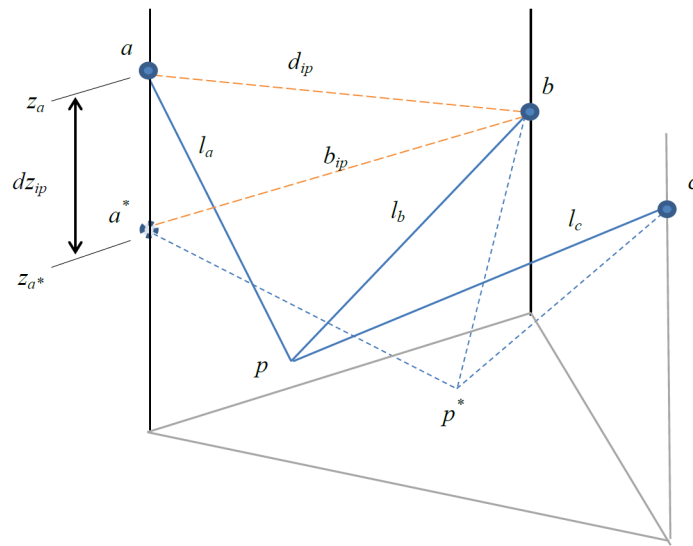


Figure 2.16: Change in pivot point height as probe is displaced (Rugbani, 2014).

The z value indicates the motor height in the z -axis. Subscript i denotes the index of the pivot point of the i^{th} joint, $i = [a, b, c]$ with i starting at a going clockwise as viewed from above the μCMM . in and ip represent the pivot points of the next and previous points to the current motor that was moved, respectively. dz represents the height difference between the motor moved and the stationary motor, whereas d gives the absolute change in height of the motor moved. b_{in} is the smallest distance between pivot points. β is the angle between two arms of the tetrahedron, further explained in Chapter 3. Furthermore, Equations 2.9-2.11 are used to calculate the probe pose once the z values are calculated.

$$x = A - By \quad (2.9)$$

$$y = \frac{-v \pm \sqrt{v^2 - 4uw}}{2u} \quad (2.10)$$

$$z = F - Dy \quad (2.11)$$

Where,

$$A = \frac{\left(\frac{c_{in} - c_i}{2z_{in}} - \frac{c_i - c_{ip}}{2z_{ip}} \right)}{\left(\frac{x_i - x_{ip}}{z_{ip}} - \frac{x_{in} - x_i}{z_{in}} \right)} \quad (2.12)$$

$$B = \frac{\left(\frac{y_i - y_{in}}{z_{ip}} - \frac{y_{ip} - y_i}{z_{in}} \right)}{\left(\frac{x_i - x_{ip}}{z_{ip}} - \frac{x_{in} - x_i}{z_{in}} \right)} \quad (2.13)$$

$$D = \frac{y_{in} - y_i}{z_{in}} - B \frac{x_{in} - x_i}{z_{in}} \quad (2.14)$$

$$F = A \frac{x_{in} - x_i}{z_{in}} - \frac{c_{in} - c_i}{2z_{in}} \quad (2.15)$$

$$u = 1 + B^2 + D^2 \quad (2.16)$$

$$v = 2DF + 2x_i B - 2AB + 2y_i \quad (2.17)$$

$$w = A^2 + F^2 - 2x_i A - c_i \quad (2.18)$$

Additionally,

$$x_i = g_i + h_i z_i \quad (2.19)$$

$$y_i = m_i + n_i z_i \quad (2.20)$$

Equations 2.19 and 2.20 represent vertical error that the spherical joints travel on. With g and m equal to the intercept and h and n equal to the slope.

$$c_i = l_i^2 + x_i^2 + y_i^2 + z_i^2 \quad (2.21)$$

2.7 Existing parallel CMM

Renishaw[®] designed and manufactured a parallel CMM, which has "*already changed the thinking of thousands of production engineers, making it their new gauge of choice*" (Renishaw[®], 2018). Figure 2.17 shows the Renishaw[®] Equator[™] 300 along with its working area in normal and extended height.

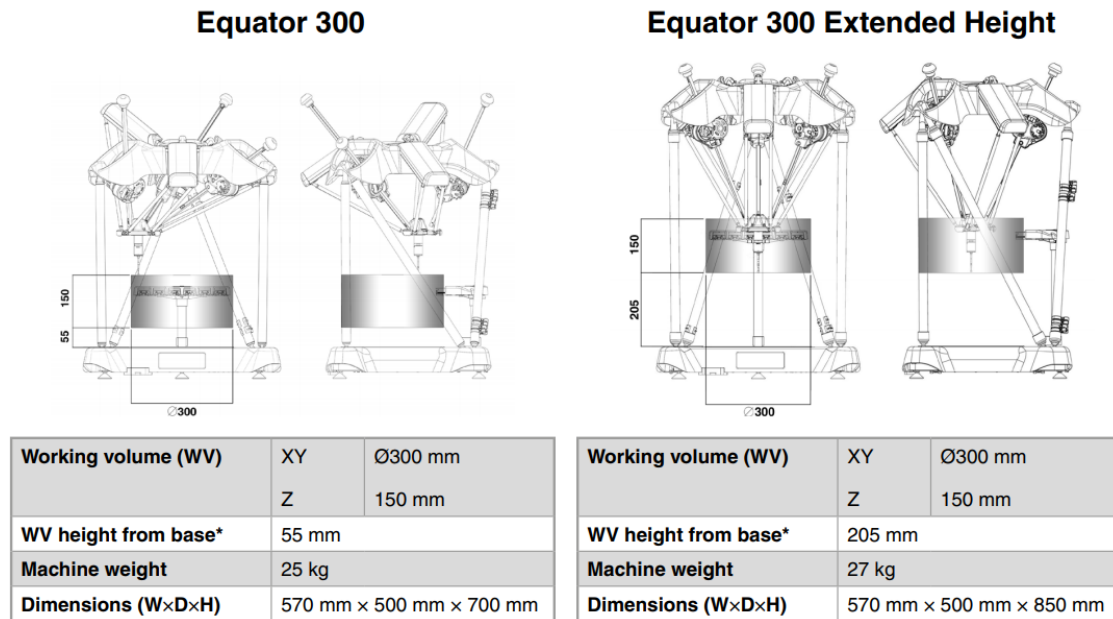


Figure 2.17: The Equator[™] 300 parallel mechanism CMM with working volume and dimensions (Renishaw[®], 2016).

Renishaw[®] does not provide much technical information on how the Equator[™] 300 works, however, simple deductions can be made from the figures provided in Renishaw[®] (2018) and Renishaw[®] (2016). Three sets of

two arms are placed on spherical joints at either end. On the top end of the arm set is a cantilever actuator that rotates from a centre point fixed on the structure with rotation being about the horizontal axis. On the bottom end is where all the arms connect to the same part that the probe is attached to. This movement allows the probe to stay vertical at all times while within the work area. Each arm set has a third additional arm placed directly opposite, which is attached to the probe plate and extends out the top of the CMM. The third arm is used as a linear actuator with a linear encoder to measure movement in tangent with the possibility of a rotary encoder placed on the cantilever arm actuators. Table 2.3 provides some additional information on the specifications.

Table 2.3: EquatorTM 300 specifications. Adapted from: Renishaw[®] (2016).

Comparison uncertainty	± 0.002 mm
Scale resolution	0.0002 mm
Probe type - touch-trigger	Renishaw 3-axis TP20 kinematic touch-trigger
Operating temperature	+10 °C to +40 °C

2.8 Literature conclusion

The literature review has shown the necessary methods of performing measurements by which the error model will be tested, and these are provided by the ISO standards. The literature review also provided the necessary information for making a decision on the equipment to use for error measurement.

Methods of measuring the accuracy of the existing μ CMM were described. The literature also describes methods of increasing accuracy with calibration. Various calibration methods were provided as well as the equipment necessary for calibration.

Although thermocouples are the most versatile devices used for measuring temperature (Hughes, 1995), thermocouples are not very accurate and degrade over time (Kerlin and Johnson, 2012). The advantage of using an RTD for measuring temperature is that they are the most linear temperature measurement devices. However, thermistors are loosely defined as "the most sensitive" temperature measurement devices. In this case, this is a desirable characteristic, since the temperature measurement range is small, which will ensure that the measurement range is within a linear region of the thermistor. Furthermore, the thermistor will provide a greater temperature measurement resolution.

Chapter 3

The μ CMM, user interface and software

3.1 Machine description

This section describes the components of the μ CMM to provide a better understanding of the machine's working principles. The μ CMM is comprised of two main structures: (1) the frame, shown in Figure 3.1 (a), and (2) the tetrahedron, shown in Figure 3.1 (b).

A Zaber[®] T-LSR motor is attached to each upright column of the μ CMM frame. The T-LSR motor series consists of a lead screw, stepper motor with encoder, and a lead screw bracket. The motors are connected to each other through a serial communication cable. Of the three motors, a primary motor can be selected. As a result, only one serial connection, between the computer and motors, is required for communication. The serial communication involves control of the motors and feedback of encoder values, which is provided in an RS422 serial format. One advantage of a single connection is that the motors can operate either simultaneously or individually. The lead screw is then rotated by the stepper motor, which displaces the lead screw bracket vertically with a travel of 450 mm.

A spherical joint—also referred to in this text as a pivot point—is attached to each of the three lead screw brackets on the μ CMM frame. A spherical joint is imperative to the movement of the μ CMM, otherwise, the probe would be limited to a vertical displacement only. An in-depth description of the need for a spherical joint can be found in Rugbani (2014). In addition, the spherical joint is also attached to a bracket that houses a laser and a linear slider with ball bearings (Figure 3.2). The linear slider bracket moves along the direction of the tetrahedron arm while the laser measures its displacement. A reflector

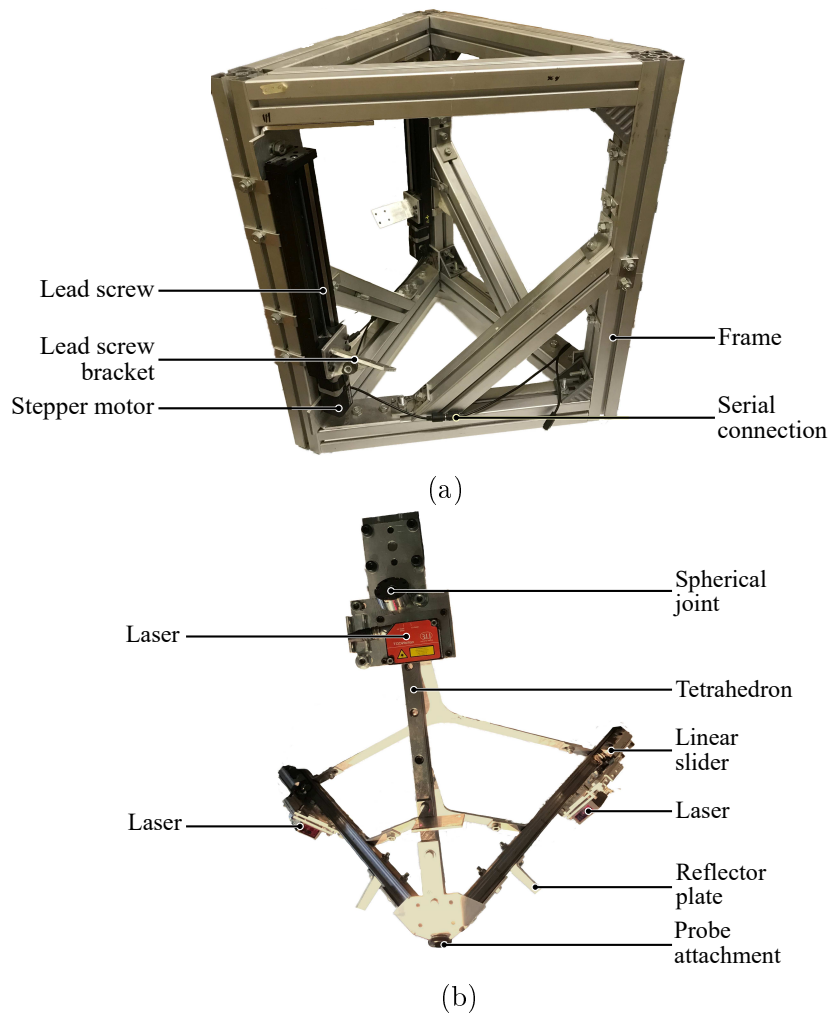


Figure 3.1: (a) The μ CMM frame structure and (b) tetrahedron.

plate (Figure 3.1) is placed on the tetrahedron arm, which is a reference point for the laser measurement as the linear slider bracket displaces relative to the reflector plate. The lasers used are microEpsilon[®] lasers with a 200 mm measurement range and an accuracy of 50 μ m.

Figure 3.3 shows the Renishaw[®] TP8 force contact probe used for the μ CMM of this project. The probe can be adjusted to have a trigger force between 0.1 N and 0.3 N. The stylus fitted to the probe is 21 mm long with a 2 mm diameter ball tip. When the probe makes contact with an artefact, and the contact force exceeds the trigger force, the probe sends a signal notifying that contact was made.

Rugbani (2014) provided calibrated values for the system parameters, which are used in Equations 2.3 to 2.20 to calculate the probe coordinate.

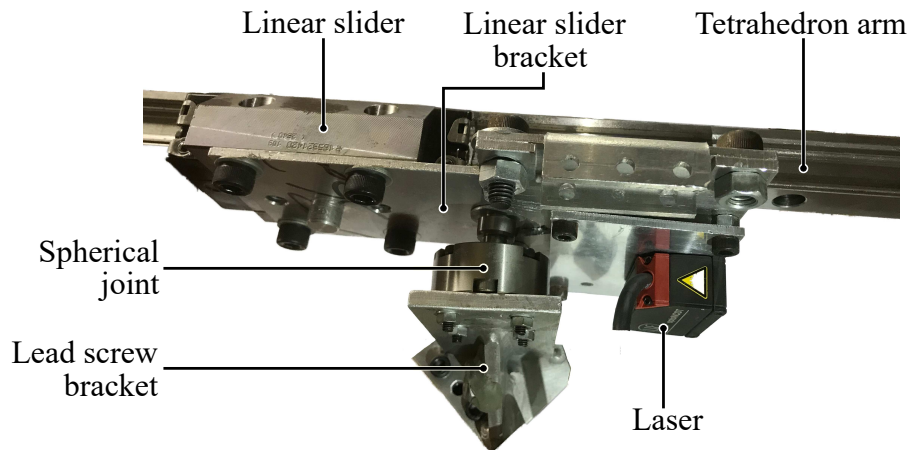


Figure 3.2: Spherical joint attached to a bracket of the linear slider.

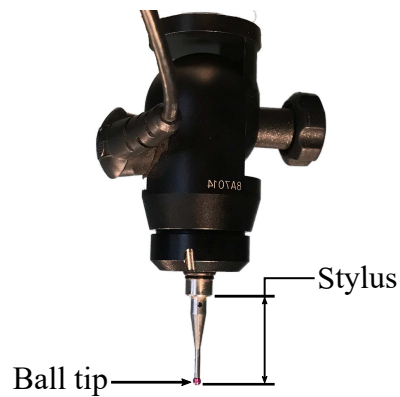


Figure 3.3: Probe used for the μ CMM.

These values are shown in Table 3.1 and are referred to as the expected values when a new calibration is performed. The dead distance d of each arm is the distance between its spherical joint and the probe tip, minus the laser measurement. Angle β_{in} is the angle between two tetrahedron arms and b_{in} is the distance between the pivot points of two arms in the xy -plane.

Table 3.1: Original calibrated parameters of Rugbani (2014).

Arm	d_i	b_{in}	β_{in}
a	294.02	460.10	68.51
b	289.90	458.89	68.73
c	289.40	458.09	68.36

3.2 Machine characterization

The μ CMM is controlled through LabVIEWTM, which is a modular programming platform used as the link between the μ CMM equipment and the computer used. This allows for the acquisition of data and the crucial controlling of the μ CMM. The relationship between data transfer and how each component interacts is shown in Figure 3.4.

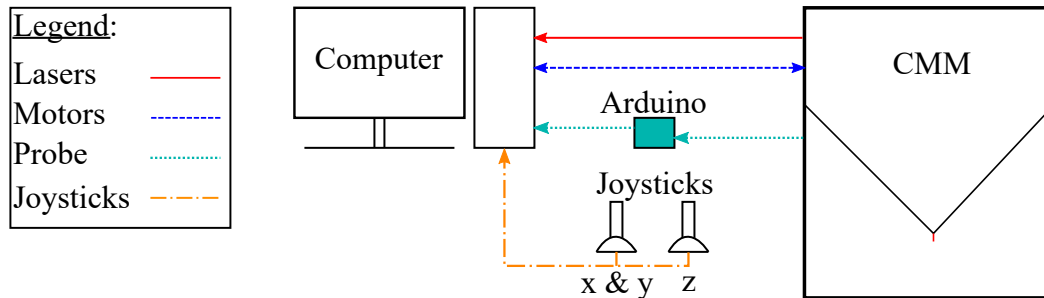


Figure 3.4: Hardware interaction on the μ CMM.

van Zyl (2016) modified the original software presented by Rugbani (2014). The main modification was the addition of LabVIEWTM software structure to incorporate and allow for the operation of the μ CMM with two joysticks: one controlling movement in the x-y plane, and another controlling movement in the z-direction. The state of the μ CMM software meant many challenges were encountered during initial familiarization with the machine. Above all, the software presented reliability issues where the joysticks would not be registered and the probe state was incorrectly communicated. As a result, a significant amount of time was required to debug and repair the code to provide a reliable system, aimed at making the accuracy measurement and compensation process efficient and reliable.

3.3 Joysticks

The main problem experienced with the joysticks was the software's failure to register the joystick. Initially, the problem was thought to be because this project uses a different computer to run the UI. However, research on National Instruments forums and joystick examples showed that a computer can register connected joysticks in different serial indexes. As a first approach, a means of finding the exact index value of each joystick was needed. Once this was achieved, additional functionality that allows the setting for each joystick index was required. The solution was to use a LabVIEWTM function that allows the user to step through the serial index positions (Figure 3.5) while showing the device name and type in each index. Finally, once the index position of each

joystick has been identified it can be set in the UI and lets LabVIEW™ know which serial devices are the joysticks.

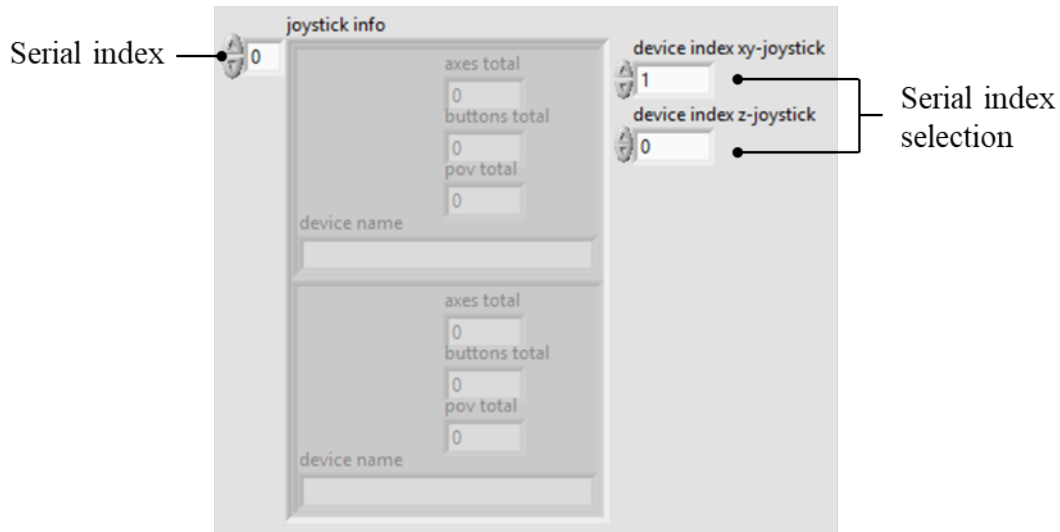


Figure 3.5: Serial device identification and selection.

Adding the functionality depicted by Figure 3.6 to the existing LabVIEW™ code was achieved by the addition of four LabVIEW™ functions. These functions are contained within the figure depicted, along with the addition of a flat sequence structure. The flat sequence structure helps when it is desired to complete one task before the other task is executed, and can be seen by the light grey vertical lines in Figure 3.6.

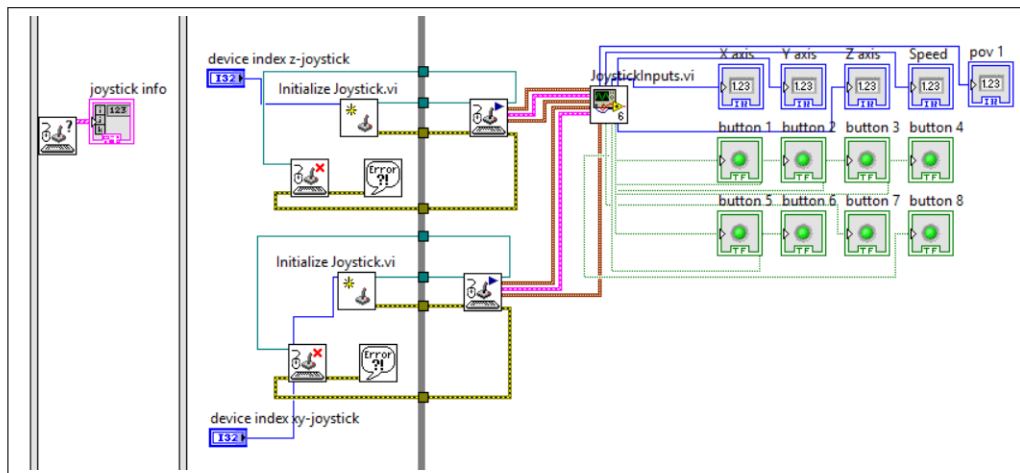


Figure 3.6: LabVIEW™ block diagram for joystick index identification, setting and operation.

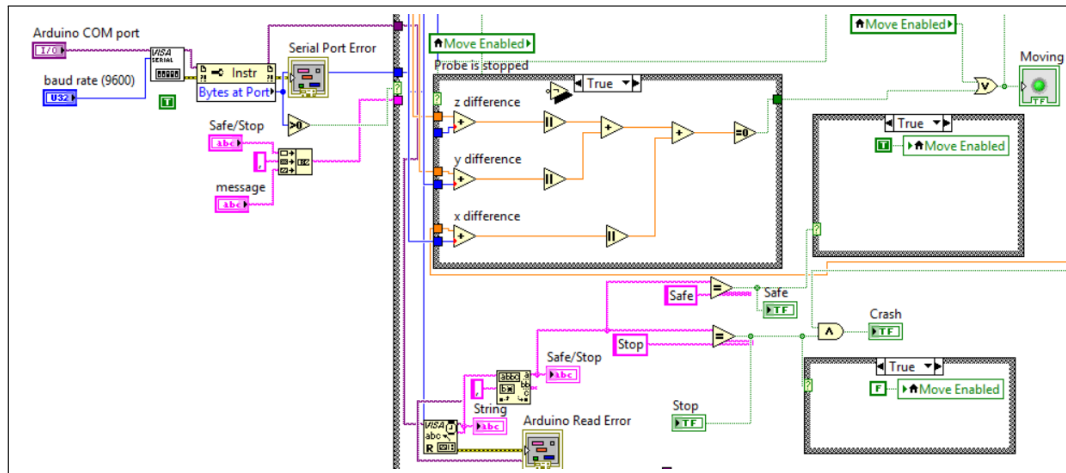
3.4 Arduino[®] and LabVIEW[™] interface

The probe state is communicated to LabVIEW[™] via a serial port, which is connected to a microcontroller unit called an Arduino[®]. The Arduino[®] acts as a mediator and processes the state of the probe through code compiled on it, which then outputs one of two strings: "safe" or "stop". If the probe has made contact with an object, then the "stop" string is sent to LabVIEW[™], and if the probe is not in contact a "safe" string is sent. This approach restricts the CMM from moving once object contact has been made unless movement is away from the object.

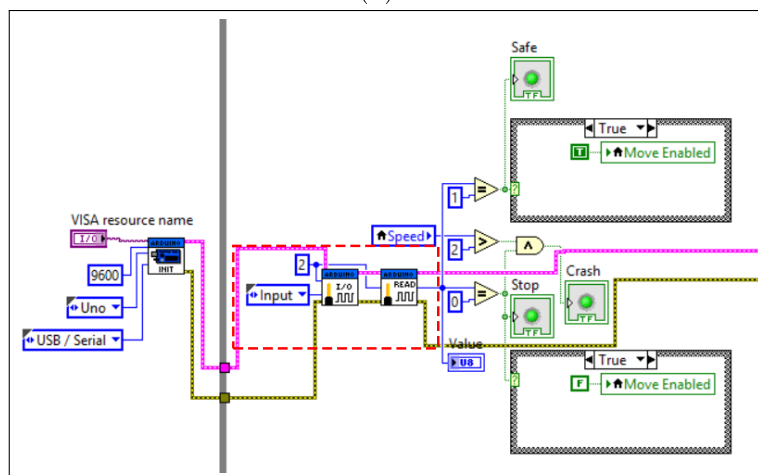
As previously mentioned, this approach of communicating the probe state to LabVIEW[™] was ineffective, since LabVIEW[™] received random characters instead of a "safe" or "stop" string. This meant that, even though the joysticks had been successfully implemented, the machine would not be able to move because it resides in an unknown state. Initial attempts to rectify this saw that the first few sets of strings were correctly sent and received, but the string state would either not change or it would turn into random characters. Originally, it was thought to be either a problem of incorrectly matched BAUD rates between the Arduino[®] and LabVIEW[™], no use of start and stop bits to indicate the start and end of the string, or serial buffer memory overflow or leakage. When these possible problems were investigated and corrective measures were taken, the output yielded the same results.

For this reason, a different approach of reading the probe state was attempted. The goal was to turn the Arduino[®] into a data acquisition device (DAQ), which would allow the analogue input port of the Arduino[®] to be utilized, the value read by the port to be sent to LabVIEW[™] via serial communication, and all processing of source code to be performed in LabVIEW[™]. For the most part, National Instruments provide code for an Arduino[®] that makes it function as a DAQ, but attempts at using the provided code were unsuccessful. However, after studying the code, it was noticed that the provided code did not support the specific version of the Arduino[®] Microprocessor used. Hence, the code was modified and implemented with successful results. Figure 3.7 (a) shows the original LabVIEW[™] block diagram, whereas Figure 3.7 (b) shows the new, successfully working, block diagram.

The LabVIEW[™] code (Figure 3.7 (b)) was designed to allow the user to set the type of Arduino[®] Microcontroller, BUAD rate, serial port location, and the type of communication used; forming part of the initialization process. The section outlined by the red dashed block, in Figure 3.7 (b), is where the pin being used on the Arduino[®] can be selected and is set as an input since a voltage is read from the Renishaw[®] probe system. The pin state is continuously read and processed as boolean values of either 1 or 0, indicating



(a)



(b)

Figure 3.7: (a) Original and (b) new LabVIEWTM block diagram for Arduino[®] communication.

the probe is not in contact or the probe has made contact with an object, respectively. The benefit of the new code structure is that the probe state can be read accurately and instantaneously. What is important to note in Figure 3.7 (a) is the high complexity of the original code used. This code uses more processing time, is harder to debug for errors, and can contribute to the likelihood of making errors. Comparatively, Figure 3.7 (b) shows the more efficient code for using the Arduino[®] as a DAQ, with many safeguards in place to ensure a consistent and reliable operation.

3.5 Machine settling time

During movement of the machine and making an attempt at measuring the laser values, it was noticed that these values would not stay fixed as soon as the μ CMM stopped moving, rather, they would take some time to settle. The first matter was to determine the settling time of the machine. The motors were moved in varying combinations with varying distances and at varying speeds, while being timed with a stopwatch. What was important was to find the maximum measured settling time so that a confidence was given that all settling times were covered. Figure 3.8 shows that the maximum measured settling time for each laser, as a function of motor encoder counts, was approximately 1.3 seconds.

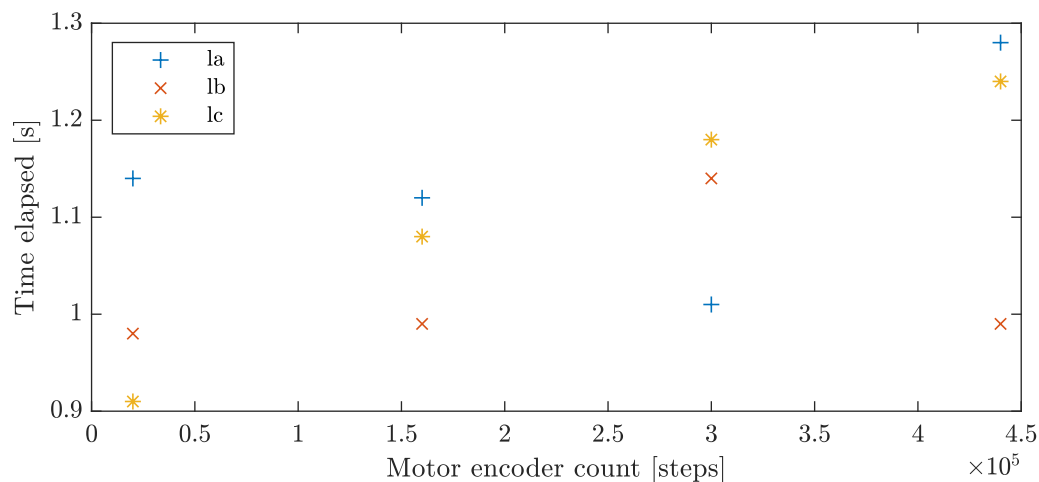


Figure 3.8: Motor encouder counts vs. settling time.

A loop was incorporated into the code which allowed for a count variable to be made whilst in a "while loop", which would normally reset any variable. This count variable was set to the maximum measured settling time along with a factor of 2 to allow for any error. Moreover, the idea of not putting in a factor that was too high was to increase the μ CMM's measurement speed of a specimen.

3.6 Machine measurement

The measurement data is returned from LabVIEWTM and saved into a .csv file. The format of how the .csv file is arranged is provided in Table 3.2. The risk of the machine not settling within the required time is still a reality. Consequently, the laser data is saved at the start and at the end of a motor movement and is indicated with a 1 or 0, respectively. Similarly, the probe touch is indicated with a 1 for probe contact with an object and 0 for no

contact. The motor moved column saves the motor moved which is either 1, 2 or 3. The fourth column saves the last direction moved by the joystick, which is either negative or positive 1. As a result, this data can be used for whether the motor was moved up or down. Additionally, with probe contact made and the last motor movement direction known, probe movement is only allowed if the motor moves in the opposite direction. Thus, only allowing probe movement away from the artefact.

Table 3.2: Format of saved data from μ CMM measurements.

start/stop (1/0)	probe	motor moved	direction	laser _a	laser _b	laser _c
---------------------	-------	----------------	-----------	--------------------	--------------------	--------------------

Chapter 4

Thermal error model

An initial goal for developing an error model was to create a model that would be simple to derive and implement. The development of such a model was to verify that the use of an error model shows measurement accuracy improvement, even a small improvement. This section provides an overview of the methods and assumptions used for deriving a prototype error model. However, the method discussed was not the final method used, as this study aimed to further investigate more effective error models.

4.1 Simple model derivation

The first approach was to develop a linear error model as it was believed to be the simplest type of model to derive. This model would not need excessive amounts of processing power to execute calculations. However, the geometric layout of the μ CMM meant that deriving a linear error model, without the use of trigonometry, would be unnecessarily challenging or would require too many assumptions and simplifications. Therefore, part of the probe position was related to an angle between the μ CMM arms and the upright columns, which support the motors.

Assumptions were made when developing the error model. The assumptions utilized were motivated to show that the integrity of the error model was not compromised. An underpinning assumption to this particular error model was that the expansion and contraction are assumed to be linear with no bending in the structure. The thermal error model was divided into two sections to simplify the development process, namely: (1) vertical error (error in the z-direction); and (2) planar error (error in the x-y plane). In order to correctly characterize the error model, a thermal expansion model for the μ CMM structure must first be derived.

4.1.1 Structure expansion

The error model that was derived, determines the probe position error relative to the μ CMM structure. This requires that the dimensions of the μ CMM are known and defined in the Cartesian coordinate system. However, the μ CMM structure is not immune to temperature changes. Consequently, the ever-changing dimensions of the structure must be determined before the error is calculated. The aim was to equip the machine with strategically placed temperature sensors on the tetrahedron and structure to measure the temperature. Then, through the use of a linear thermal expansion equation:

$$\Delta l = \alpha l_0 \Delta T, \quad (4.1)$$

the change in length of the μ CMM structure can be calculated with the different variables defined as:

$$\alpha = \text{Thermal expansion coefficient } (10^{-6} \frac{m}{mK}),$$

$$l_0 = \text{Original length before temperature change } (m),$$

$$\Delta T = \text{Change in temperature } (K).$$

The vertical error of the probe is directly attributed to a change in length in the z-direction of the structure and was further evaluated in Section 4.1.2. The planar error—error in the x-y plane—is calculated relative to the three main corners of the tetrahedron. Therefore, the x- and y-coordinates of corners a, b, c are to be defined as a function of the change in length of the tetrahedron sides. Figure 4.1 (a) illustrates the μ CMM tetrahedron structure from the top view, including the coordinate variables of each corner. To make calculations possible, structure joints are assumed to be pin connected. Additionally, the base of the structure is assumed to be fully constrained in the z-direction.

The pinned boundary condition at point A restricts translation in the x- and y-direction but allows rotation about the z-axis (perpendicular to the page). The sliding boundary condition at point B assumes frictionless movement in the x-direction and restricts movement in the y-direction. In combination, these boundary conditions mean that the μ CMM, as a whole, cannot translate or rotate on or about any axis, while still allowing each side of the structure to freely expand or contract. This holds true irrespective of whether

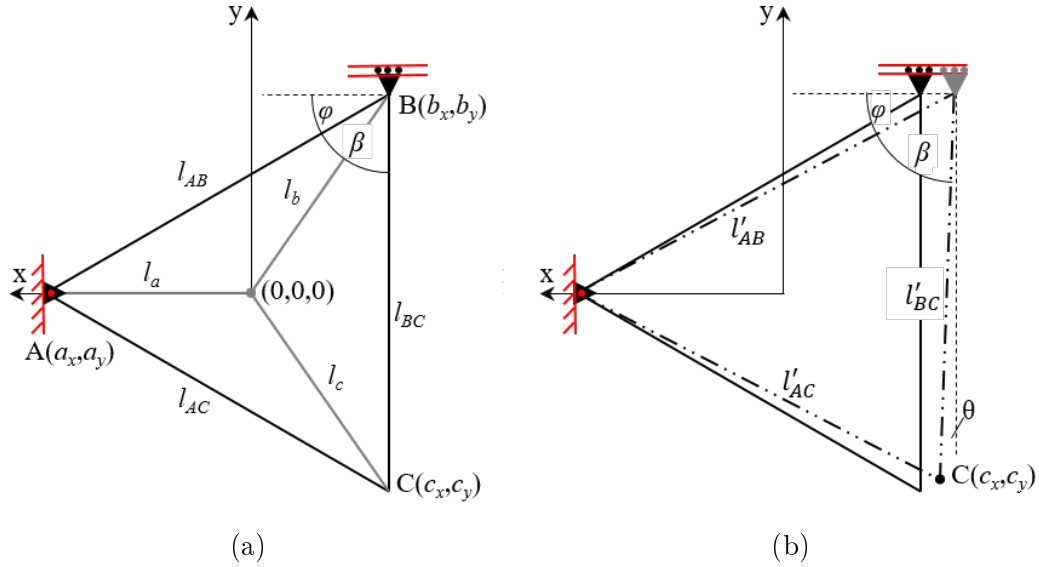


Figure 4.1: Machine top view with boundary conditions on (a) original and (b) deformed μ CMM.

the sides expand equally or unequally. The resulting boundary conditions allow for three variables to be fixed, which are the values in red print in Table 4.1. Additionally, Table 4.1 shows the coordinates of each point for the unexpanded state of the μ CMM, when each side length is 650 mm in length.

Table 4.1: Fixed and variable coordinate points for the μ CMM.

Coordinate variables	Value [mm]
(a_x, a_y)	$(-375.27, 0.00)$
(b_x, b_y)	$(187.63, 325.00)$
(c_x, c_y)	$(187.63, -325.00)$

With three variables subject to change, a set of at least three equations is required to calculate their positions as a function of temperature. Variable b_x is calculated by considering Pythagoras and using the three known, and fixed, variables:

$$l'_{AB}{}^2 = (a_y - b_y)^2 + (b_x - a_x)^2, \quad (4.2)$$

where,

$$l' = l + \Delta l. \quad (4.3)$$

Then isolating variables b_x yields:

$$b_x = \sqrt{(l'_{AB})^2 - (a_y - b_y)^2} + a_x. \quad (4.4)$$

Calculating variables c_x and c_y is a more involved process. Consider point B and point C in arbitrary positions (Figure 4.1 (b)), with the side lengths known from the measured temperature and Equation 4.3. The angle β can then be calculated using the Cosine rule:

$$\begin{aligned} l'_{AC}{}^2 &= l'_{AB}{}^2 + l'_{BC}{}^2 - 2l'_{AB}l'_{BC} \cos(\beta), \\ \beta &= \cos^{-1} \left(\frac{l'_{AC}{}^2 - l'_{AB}{}^2 - l'_{BC}{}^2}{2l'_{AB}l'_{BC}} \right). \end{aligned} \quad (4.5)$$

Similarly, ψ can be calculated with the known fixed variables and the calculated variable b_x :

$$\psi = \sin^{-1} \left(\frac{b_y - a_y}{l'_{AB}} \right). \quad (4.6)$$

With ψ and β known, θ can be calculated:

$$\theta = \psi + \beta - 90^\circ. \quad (4.7)$$

This allows for variables c_x and c_y to be calculated from the opposite and adjacent lengths to angle θ as follows:

$$c_y = b_y - l'_{BC} \cos(\theta), \quad (4.8)$$

$$c_x = b_x + l'_{BC} \sin(\theta). \quad (4.9)$$

4.1.2 Vertical error

The vertical error of the probe is assumed to be only influenced by the change in length of the three vertical support columns. Each vertical support column includes a leadscrew and leadscrew housing that is directly connected to a

motor, which is a heat source to the system. Additionally, the friction between leadscrew sliders and bearings is also believed to be a source of heat generation that affects vertical expansion. Contributions to the vertical probe error from elongation of μ CMM arms are considered negligible since the elongation has a vertical and horizontal component. As a result, the vertical component from the arm is assumed to be significantly smaller than the direct elongation to the vertical support columns. A diagram of the μ CMM is depicted in Figure 4.2, which shows the probe in an arbitrary position. Variables z_i with $i = a, b, c, p$, represent the vertical displacement of arms a, b, c, and probe p, respectively.

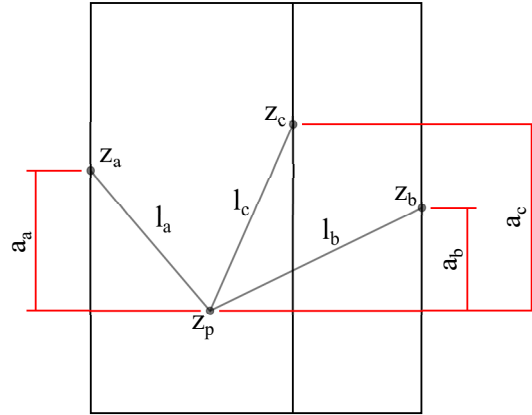


Figure 4.2: μ CMM sideview with arbitrary probe position.

The vertical displacement of point z_i is provided by the operating software of the μ CMM, which is calculated by using laser measurement data into Equations 2.3 to 2.8. Point z_p is determined by the z component of the final probe position output of the operating software. Distances a_i , are defined by the following equation:

$$\delta_i = z_i - z_p, \quad i = a, b, c. \quad (4.10)$$

Points z_i will move, in the vertical direction, according to a change in temperature on the columns—the resulting shift denoted by z_i' . Applying Equation 4.1 to each vertical column of the tetrahedron, point z_i' can be calculated as follows:

$$z_i' = z_i + \Delta l. \quad (4.11)$$

With the new position of each vertical point calculated, compensation can be made on the probe's vertical distance. It should be noted that each point

will not necessarily move in the same direction or with the same magnitude, due to uneven structure heating. Therefore, using an approach of adding the direct heat-related elongation to the probes vertical position, would be inaccurate. Instead, another approach was used where it is assumed that, since the structure is so large, compared to the small work envelope (100 mm x 100 mm x 100 mm), that the probe moves as an average between each point:

$$z_p' = \frac{z_a' + z_b' + z_c'}{3} \quad (4.12)$$

4.1.3 Planar error

The planar error of the probe is the error contribution in the xy-plane. With the assumptions made in Section 4.1.2 and the assumption that the structure does not bend under thermal change, the error is assumed to be only a function of thermal change on the μ CMM's arms. Since each arm is on a linear slider, and the probe is connected via a fixed bracket, any expansion on an arm will result in a smaller distance being measured by a laser because the arm expands away from the probe. With the derivation of the planar error model, the probe is considered to be in an arbitrary position, illustrated by Figure 4.3.

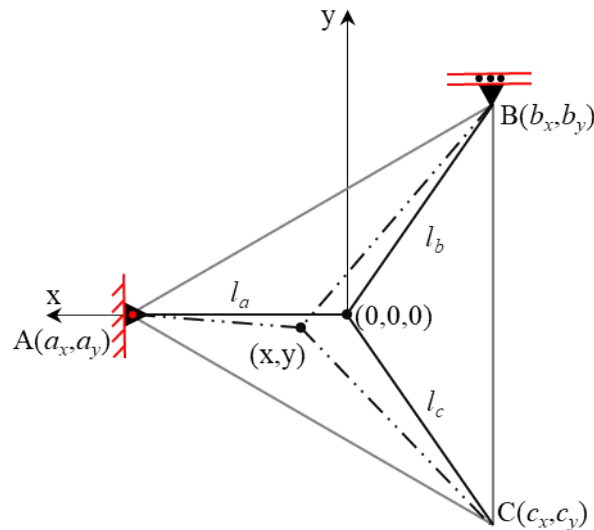


Figure 4.3: Diagram of the displaced μ CMM.

A similar approach to that used in Section 4.1.2 is used where, the change in x- and y-coordinate for each arm as a result of temperature change, is considered. An average of the displacement for each arm is then taken as the final probe point. First, each arm is isolated, with its length treated as the

hypotenuse that forms a right-angle triangle in Figure 4.4. Then, angle α_k is calculated with the following equation:

$$\alpha_k = \cos^{-1} \left(\frac{|k_y - y|}{\sqrt{(k_x - x)^2 + (k_y - y)^2}} \right) \quad (4.13)$$

where k represents arm a , b , or c .

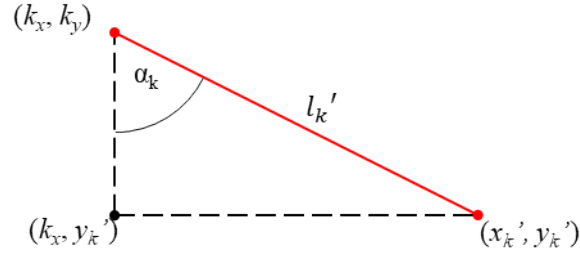


Figure 4.4: Arm l_k forming a triangle to calculate new x- and y-points.

With angle α_k , arm length l_k , and change in length of l_k —denoted by l'_k —known, the new x- and y-coordinates for each arm, x'_k and y'_k , can be calculated using Pythagoras:

$$|k_x - x'_k| = \frac{\delta_k}{l'_k} \sin(\alpha_k), \quad (4.14)$$

$$|k_y - y'_k| = \frac{\delta_k}{l'_k} \cos(\alpha_k), \quad (4.15)$$

Once x'_k and y'_k are known, the final planar probe position can be calculated using Equation 4.16 and 4.17.

$$x_p' = \frac{x_a' + x_b' + x_c'}{3} \quad (4.16)$$

$$y_p' = \frac{y_a' + y_b' + y_c'}{3} \quad (4.17)$$

4.2 Error model results

The thermal error model was used to help understand the effects that a change in temperature has on the μ CMM and its probe coordinate. The model was first used to determine an arbitrary probe coordinate at the standard temperature of 20 °C, according to ISO 1. Then a temperature change of +1 °C was implemented into the model to observe the effects. Table 4.2 shows the displacement for the x- and y-coordinate is 0.02 mm and the change of the z-coordinate is 0.01 mm.

Table 4.2: Thermal error model probe coordinate calculation.

Temperature	$x[mm]$	$y[mm]$	$z[mm]$
20 °C	80.007	-98.917	126.673
21 °C	80.029	-98.937	126.682
Difference	0.022	0.020	0.011

Considering that the accuracy of a laser on the μ CMM is only 0.05 mm and a 1 °C change in temperature gives a probe displacement that is less than the μ CMM laser accuracy, the effects are considered negligible given the other inaccuracies of the μ CMM. Hence, focus was placed on investigating the accuracy of the μ CMM. Since a decision was made not to use thermal error compensation, the assumptions made for the thermal error model were seen as accurate. If the method of thermal error compensation was used and the simple error model was verified, then the assumption of that the system does not bend as a result of thermal change, would have been removed.

Chapter 5

Accuracy test

Investigating the current state of the μ CMM's accuracy gave crucial insight into the state of this project. In addition, a better idea was given as to what steps are to be followed in pursuit of improving the current accuracy. A similar approach used by ISO 10360 for acceptance and re-verification tests, whilst adhering to its requirements, will be used for this project and described in this section.

5.1 Calibrated measurement specimen

It was important to find a calibrated measurement specimen that could sufficiently test the accuracy of the μ CMM, according to ISO 10360. Moreover, the required test specimen had to cover the measurement range of the μ CMM without exceeding its bounds. Therefore, measuring a 100 mm gauge block was found suitable for initial testing. Once the accuracy was verified, a more complex specimen could be measured for further verification. Additional reasons for utilizing a gauge block instead of specimens from ISO 10360-2 was the availability of the gauge block specimen as a calibrated artefact. The gauge block would also allow for a simpler straight line post-processing calculation to be made. Straight line post-processing rules out the possible error that could be incurred from post-processing of measuring more complex shapes. Therefore, sole focus is placed on any possible measurement errors of the μ CMM.

5.2 Experimental procedure

The general procedure used to measure a test specimen is described below and is further depicted in the diagram of Figure 5.1.

1. Turn machine on and follow necessary steps in the manual of Appendix A for preparing the UI and machine,

2. place specimen in machine work area,
3. measure the specimen, and
4. perform post-processing of the results.

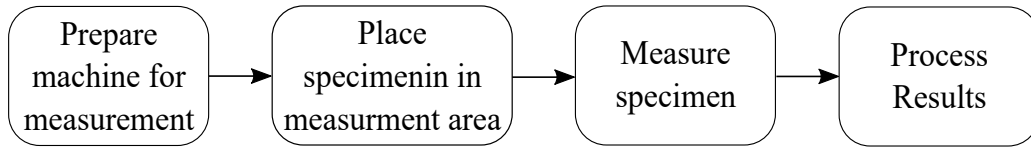


Figure 5.1: Specimen measurement process.

Because measurements are made in three dimensions, a simple measurement of probing either side of the gauge block would not be sufficient to provide the gauge block length, since the axes of the gauge block and μCMM do not align. Figure 5.2 gives a simple illustration of how two points are not sufficient for measuring a misaligned block length in three dimensions. It can be seen that the measured length is larger than the actual length. As a result, a relationship needs to be made between the μCMM 's coordinate system and the axes of the gauge block. This is achieved by probing multiple points (around three) on the side and top of the gauge block. The result gives points that can be used to determine the horizontal and vertical plane and can be chosen as the planes through which the horizontal and vertical axes run. The process involved in determining a plane from three points is as follows:

$$\vec{n} = (\vec{P}_1 - \vec{P}_2) \times (\vec{P}_1 - \vec{P}_3), \quad (5.1)$$

where,

$$\vec{n} = (n_x, n_y, n_z) \in \mathbb{R}^3,$$

$$\vec{P} = (p_1, p_2, p_3) \in \mathbb{R}^3.$$

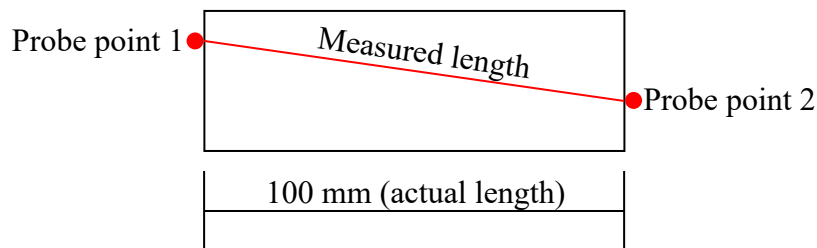


Figure 5.2: Example of probing a gauge block with only two points.

This yields a normal vector for the top and side plane through which the horizontal and vertical axes run. As a result, the angles between planes can be calculated with Equation 5.2, as well as the angle between the line measurement vector and the normal planes by using Equation 5.3. If the length vector is transformed onto each plane, the true length will be revealed.

$$\phi = \arctan \left(\frac{|\vec{n}_T \times \vec{n}_S|}{\vec{n}_T \cdot \vec{n}_S} \right) \quad (5.2)$$

Where \vec{n}_T and \vec{n}_S are the normal vectors that represent the top- and side-plane, respectively. Furthermore, \vec{v} represents the line vector measured for determining the gauge block length.

$$\theta = \arcsin \left(\frac{|\vec{n} \cdot \vec{v}|}{|\vec{n}| \cdot |\vec{v}|} \right) \quad (5.3)$$

Equation 5.3 can then be used to determine the angle of the line vector between the top and the side plane. Once these angles are determined, the line vector can be transformed onto each plane to determine the true measured gauge block length.

As mentioned in Section 2.6, the lasers measure the movement of the tetrahedron, which then saves this data for post-processing. The post-processing is performed in MATLAB[®]. An important aspect of the code for post-processing was the ability to distinguish between upward and downward movement of each motor. This was so that the absolute movement of the system could be measured and stored in memory before another movement is made. The code structure of post-processing is provided in Section D and further simplified in Figure 5.3.

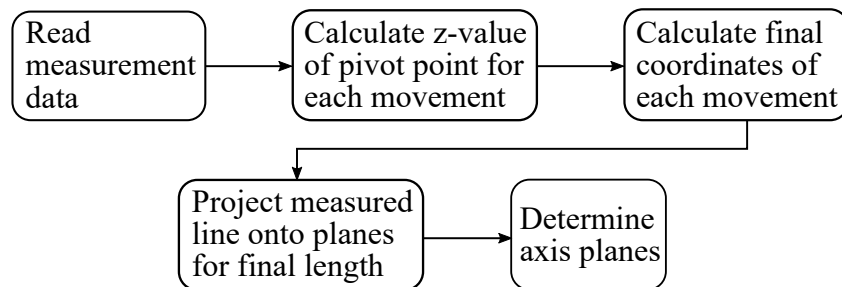


Figure 5.3: Post-processing method followed after a specimen measurement.

Once this is determined, the code proceeds to calculate the vertical delta distances at the start of measurement and during each movement. As a result, the new z-distance is provided based on Equations 2.3 to 2.8, allowing the calculation of the final probe pose using Equations 2.9 to 2.11.

5.3 Measurement results

After performing the experimental procedure of measuring the gauge block, the stored measurement data was processed to give the gauge block length. This revealed the accuracy of the μ CMM. Initial measurement results are given in row 1 of Table 5.1. Additionally, Figure 5.4 shows the alignment of the side and top plane. Clearly, the planes do not have the expected angle of $\phi = 90^\circ$ but have an angle of $\phi = 78.22^\circ$.

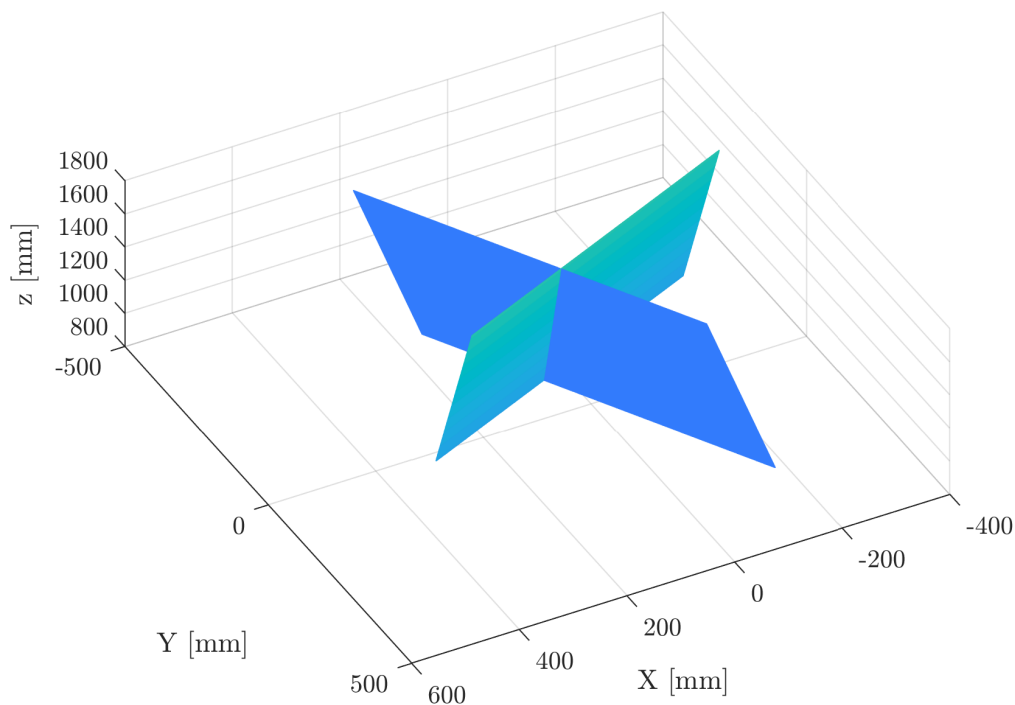


Figure 5.4: Calculated top- and side-plane from accuracy test.

Consequently, this error in the angle between planes shows that either the calculation process is incorrect, or the machine measurement is inaccurate. Correspondingly, the post-processing code was revisited. Initial findings showed that the calibrated b_{in} distance between spherical centers, which was taken from Rugbani (2014), would invoke complex values in Equation 2.5 and 2.6. As a result, a numerical approach—described in detail in

Table 5.1: Post-processing results.

Angle between planes [deg]	Angle between line and top plane [deg]	Angle between line and side plane [deg]	Calculated length [mm]
$\phi = 78.22$	$\theta_1 = 48.31$	$\theta_2 = 27.92$	$L = 35.54$
$\phi = 83.89$	$\theta_1 = 38.73$	$\theta_2 = 26.37$	$L = 60.87$

Chapter 7—was used to get a b_{in} distance which would not give a complex value. Importantly, there would be no concern in the accuracy of calculating the b_{in} distance assuming that the other calibrated machine parameters given by Rugbani (2014) are correct.

In addition, after investigating other aspects of the code, it was found that the μ CMM lasers have a start of measurement range of 60 mm, which was not compensated for in Rugbani’s original calibrated dead distance. With the addition of the calculated b_{in} distance and the start of measurement to the dead distance, the calculated length and angles improved to the values shown in row 2 of Table 5.1. The calculated length nearly doubling and ϕ getting closer to the expected 90° with a value of $\phi = 83.89^\circ$, yet still incorrect.

5.4 Measurement conclusion

With the gauge block measurement that was made to test the current accuracy of the μ CMM, it was found that the system has some inaccuracy’s that returned undesired results. As a result, attempts were made to improve the accuracy by altering the fundamental machine parameters. Interestingly, changing the original calibrated parameters to ones which made more sense, showed a significant improvement in results, however, the desired measurement was still not achieved. For this reason, a deeper investigation was made into the machine parameters, calibration, and errors in the software and physical μ CMM measurement.

Chapter 6

μ CMM investigation

The results of Chapter 5 steered the project in the direction of finding any possible design, equipment, mechanical, electrical, or code errors that lead to the current inaccuracy of the μ CMM. However, as a result of the preliminary tests related to the altered machine parameters, a new hypothesis was proposed: the most significant improvement in measurement accuracy can be achieved by re-calibrating the machine parameters.

6.1 Manual parameter tweaking

The first step was to manually adjust the machine parameters in an attempt to reach the desired gauge block length, as it was the least time intensive approach. The parameters available for adjustment include: (1) laser dead distances d , (2) structure b_{in} distances, and (3) tetrahedron β_{in} values. Each parameter set was individually tuned to observe the effect it had on the final result. Then a final combination of parameters was tuned based on observations, in order to get the desired gauge block length. Table 6.1 shows the summarised results before and after parameter tuning.

Table 6.1: Summarised results from before and after tuning for different measurement data.

	ϕ	d	b	β	b'	Final length
100 standard	83.89	354.02	433.94	68.50	0.00	60.87
100 tuned	55.59	376.97	487.83	73.00	0.50	100.00
100 standard	94.76	354.02	762.44	68.50	0.00	273.27
100 tuned	49.29	376.97	834.95	73.00	0.50	123.04
90 standard	162.73	354.02	762.36	68.50	0.00	28.36
90 tuned	110.46	376.97	834.87	73.00	0.50	450.24

It was noticed that the b_{in} distance couldn't be significantly changed from the calculated minimum b_{in} distance. This was because Equations 2.9 to 2.11 use the x and y variables as the position of each spherical joint. Therefore, when the b_{in} distance is significantly changed, complex values arise. However, a small change in the b_{in} value by subtracting b'_{in} , in the order of 1 mm, makes a large difference in the results.

After tuning the parameters, the calculated gauge block length was 100 mm (Table 6.1), which was the actual length. The next step was to apply these tuned machine parameters to additional measurement data. The project proceeded to measure the gauge block again as well as one smaller one. Findings showed that neither the standard or tuned parameters used for the first gauge block measurement showed successful results for the other gauge block measurements.

These results suggest that system inaccuracies are not from coding. However, the calibration of the system is a direct concern as these parameters are the underpinning variables of the final results. Therefore, calibration of the μ CMM needs to be thoroughly addressed while considering all factors that may influence calibration, measurement, and the final results.

6.2 Numerical calibration

Tuning each machine parameter manually presents many challenges. This relies on the user's judgment for what is considered to be sufficient. This can also prove to be a tedious and difficult task since it requires a lot of time to find and test the many different parameter values and combinations possible. For this reason, an alternative method should be used, which addresses the drawbacks of manual parameter tuning and ideally tweaks the parameters correctly to give the final desired results. One such method very well suited for this kind of problem is a numerical optimization method.

Numerical optimization applied to the calibration of the μ CMM is aimed at finding the optimum combination of parameters that will give the best final result for measurement. For instance, if multiple measurements of a gauge block of 100 mm are performed, then the goal of the numerical calibration is to tune each machine parameter such that the post-processing of the physical measurement data will yield the desired 100 mm each time. Obviously, this is assuming that a numerical solution can be reached.

6.2.1 Single arm numerical calibration procedure

In a greater attempt to reduce the possibility and complexity of any errors being from the post-processing code, it was decided that starting with the fundamentals of the machine and systematically working towards the entire measurement, would help the process by eliminating possible factors that might be affecting it. As a result, based on the machine equations, it was decided that Equations 2.3 to 2.8 would first be considered. For this reason, the focus can be placed on first ensuring that the z-value of each spherical joint—or pivot point—is correct, since the rest of the machine equations rely on this being correct. Initially, further simplification is made by optimizing the parameters for only one arm, namely: Dead l_i and l_{in} , b_{in} , and β_{in} . Accordingly, the numerical optimization process becomes simpler and less intensive for processing. The procedure for numerical calibration is outlined by Figure 6.1.

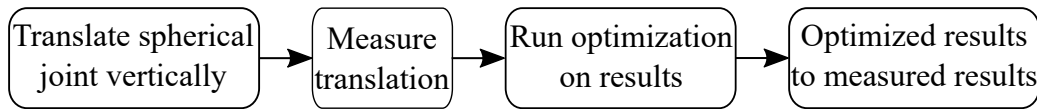


Figure 6.1: Process outline for numerical calibration procedure.

The vertical position of the spherical joint is measured by using a digital height gauge with accuracy and resolution of 0.01 mm, as can be seen in Figure 6.2. The height gauge is placed on the lead screw bracket of the μ CMM. As the bracket is displaced vertically, the height gauge is moved accordingly and the height difference measurement is recorded. Correspondingly, for every movement made, arm laser distances are recorded. The optimization algorithms used are a gradient descent algorithm "fmincon" and a genetic algorithm "ga" in MATLAB[®] and is shown by the code in Appendix B.



Figure 6.2: Height gauge measuring the change in pivot point height on the lead screw bracket.

The objective function uses Equations 2.3 to 2.8 and outputs the square of the difference between the measured and calculated height differences. The standard deviation of these errors, σ_{err} , are then calculated and, finally, the parameters that yield the smallest standard deviation of the error are the ones chosen. Multiple measurements were taken, with differing numbers of points. The results shown in Table 6.2 are calibration results for the GA since it showed better results than the "fmincon" algorithm. Additionally, the calibration results are shown for only one arm, since the same behaviour was observed for the remaining two arms.

Table 6.2: Numerical optimization results for one arm.

	d_i [mm]	d_{in} [mm]	β_{in} [mm]	b [°]	σ_{err} [mm]	Measured points
Test 1	320.94	372.83	79.93	493.01	0.91	6
Test 2	363.95	374.49	62.75	421.38	0.76	11
Test 3	383.07	343.57	69.16	453.62	0.11	3
Expected	349.40	354.02	68.36	434.60	-	-

Observing Table 6.2 shows values that are significantly different from the original calibrated parameters. Moreover, except for d_{in} of test 1 and test 2, these parameters are not similar. This could be that some of the measurement points have a larger degree of error, for example the pivot point could be displaced by 10 mm and the μ CMM measured it as 15 mm, but another pivot point displacement of 30 mm could have the μ CMM measure it as 31 mm. When the values with a larger degree of error are present in a set of measurement data, it influence the numerical optimisation negatively as it is trying to find the best match for all the measurement data.

Observing Figure 6.3, it can be seen that the error is larger for certain measurement points, particularly between 5 mm and 15 mm. As a result, the third and final measurement consisted of three points, which were some of the lowest errors in the previous measurements. What is important to note is that test 3's parameters are significantly closer to the expected parameters, except for d_i . Nonetheless, the resulting final standard deviation of test 3 error was 0.11, which was 12.33 % the error of test 1 and 14.80 % the error of test 2. This is a significant improvement over the original results. However, the fact that the other measurements had undesirable results cannot be ignored.

6.2.2 Multi-arm numerical calibration

After calibrating one arm of the μ CMM, it was decided to approach calibrating all three arms with the aim that these variables will influence the optimization in a manner that steers it to the global minimum rather than a local minimum.

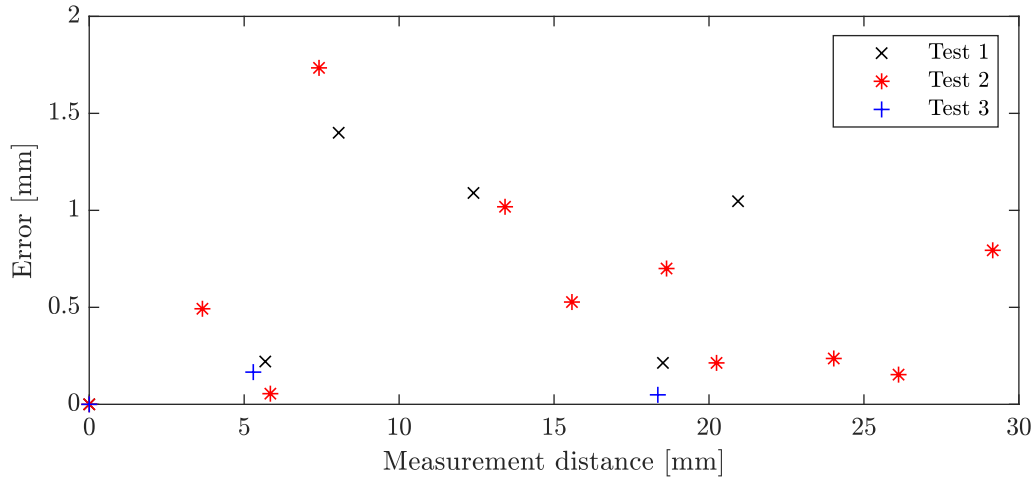


Figure 6.3: Error data for three measurements performed with the GA algorithm.

The measurement process is similar to the single arm and the one outlined in Figure 6.1, except with the same process applied to all three tetrahedron arms. After an arm was measured the μ CMM was reset to the measurement start position, and therefore, three measurements were performed in total. Similar to Section 6.2.1, both the "fmincon" and GA algorithm were used for optimization, with the GA algorithm showing the best results. The results for the GA algorithm are provided in Table 6.3 along with the calibrated parameters in Table 6.4.

Columns 1 through 3 of Table 6.3 are the measured z-value's of each pivot point. Columns 4 through 6 are the differences between the measured and calculated z-value of each pivot point. All nine machine parameters that need to be calibrated were optimized simultaneously. More specifically, the optimization algorithm alters the value for each parameter to find an error between the measured and calculated z-value that is the smallest. Therefore, the final error is subject to the combination of parameter values.

Importantly, if an arm is to be calibrated, it means that it needs the dead distance of the previous arm, its own dead distance, the angle β_{in} between arm_a and arm_c , and finally, the b_{in} distance between pivot point i and pivot point in as described by Equations 2.3 to 2.8. Because these equations need the laser length of more than one arm and, hence, the dead distance of more than one arm, it means that each dead distance needs to be applied and optimized twice, as illustrated by Figure 6.4.

Consequently, the dead distance needs to satisfy the z-height error from two equations. As a result, the optimization algorithm tries to find calibration parameters that will minimize the overall error without specifically calibrating

Table 6.3: Three arm calibration results for 15 measurement points per arm.

Error: measured–calculated height					
z_a	z_b	z_c	err_b	err_a	err_c
0.00	0.00	0.00	0.01	12.45	43.85
5.44	3.49	2.66	2.45	8.96	41.19
8.17	6.13	6.19	-0.05	6.32	37.66
11.26	10.66	8.96	0.07	1.79	34.89
15.81	12.45	11.75	-0.02	0.00	32.10
19.54	15.85	16.04	1.74	-3.40	27.15
22.51	20.53	19.54	1.98	-3.92	23.65
26.54	27.58	23.19	2.82	-2.90	20.00
29.58	31.37	26.23	2.68	-2.71	16.96
32.50	34.10	30.77	3.20	-1.95	12.42
37.77	39.38	35.64	4.01	-1.98	7.86
39.66	41.08	39.23	4.19	-1.70	4.27
46.75	43.37	43.53	4.83	-1.69	-0.03
49.18	46.26	47.87	5.19	-1.40	-4.37
51.58	49.18	52.04	5.20	-1.04	-8.54

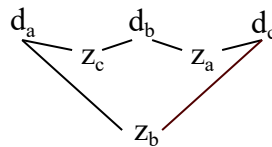


Figure 6.4: Dead distances needed to calculate each pivot point z-height.

each arm. This could result in negative effects since more local minima could be introduced as a result of the simultaneous optimization. Moreover, the algorithm could favour the calibration of variables attributed to one arm if the error of two arms are inversely proportional to each other. This could be a possibility of the large error seen in column 3 of Table 6.3 where the algorithm favoured arm_a as opposed to arm_c since the error values for arm_a are significantly lower than that of arm_c . However, this does suggest the possibility of one arm being able to measure more accurately than the other.

One interesting pattern to note in Table 6.3 is the decreasing error value. Figure 6.5 plots the combined errors of each arm as a function of the average displacement from the starting point. The plot shows a general descent in error as there is an increase in displacement from the start of measurement point. This leads to the possibility that the design of the μ CMM could be better suited to larger displacements. However, the height difference error is largely influenced by that of arm_c . In fact, the error of arm_a has a smaller error for the smaller displacements as opposed to the larger displacements,

Table 6.4: Calibrated parameters.

Arm	d_i	β_{in}	b_{in}
a	354.99	76.04	480.98
b_{in}	382.53	73.15	479.72
c	354.94	73.50	483.91

whereas arm_b shows similar behaviour to arm_c . These results were consistent throughout each optimization that was run.

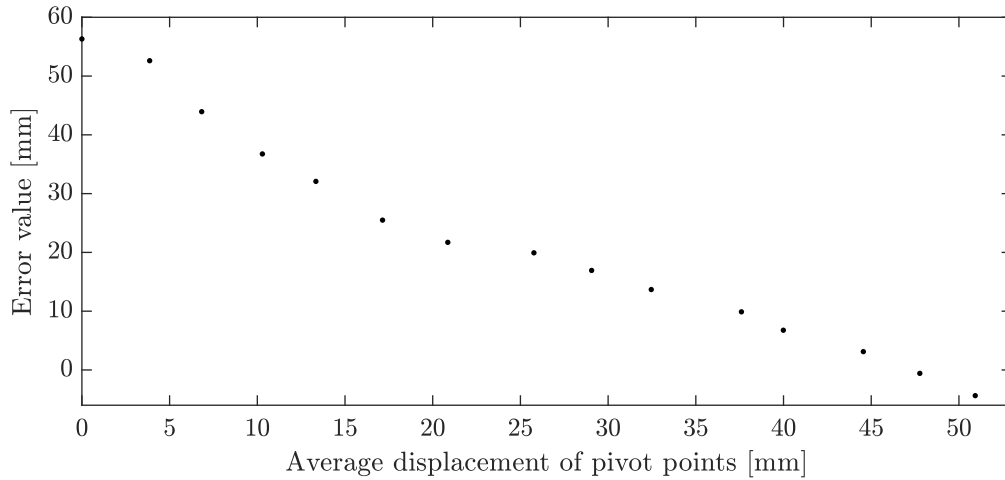


Figure 6.5: A plot of the combined error of each arm.

In an attempt to decrease the errors from the simultaneous calibration, a different approach was designed to calibrate each arm individually to rule out possible negative influences of simultaneous calibration. As a result, the calibration code was modified with the sequential parameter calibration process for each arm. The sequence of parameters that need to be calibrated are as follows:

1. arm_a : d_c , d_a , β_{ca} and b_{ca}
2. arm_b : d_b , β_{ab} and b_{ab}
3. arm_c : β_{bc} and b_{bc}

Therefore, the objective functions are less complex and are less likely to be subject to local minimum or other error increasing influences. Furthermore, calibrating each of these individually, present possibilities of speeding up the optimization process whilst reducing the complexity. Moreover, as the optimization progresses and each parameter is calibrated based on previous parameters, greater harmony between parameters and errors could be achieved. However, because the parameters of the next arm are based on the previous

arm, it means that if the previous arm is incorrectly calibrated, then the next parameters could have an accumulated error. With the necessary modifications to the optimization algorithms made, the calibration process was run.

A large improvement over results from the simultaneous calibration can be seen in Table 6.5, which supports the theory of individual calibration. Not only does this method show an improvement in the overall error and accuracy of each arm, but it also drastically decreases the error differences between each arm. For arm_b and arm_c less deviation is shown, but the same pattern of increase in accuracy as displacement is increased is exhibited for the individual arm calibration. This recurring pattern suggests that the μ CMM is less accurate at smaller vertical displacements of the pivot point.

Table 6.5: Individual arm calibration error results.

err_a [mm]	err_b [mm]	err_c [mm]
0.02	12.45	6.51
1.72	8.96	4.26
-0.77	6.32	1.58
-0.96	1.79	0.00
-1.49	0.00	-1.74
-0.29	-3.40	-2.14
-0.37	-3.75	-2.76
-0.01	-2.47	-2.29
-0.43	-2.16	-1.97
-0.25	-1.30	-1.60
-0.04	-1.20	-1.01
-0.06	-0.87	-0.68
-0.17	-0.79	-0.33
-0.08	-0.43	0.06
-0.29	0.02	0.37

Although the results of the individual arm calibration show a significant improvement, some errors are still up to 13 mm, with a large majority of errors being in the 1 to 3 mm range. This is unsatisfactory since the μ CMM is designed and intended to measure with an accuracy of at least 10 μ m. Therefore, a "brute force" optimization method was proposed to test and calibrate the machine to see if the results can be improved. The combined and individual arm calibration method will also be used for the brute force method to compare results.

A set of random numbers are generated for each of the parameters. The numbers generated are within the estimated range of parameter values ex-

pected based on previous calibration results and geometrical data of the μ CMM. The brute force algorithm generates a set of random numbers within an estimated range for each parameter, and then these combinations are tested using Equations 2.3 to 2.8. Finally, the combination with the lowest absolute error value is the one chosen, keeping in mind that 80×10^6 iterations are made—the maximum memory limit that MATLAB[®] allows. Interestingly, Figure 6.6 shows the error values for the combined brute force calibration have a considerable improvement in the errors of arm_c , however, an increase in error was experienced with the remaining arms. Importantly, the overall error value is lower for the combined brute force as opposed to the combined GA optimization. This is believed to be because the brute force algorithm cannot be subject to a local minimum of an objective function.

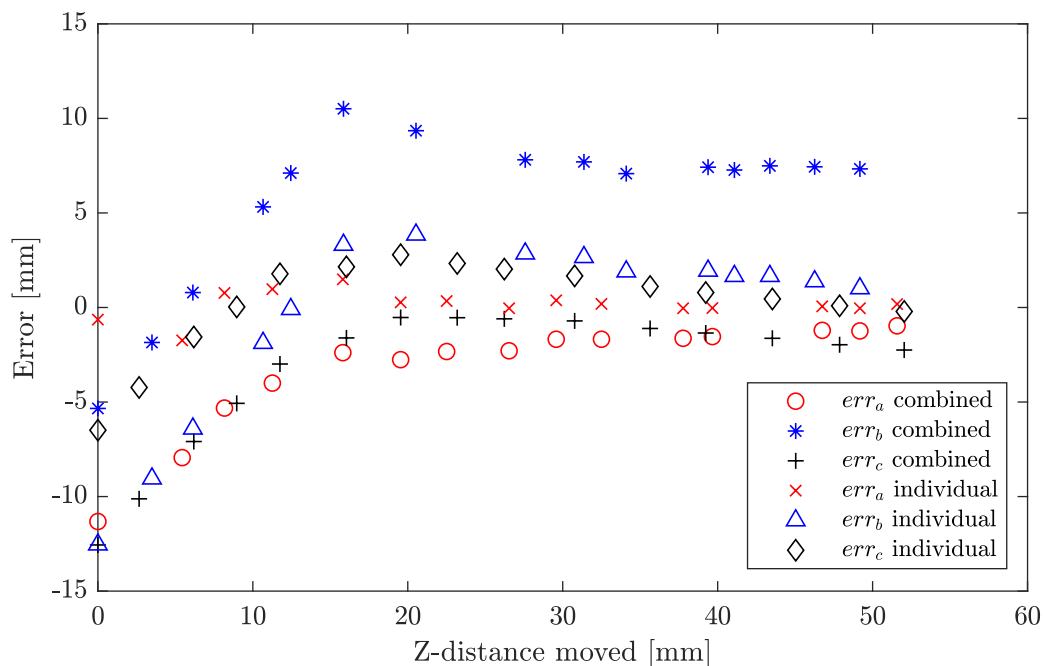


Figure 6.6: Individual and combined brute force calibration error results.

A second important point to note on the brute force algorithm is the results of the individual calibration. In comparison to the individual calibration of the GA, the yielded error values are largely similar to each other, however not exactly similar. This led to a suspicion that there may be no unique solution for this calibration and that the GA algorithm is likely to converge on local minima. Consequently, with the different algorithms and combinations used still not showing satisfactory results, a decision was made to investigate the reasons behind these results.

6.3 Error investigation

As previously mentioned, the uncertainty for the current calibration errors needs to be investigated. Considering that the fundamental code was used in Section 6.2.1 and 6.2.2, it is likely that the errors seen from the numerical calibration are not algorithm or coding errors. As a result, the movement of the machine needs to be investigated as well as how the laser is influenced by the movement.

Initial observations of the laser measurement data for each of the three tests showed that, although each pivot point was starting at the same vertical point, each laser distance value for the start of measurement was slightly different, as can be seen in Table 6.6. It is important to note that these variables are all within the expected accuracy of the laser. Moreover, the start point is set by an exact encoder count value on each of the Zaber[®] T-Series[™] motors. However, the value for l_a in row 2, column 3, shows a deviation on the other laser values which is not within the allowable measurement accuracy errors.

Table 6.6: Start of measurement laser data for test 1, 2 and 3.

l_a [mm]	l_b [mm]	l_c [mm]
33.70	33.51	37.14
33.41	33.56	37.08
33.70	33.51	37.14

Each pivot point reaches the starting point as set by the exact encoder count. This point is reached and given ample time for settling before any measurement is performed. There is a possibility that, when performing a measurement experiment on the μ CMM, that the movement during the measurement could settle the tetrahedron in a different starting position, albeit at the same pivot point heights. To test this, one arm was moved vertically and set to return to its starting point, and then the laser value at the start and end of measurement were compared. The suspicion of different settling positions was confirmed during this test.

This was believed to be because of friction between the arms and the linear sliders of the system, and while the lasers were measuring and returning data, a light force was applied on the end of one tetrahedron arms. Indeed, a change in the laser distance value was noticed and this value did not return to the same value before applying the force. The laser value was permanently shifted by up to 2 mm. This presents large challenges, since how can a machine be calibrated when there are uncertainties of 2 mm, but an accuracy of 10 μ m is expected. An attempt was made at improving the settling behaviour of the tetrahedron

by making sure that all joints, bearings and sliders were sufficiently lubricated, but no difference in results was seen.

Therefore, it was believed that it was necessary for the tetrahedron to be properly settled before each measurement. This was achieved by developing a set of movements that the μ CMM needs to perform before any measurement can be performed. The process involves moving each motor to its maximum and minimum height and then setting it back to the original starting point. The intent was not to reduce the bad settling behaviour but to make the starting point more consistent. Table 6.7 shows the results for the settling procedure performed a total of 6 iterations. The maximum deviation in starting distance is 0.45 mm, which shows no improvement over starting points without the settling sequence. Consequently, this result adds to the measurement uncertainty of the μ CMM.

Table 6.7: Laser distances for starting point.

	l_a [mm]	l_b [mm]	l_c [mm]
Test 1	34.11	32.41	37.65
Test 2	34.21	32.17	37.82
Test 3	34.11	32.46	37.67
Test 4	34.26	32.22	37.78
Test 5	33.86	32.17	37.53
Test 6	33.86	32.17	37.50
Range	0.45	0.29	0.32

In addition, during the settling error investigation, one specific error was noticed: A pivot point was displaced upward from the starting point, but the laser distance value remained the same until the pivot point displaced past an average of 4 mm. The value seems quite inconsistent and is difficult to pin down as a specific value where the laser starts measuring. However, this value is affected based on the point where the tetrahedron has settled, which shows that there is some play in the system before these displacements are read by the lasers. This further confirms the error values exhibited in the numerical calibration where the larger displacements have a smaller error value.

During this test, it was observed that the ratio between the distance displaced by the pivot point and the laser displacement measured are not proportional to each other. In fact, a much larger vertical pivot point displacement is needed for a small displacement to be shown in the laser values. An experiment was conducted to test the relationship between laser displacement and probe movement. The probe was displaced from one side of a 100 mm gauge block to the other side since this is within the work area of the μ CMM. While the probe was moved the laser displacement was recorded. With a 100 mm

displacement of the probe tip, only a displacement of 5 mm was shown on the laser—a ratio of 20:1 for probe versus laser movement. This, in essence, makes the laser 20 times less accurate and takes the laser reading from a 0.05 mm accuracy to 1 mm accuracy.

With considering all the errors, a final measurement experiment was performed to test the numerical calibration method, however, the difference here is that a large data measurement set, 100 measurement points, with large vertical point displacements was the goal. The large data set is to increase the chance of accurate measurements being made. Figure 6.7 shows that larger displacements are more accurate, however, the expected accuracy was still not reached.

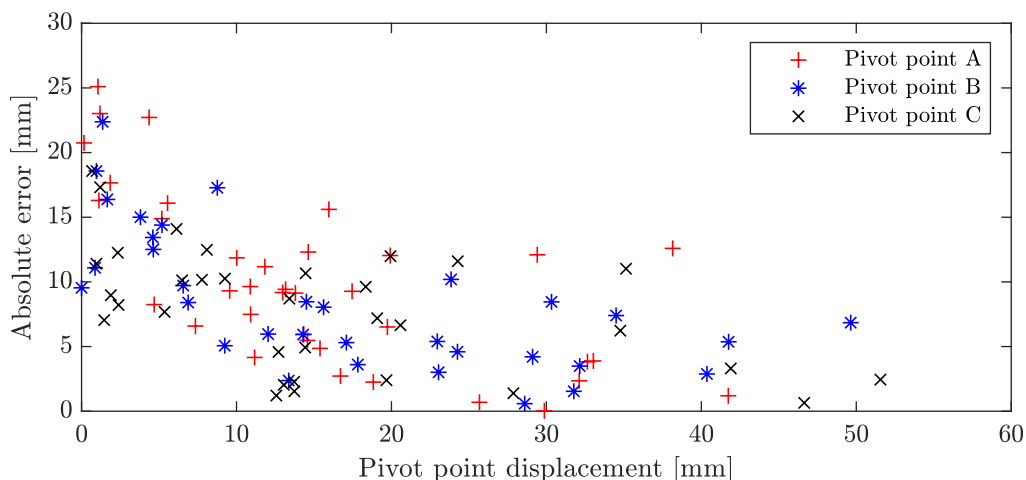


Figure 6.7: Measurement errors for calibration using a large dataset and large displacements.

In addition to the aforementioned accuracy tests, a final test was performed to observe the characteristics of vertical motor movement and laser displacement. Therefore, an experiment was conducted where each pivot point was displaced vertically from approximately 0 mm to 100 mm for a total of 5 iterations per arm. The relationship between motor displacement and laser displacement are then expressed as a ratio between each other by the following equations:

$$\delta l = l_s - l_f, \quad (6.1)$$

$$\delta h = h_s - h_f, \quad (6.2)$$

$$ratio = \frac{\delta h}{\delta l}. \quad (6.3)$$

Where l and h represent the displacement of the laser and height gauge, and subscripts s and f denote the start and finish points, respectively.

Theoretically, these ratios should be the same if not very similar. However, the ratios shown in Table 6.8 prove otherwise, since the ratios for different tests on the same arm are not similar. Furthermore, when comparing the ratios between l_a , l_b , and l_c , different ratios are found.

Table 6.8: Repeated ratios of laser versus height gauge displacement.

	$\frac{l_a}{h_a}$	$\frac{l_b}{h_b}$	$\frac{l_c}{h_c}$
Test 1	5.03	6.20	5.53
Test 2	5.12	6.31	5.58
Test 3	5.39	6.34	5.61
Test 4	5.39	6.30	5.58
Test 5	5.56	6.41	5.65
σ	0.20	0.07	0.04

6.4 Investigation conclusion

A thorough test was made of different calibration methods for calibrating the μ CMM. Methods included manual parameter tweaking and numerical calibration—linear descent, genetic algorithm, and brute force. These parameters were forced to be within a set range that was expected based on calibration from Rugbani (2014), as well as physical gauging to get measurement estimates of parameters. The manual tuning of parameters showed unfavourable results on the gauge block measurement. As a result, a decision was made to simplify the parameter calibration process in order to increase the speed of optimisation and reduce the inherent complexity errors. This was achieved by the parameters based only on the z-value measurement and equations.

This attempt also proved to be unsuccessful, although many attempts were made with numerous calibration methods such as a GA and random number brute force algorithm. However, many crucial discoveries were made in the interim, which gave a much deeper understanding of the errors experienced between the expected and calculated measurements. It was found that the tetrahedron of the μ CMM does not settle in the same position as a result of possible friction or design errors. Furthermore, the relationship between vertical pivot point displacement and laser displacement is not proportional, with a

much larger pivot point movement needed to read a small laser displacement. As a result of these flaws, the laser displacement does not change until a certain amount of displacement of the pivot point is made, which leaves a "dead spot" in the measurement and requires larger displacements to be able to read displacement. This was confirmed by the optimization where the error values were much lower for larger displacements.

Further discoveries were made that showed, for a set arm movement, the ratio between movement and laser measurement were inconsistent. Furthermore, when comparing the ratios between each arm, for the same distance, the ratios were significantly different as well. This exposes more flaws in the system and proves that these errors will, undoubtedly, make calibration a difficult process, as well as not being able to achieve a measurement accuracy lower than the sum of the errors found.

With the given results, it was decided that a physical calibration with another CMM of the μ CMM would be the best approach moving forward since the numerical approach yielded unsuccessful results.

Chapter 7

μ CMM calibration

A physical calibration of the μ CMM provides a robust and accurate method for determining the machine parameters. This was used as a last resort since, if the numerical calibration showed successful results, it could have been a simpler and quicker method for calibration, particularly if probes are exchanged. Therefore, with the numerical calibration being unsuccessful, this chapter provides the calibration method used to determine the parameters of the tetrahedron as well as the post-processing of the calibration results. For reference, CMM₂, the μ CMM of this project, is calibrated by CMM₁, the external CMM (Figure 7.1). CMM₁ undergoes its own calibration process before any calibration is performed on CMM₂.

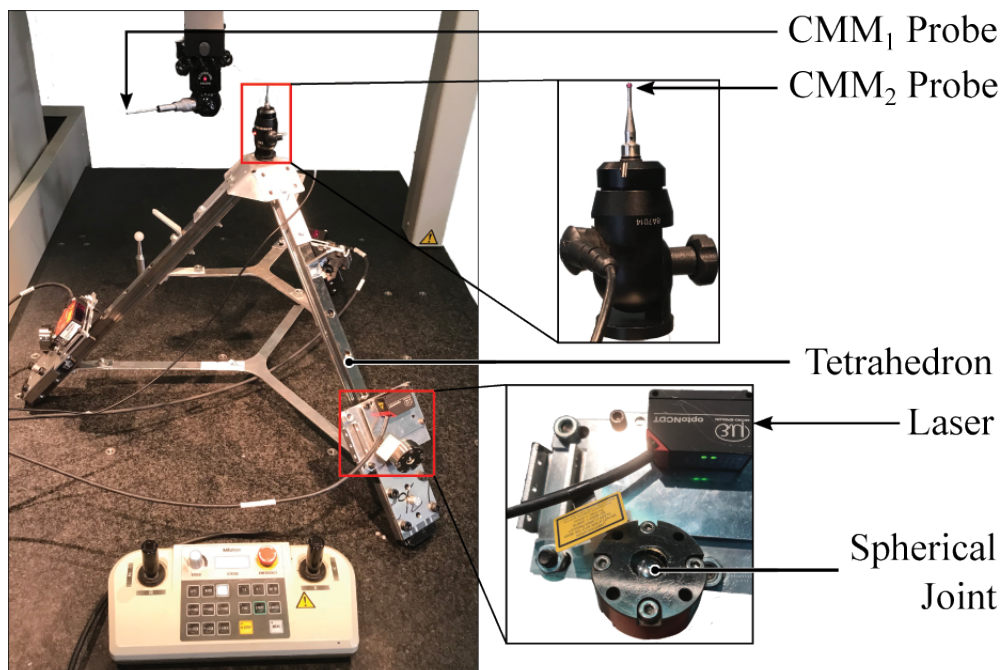


Figure 7.1: The tetrahedron on CMM₁'s platform for calibration.

7.1 Calibration method

The calibration of parameters of the tetrahedron begins with placing it in the work area of CMM₁. Initially, all sliders that support the spherical joints and lasers are at their maximum distance from the reflective plate. The tetrahedron is added to the reference coordinate system of CMM₁ by measuring a cylindrical hole and a flat surface of the tetrahedron. Measuring these points and shapes give sufficient variables for determining the tetrahedron x-, y-, and z-coordinate system. Each spherical joint is then measured by probing a minimum of 5 points as far as possible from each other so that the sphere can be fully characterized for the most accurate results. Next, the sliders are moved up as close to the reflector plate with the lasers still being within measuring range. The same spherical joint measurement process is followed. Finally, the probe tip is measured with the same procedure as a sphere, since the tip is spherical.

During the calibration, all laser measurements were recorded for later reference when calculating the dead distance. CMM₁ returns the coordinates of the centre point, the radius, and roundness of every spherical joint measured.

7.2 Calibration data processing

The coordinates of the spherical joints coupled with the laser measurements, are used to calculate the tetrahedron parameters. Finally, these parameters are used to calculate the remaining b_{in} values, since this could not be measured on the tetrahedron. The first order of processing is to determine the line on which each linear slider travels. This is achieved by creating a parametric equation that describes the vector of the linear slider path:

$$\vec{r}_i = \vec{P}_{i1} + t \frac{\vec{P}_{i1} - \vec{P}_{i2}}{|\vec{P}_{i1} - \vec{P}_{i2}|}, \quad i = a, b, c. \quad (7.1)$$

As a result, the line on which the slider travels and, hence, the path of each laser, is determined. Theoretically, these paths should intersect as part of the design of the μ CMM. In contrast to the theoretical expectations, Figure 7.2 shows the paths travelled by the spherical joints do not have the same intersection point and, furthermore, the probe point does not intersect with any of the vector lines.

This discovery shows major inaccuracies in the manufacturing of the μ CMM tetrahedron, with many of the expected design parameters not being reached. Consequently, a new vector path needs to be determined for the

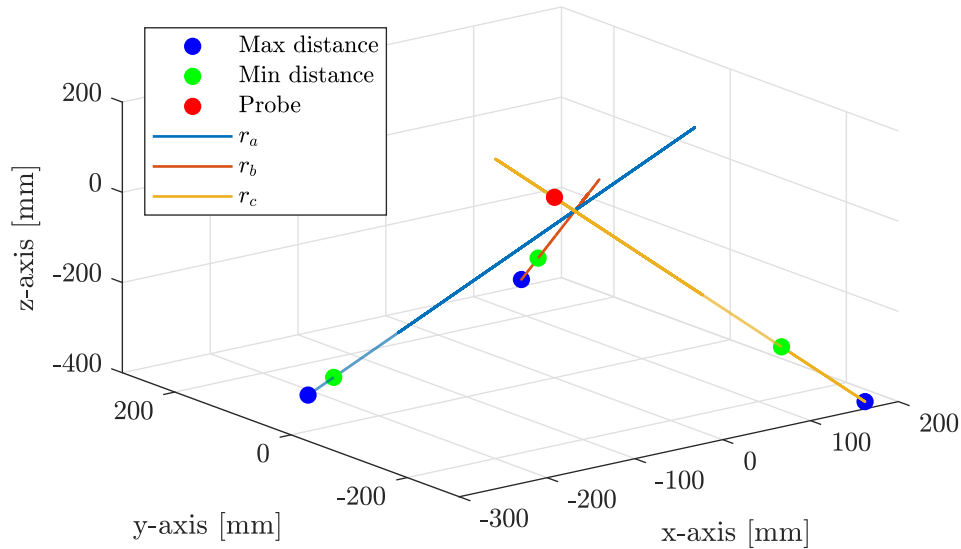


Figure 7.2: A visualisation of slider paths, maximum slider movement, and probe point.

laser to probe point. Two methods were proposed: (1) join the starting vectors and the probe point using Equation 7.1, or (2) create a theoretical probe point that is at the minimum distance between each vector.

The process of creating a theoretical probe point with the minimum overall distance involves determining the distance between any two points on two vectors:

$$L = (\vec{S}_i - \vec{S}_{in}) \cdot (\vec{S}_i - \vec{S}_{in}). \quad (7.2)$$

Where S_i and S_{in} denote the position vector for the two linear paths travelled. Equation 7.2 is then put into an optimization algorithm that finds the smallest value for L . This process is repeated for all combinations of the three tetrahedron arms. Finally, the average between the three coordinates is taken to find the theoretical probe point.

A major drawback, however, of the vector paths and probe point not intersecting is that the dead distance value now changes as a result of the position at which the pivot point resides. This is demonstrated by Figure 7.3, where l represents the laser length reading, d the dead distance, and L' the overall length for the new vector path. The change is due to the fact that, if the pivot point vector intersected the probe point, the relationship could be described linearly—as the laser distance increases linearly, so does the overall distance from the intersection point. However, with a new intersection point and pivot

point distance physically constrained to changing linearly on the original vector path, the distance between the pivot point and new probe point does not increase linearly. To summarise, this is because the path that the joint travels on and the path from the joint to the new probe point are not collinear.

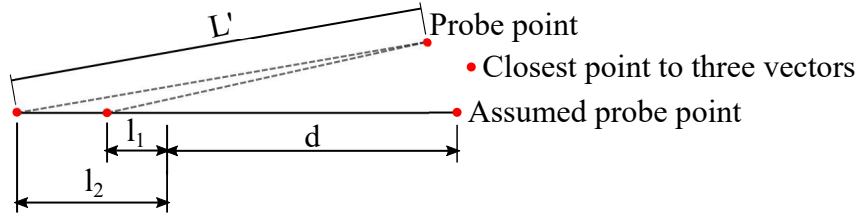


Figure 7.3: Illustration for a probe point not on the linear path of the slider.

Equation 7.3 describes the relationship between the pivot point r , intersection point p , and dead distance d .

$$L = l + d \quad (7.3)$$

Since the laser distance l changes linearly, and the overall distance between the new intersection point L' does not increase linearly, the only variable left to satisfy the nonlinear behaviour of Equation 7.3 is that the dead distance d be nonlinear. For simplicity, the average between the dead distances between the two points is taken as the dead distance. Once the dead distance is calculated the angles between tetrahedron arms can be calculated.

The β_{in} values of the tetrahedron are calculated by determining the angles between the new vectors that intersect the probe point with the following equation:

$$\beta_i = \cos^{-1} \frac{P_i \cdot P_{in}}{|P_i - P_{in}|}. \quad (7.4)$$

The final value left to be determined for calibration is the b_{in} distance between pivot points. This distance can be calculated by using the cosine rule of Equation 7.5 to determine the distance between pivot points (Figure 2.16). If all joints are set to the same vertical height, then the distance between pivot points will be the b_{in} distance. However, due to manufacturing inaccuracies, the same height setting for the motors does not necessarily mean that the pivot points will be at the same height. Therefore, a method was devised to calculate the b_{in} value based on the previously calibrated parameters.

$$b_{in} = \sqrt{l_i^2 + l_{in}^2 - 2l_i l_{in} \cos(\beta_{in})} \quad (7.5)$$

The idea is the minimum distance between pivot point i and in can be found, and if it is found, this will be the b_{in} value. Therefore, the method first involves displacing the pivot points vertically for their entire range while recording the laser measurement data. Finally, the laser measurement data and the angle β_{in} between tetrahedron arms are used to solve Equation 7.5. The key here is: all the laser measurement data is applied, but the minimum value for b_{in} calculated from the data set is the final b_{in} value.

7.3 Calibration results

With the use of calibration data provided and the processing methods, described in Section 7.2, the final calibrated parameters were determined and are provided in Table 7.1. These parameters are very similar to the original calibrated parameters of Rugbani (2014). However, there are differences in the calibrated values, particularly for the dead distance. Important to note is that the dead distance excludes the start of measurement value of the laser, which is 60 mm.

Table 7.1: Calibrated parameter values.

Arm	d_i [mm]	β_{in} [°]	b_{in} [mm]
a	314.48	68.25	454.17
b	312.74	69.03	462.47
c	309.42	68.70	461.67

As was mentioned in Section 7.1, the b_{in} value was determined by moving the pivot point vertically at a constant velocity. During this movement, the laser displacement measurement was recorded. In the process of determining the b_{in} value, an interesting discovery was made on the μ CMM while plotting the laser distance values. As the pivot point moves down from its topmost point down to the bottom point, the laser value decreases until a point and then increases in value again. Figure 7.4 shows that the laser distance value appears to follow the form of a parabolic curve. Admittedly, this behaviour was not expected. The behaviour observed applies to all three arms.

What can be observed from Figure 7.4 is that a displacement between approximately -25 mm and 25 mm, the rate at which the laser value changes is slower. This could further explain the reason behind the μ CMM being more accurate at higher displacements from the starting point. An observation

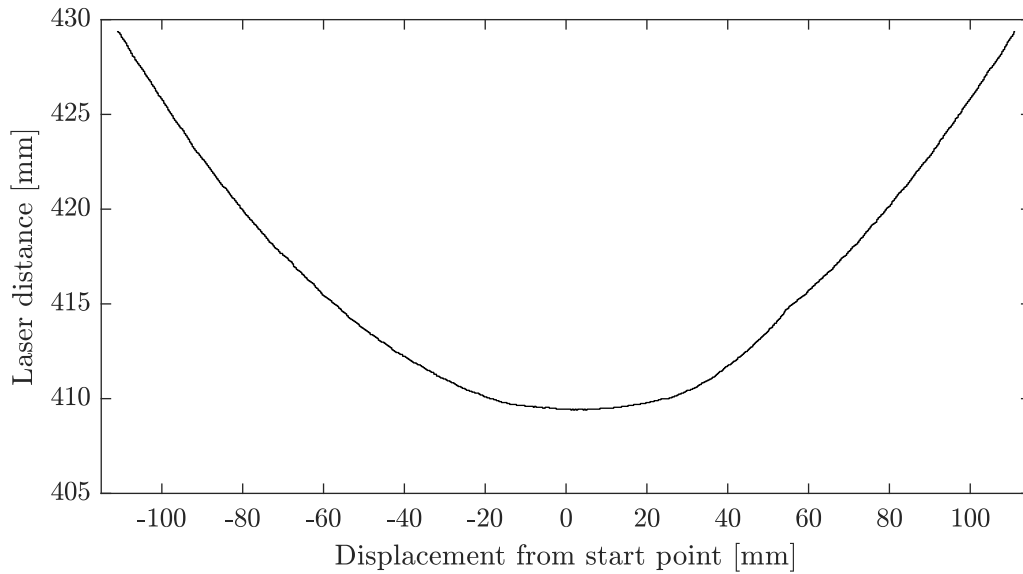


Figure 7.4: Laser distance value plot as a function of data points recorded.

was made that larger displacements alone are not sufficient for improving the chances of an accurate measurement. Rather, displacements that are far from the initial pivot point would increase the chance of an accurate measurement.

This further supports the inability to perform an accurate numerical calibration on the structure, since the behaviour of the laser displacement is not linear. More importantly, two solutions exist, since one laser value could represent an equivalent displacement on either side of the start of measurement point. This could explain some of the problems encountered with the numerical calibration approaches.

Chapter 8

Post-calibration

After calibration had been performed, it was necessary to validate the calibration parameters. Given the drawbacks of the system where the motors need a vertical displacement of at least 15 mm, measuring a gauge block of 100 mm proves to be a difficult task since the gauge block is already at the limit of the 100 mm work envelope of the μ CMM. Hence, small movements would be required to measure points on both ends of the gauge block. Nonetheless, a measurement was performed on a 100 mm gauge block while attempting to make larger displacements, to observe if any differences were made from the previous gauge block measurement results. However, the measurement and processing of the data with the new calibrated parameters showed unfavourable results. This could be due to many factors, such as the settling of the tetrahedron, which creates a dead spot, or the measurement ratio. The results showing that the laser displacement of the tetrahedron moves in a parabolic manner instead of a linear manner, meant that the possibility of using the incorrect direction of movement for calculation, as well as incorrect lengths, were very high. Therefore, a simpler test was devised where the variables of larger displacements and direction of movement could be controlled.

8.1 Micrometer stage calibration test

A test was devised where the movement of the probe could be measured and compared to the numerical processing of the movement of the probe tip. Figure 8.1 shows a 3-axis micrometre stage (3MS) and a v-block that was used to measure the movement of the probe tip, which would give a reference value for comparison. The 3MS is an Edmund Optics micrometre stage with a 50 mm x 50 mm x 25 mm measurement range in x, y and z, respectively, and a 0.01 mm accuracy.

Firstly, the 3MS was placed in the centre of the μ CMM structure with

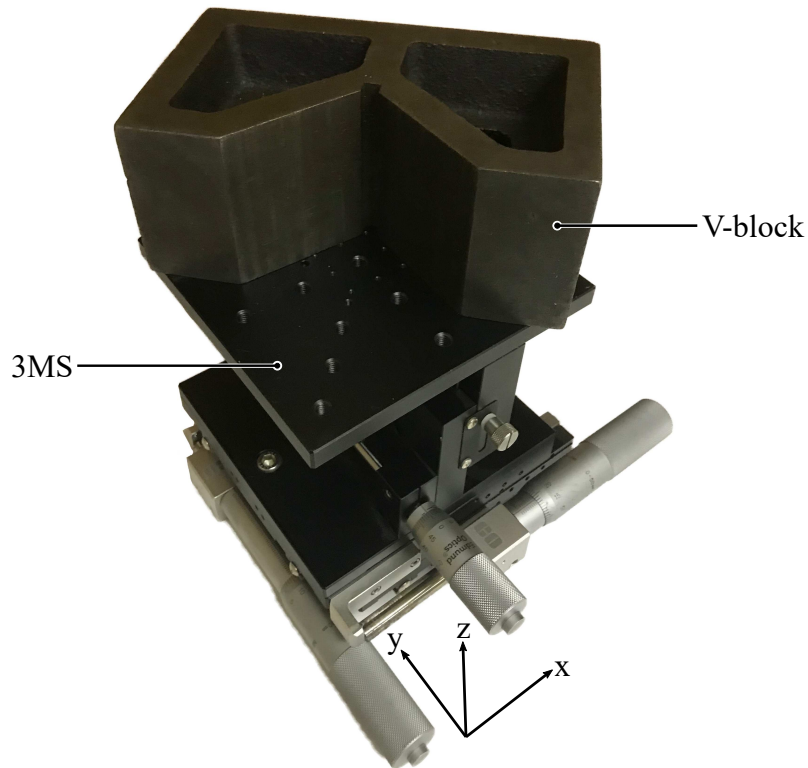


Figure 8.1: 3-Axis micrometer stage.

a v-block attached to it. The alignment of the flat planes of v-block to the axes of the 3MS was an important part of the experiment. Hence, the v-block planes were aligned with the markings on the 3MS to ensure that the v-block planes and 3MS axes align.

Secondly, the axes of the 3MS were then aligned with the axis of the μ CMM arm that was moved during the calibration test. However, this alignment was not critical since, if the movement of the probe veered off to one direction, this displacement could still be measured by the 3MS.

The process started by positioning the probe in a pose where the motor bracket heights are in as much of an equal z-position as possible. This was so that the starting point of $[0, 0, 0]$ could be reached, and then the corresponding movements or measurements were from this reference point. The probe was then moved a total of 3 times from the initial point, giving a total of 4 measured points, with each movement recorded and measured in all three axes. The 3MS was moved in a manner such that the probe tip touches the v-block z-plane to determine the z-coordinate, and the same was applied to the x- and y-plane to determine the x- and y-coordinates, respectively. Each time the probe tip was touched, the probe made a signalling "beep" sound, which also saved all three laser measurements at the point of probe and plane impact in

a corresponding .csv file.

8.2 Micrometer stage test results

For the processing of the 3MS measurement experiment, the absolute distance of the probe moved between each point was calculated. Similarly, the absolute distance between coordinate points measured by the μ CMM was also calculated. These values were then compared, with an error between the calculated and measured displacement shown in Table 8.1. Additional distances were used for calculation, by determining the distance between the first and third, and first and fourth measurement point, denoted by 1–3 and 1–4, respectively.

Table 8.1: 3MS measurement experiment results.

Distance [mm]	Points				
	1–2	2–3	3–4	1–3	1–4
Measured	19.11	8.69	8.11	27.76	35.87
Calculated	20.72	11.28	8.37	31.55	39.59
Error	1.61	2.59	0.27	3.78	3.73

These error values were unsatisfactory, with an error of up to 2.58 mm for the calculated displacement and up to 3.78 mm for the accumulated measurement. However, these results show a significant improvement over the results achieved for the gauge block measurement. There was also potential to decrease the error displayed in Table 8.1 by modifying the μ CMM parameters. Because of the nonlinear movement fashion of the laser relative to the probe point, the dead distance is also nonlinear (Section 7.2). Therefore an average value was taken for the dead distance, and since the change of dead distance affected both the calculation of the β_{in} and b_{in} value, averages were taken for those values as well.

As a result, it was decided to employ a method that would describe the d , β_{in} , and b_{in} as a function of laser displacement as an aim to improve the error. This was achieved by using a linear equation for each arm and variable that was a function of the laser distance. These were then implemented into the post-processing of the 3MS measurement experiment, with the results shown in Table 8.2. However, this yielded unsuccessful results, with the error showing an increase for all measurement points.

These findings further suggest that thermal error compensation would be a redundant application to the μ CMM because the errors experienced as a combination of the drawbacks found are far too large for the thermal compensation to affect it.

Table 8.2: 3MS results with variable machine parameters.

Distance [mm]	Points				
	1-2	2-3	3-4	1-3	1-4
Calculated	21.03	12.15	8.96	32.69	41.38
Error	1.92	3.45	0.85	4.92	5.51

Chapter 9

Conclusion

With the increasing need for high accuracy CMM's realized, Rugbani (2014) from Stellenbosch University undertook a task of designing and manufacturing a novel parallel mechanism μ CMM. The aim was to provide a more cost-effective μ CMM with a higher measurement range that will allow more consumers the ability to obtain and use a CMM. Furthermore, with the lack of literature supporting high accuracy CMM's, a goal was to develop literature to allow more researchers and designers to develop competitive CMM's to push the industry further. Therefore, the project undertook the task of investigating whether an existing μ CMM reaches its specified accuracy. If so, the accuracy of the μ CMM can be increased by using thermal error compensation, since temperature changes on the structure can contribute up to 70 % of measurement error. This could provide a competitive and accurate μ CMM. As a result, project objectives were set to help the accuracy measurement process and to achieve thermal error compensation along with motivation for the project.

A thorough literature study was performed to ensure that the correct methods for investigating the accuracy of the μ CMM were used. Furthermore, the study included methods of improving accuracy through calibration and thermal error compensation along with the appropriate equipment needed to achieve this. Given the literature on thermal error compensation, a simple error model was derived for the μ CMM system. This error model would calculate the effects of a change in temperature on the probe position. Indeed, the thermal error model motivated the need for thermal error compensation in order to improve the accuracy of the μ CMM based on the specified 10 μm accuracy.

Initial testing of the μ CMM showed that there was still some software development necessary in order for it to work correctly. As a result, additions were made to the software, along with changes to other parts to ensure a more reliable, repeatable, and efficient processing and control of the μ CMM.

A measurement test was performed on a 100 mm gauge block as a first test to examine the accuracy state of the μ CMM. The gauge block was chosen to allow the use of a simpler straight line post-processing as opposed to more complex shapes, such as a sphere. Post-processing of the measurement showed unfavourable results with the original calibrated parameters. Nonetheless, these parameters were tweaked in an attempt to improve the accuracy. However, with the results not showing improvement, an approach was taken where the method for calibration could be simplified in order to allow for better determining of these parameters.

Therefore, a numerical calibration method was applied to the vertical height of each pivot point of the μ CMM and including different optimization techniques. The newly determined parameters showed improvements over the original parameters, however, the desired accuracy of the machine was still not reached. This changed the scope of the project from the original focus on thermal error compensation to focus on calibration of the μ CMM. During the calibration process, many crucial discoveries were made about the design and behaviour of the machine, such as: (1) the machine required large displacements for the laser measurement to change significantly, (2) the ratio of probe movement to laser measurement was 20:1, and (3) the tetrahedron has a position settling error. These behaviours explained some of the errors experienced and gave a better insight into the approach to calibration. The project also used other calibration methods, with unsuccessful results, before using the physical calibration of the tetrahedron with another CMM.

Finally, a physical calibration was done for the μ CMM which allowed for the machine parameters to be correctly determined. Findings on the calibration showed that none of the laser vector paths intersected with each other or the probe point, which was the aim of the initial design of the system. Consequently, the vector paths were changed to allow intersection through the probe point, as this was believed to be the most appropriate point of intersection. These parameters were then applied to the measurement of a 100 mm gauge block and showed an improvement over the original parameters, yet still not meeting the desired accuracy requirements. However, during the process of determining the b_{in} value, an unexpected laser reading behaviour was observed, which invalidated the 100 mm gauge block post processing.

With the requirement of large displacements, a simpler test was devised where it could be ensured that the pivot point will have a large displacement and the probe point could be measured simultaneously. Hence, the 3MS measurement experiment was executed. This test showed favourable results, considering the newly found drawbacks of the system, with a maximum error of approximately 3 mm.

A recommendation is made to investigate design improvements of the μ CMM which address the drawbacks mentioned in this report. If this can be successfully achieved, then the next process of thermal error compensation can be followed.

Appendices

Appendix A

Operation manual

For successful and safe operation of the μ CMM, the operation manual and safety instructions should be strictly adhered to. Operation instructions for the μ CMM are as follows:

1. Ensure that all three lasers are connected to the USB hub (Figure A.1).



Figure A.1: USB hub for connecting lasers and motors.

2. Ensure that all three motors are connected via the serial communication cable (Figure A.2) and the primary motor is connected to the USB hub.



Figure A.2: Serial connection between motors.

3. Ensure that the power supplies for both the motor and the lasers are connected to the power source (Figure A.3).



Figure A.3: Power supply.

4. Ensure that there is a connection between the Arduino[®] and Renishaw[®] probe reader (Figure A.4).

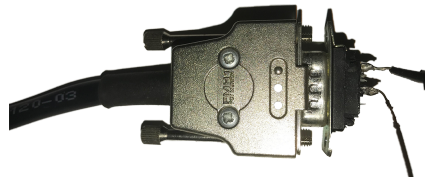


Figure A.4: Connection between Arduino[®] and probe reader.

5. Ensure that there is a connection between the Arduino[®] and the USB port on the computer in use.
6. Plug the main adapter plug into a power source.
7. Remove motor height brackets (Figure A.5).



Figure A.5: Motor height bracket.

8. Ensure the μ CMM structure is elevated with a spacer bracket (Figure A.6).



Figure A.6: Structure spacer bracket.

9. Ensure that the probe is attached to the tetrahedron and plugged into the probe reader.
10. Enter the device manager on the computer and ensure that the port settings are set as follows:
 - Motor - COM 11,
 - Laser_a - COM 5,
 - Laser_b - COM 6,
 - Laser_c - COM 12, and
 - Arduino[®] - COM 4.
11. Enter the ILD1420 tool and connect to laser_c on COM 12 once to take the laser out of analogue mode, and twice to begin and confirm measurement. Disconnect once confirmed.
12. Enter the ILD1302 tool to confirm measurement on laser_a and laser_b on COM 5 and COM 6, respectively.
13. Enter the Zaber[®] console, connect to COM 11 and click the "home" button.
14. Disconnect once motors have reached the bottom position and are no longer moving.
15. Run the pre-sequence of the μ CMM with the file called "presequence.vi" while connected to COM 11. Once the pre-sequence is completed, close the program.
16. Run the main programme, "main.vi", and connect to all serial devices.
17. Place a measurement specimen in the work area of the μ CMM.
18. Perform the measurement of the specimen.
19. Once measurement is completed, remove the specimen, disconnect from all serial devices, and close the programme.
20. Set motor heights back to the default encoder count of 247500.

21. Insert motor height brackets.
22. Unplug the system from the power source.

Appendix B

Optimization code

```

1 % Measured values from height gauge
2 dm = [0, 3.16, 5.79, 9.15, 12.4, 15.05, 17.93, 20.03,
        23.79, 26.27, 29.95]';
3 % Number of iterations to run algorithm
4 N = 100;
5
6 for i = 1:N
7 % X(1) = Dead distance of arm i, X(2) = Dead distance of
   arm in, X(3) = beta between i and in, X(4) = b
   between i and in
8 [X,Fval] = fmincon(@(x) sum(objFun(li,lin,x(1),x(2),x(3)
   ,x(4),dm).^2),...
9   [373 373 73 480], [], [], [], [], [], [],
   options)
10 [X,Fval] = ga(@(x) sum(objFun(li,lin,x(1),x(2),x(3),x(4)
   ,dm).^2),4,...
11   [],[],[],[],[300 300 60 400],[400 400 80 500]);
12 % Putting the optimized parameters into Equations 2.3 to
   2.8 to calculated the z height and then storing the
   answer in an array for each iteration
13 dc = (li + vec.*X(1)).^2 + (lin + vec.*X(2)).^2 -...
14   2.*(li + vec.*X(1)).*(lin + vec.*X(2)).*cosd(X(3));
15 dzc(:,i) = dc - vec.*(X(4))^2;
16 dc1 = sqrt(dzc);
17 err(:,i) = dc1(:,i) - dm;
18 % calculating the standard deviation of the error
19 dev(i) = std(err(:,i));
20 Xo(i,:) = X;
21 Fo(i) = Fval;

```



```
22 i = i + 1;
23 end
24
25 % Finding the set with the lowest standard deviation of
    the error, hence optimal results
26 [Y,I] = min(dev)
27 dc1(:,I)
28 err(:,I)
29 Xo(I,:)
30 Fo(I)
```

Appendix C

Post calibration processing

```
1 %% CMM translated to [0,0,0] probe point
2
3 close all
4 clear all
5 clc
6
7 set(0, 'defaulttextinterpreter', 'latex');
8 set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
9 set(groot, 'defaultLegendInterpreter', 'latex')
10
11 %% Read data
12 % Values from calibration with another CMM
13 [x,y,z,d,r] = textread('CMM.csv', '%f', '%f', '%f', '%f', '%f');
14 C = [x,y,z];
15
16 % Measurement data from 3MS experiment
17 [st, p, m, d, la, lb, lc] = textread('micro1m.csv', '%d',
    '%d', '%d', '%d', '%f', '%f', '%f');
18 L = [st, p, m, d, la, lb, lc] ;
19 s = size(L);
20 %% Variables
21
22 la1 = 50.793;
23 lb1 = 136.507;
24 lc1 = 139.025;
25 la2 = 12.3437;
26 lb2 = 2.43262;
27 lc2 = 1.7608;
28
```

```

29 probe = C(7,:);
30 pa1 = C(1,:) - probe;
31 pb1 = C(2,:) - probe;
32 pc1 = C(3,:) - probe;
33 pa2 = C(4,:) - probe;
34 pb2 = C(5,:) - probe;
35 pc2 = C(6,:) - probe;
36 x = x - probe(1);
37 y = y - probe(2);
38 z = z - probe(3);
39 probe = probe - probe;
40 som = 60;
41
42
43 %% Plot lines
44 hfig = figure();
45 % Plot points
46 plot3(x(1:3),y(1:3),z(1:3),'.b','markersize',20)
47 hold on
48 plot3(x(4:6),y(4:6),z(4:6),'.g','markersize',20)
49 plot3(x(7),y(7),z(7),'.r','markersize',20)
50
51 % Getting vector of lines
52 va = (pa1 - pa2)/norm(pa1 - pa2);
53 vb = (pb1 - pb2)/norm(pb1 - pb2);
54 vc = (pc1 - pc2)/norm(pc1 - pc2);
55 t = linspace(0,600,10000);
56 ra = @(t) pa1 - t'.*va;
57 ral = ra(t);
58 rb = @(t) pb1 - t'.*vb;
59 rbl = rb(t);
60 rc = @(t) pc1 - t'.*vc;
61 rcl = rc(t);
62
63 grid on
64 % Plot lines
65 plot3(ral(:,1),ral(:,2),ral(:,3),'linewidth',1)
66 plot3(rbl(:,1),rbl(:,2),rbl(:,3),'linewidth',1)
67 plot3(rcl(:,1),rcl(:,2),rcl(:,3),'linewidth',1)
68 xlabel('x-axis [mm]')
69 ylabel('y-axis [mm]')
70 zlabel('z-axis [mm]')
71 legend('Max distance','Min distance','Probe','$r_a$','$r_b$',
        '$r_c$',...

```

```

72     'Location', 'northwest')
73
74 set(hfig, 'Position', [900 300 500 300])
75 print -depsc2 cmmvec.eps
76 hold off
77
78 % Value determined from Equation 7.2
79 probe2 = [195.5720, -1.3245, 230.3101];
80
81 %% Dead Distances from actual probe
82 % -60mm for start of laser measurement
83 da1 = norm(pa1 - probe) - la1 - 60;
84 da2 = norm(pa2 - probe) - la2 - 60;
85 db1 = norm(pb1 - probe) - lb1 - 60;
86 db2 = norm(pb2 - probe) - lb2 - 60;
87 dc1 = norm(pc1 - probe) - lc1 - 60;
88 dc2 = norm(pc2 - probe) - lc2 - 60;
89
90 % Equations used for variable distances
91 ma = (da1 - da2)/(la1 - la2); % m = (y1 - y2)/(x1 - x2)
92 ca = da1 - ma*la1; % c = y - m*x
93 ya = @(xa) ma*xa + ca; % ya = 0.026627049789461*xa +
    3.136353928514989e+02
94
95 mb = (db1 - db2)/(lb1 - lb2); % m = (y1 - y2)/(x1 - x2)
96 cb = db1 - mb*lb1; % c = y - m*x
97 yb = @(xb) mb*xb + cb; % yb = -0.044612765097454*xb +
    3.158420181264962e+02
98
99 mc = (dc1 - dc2)/(lc1 - lc2); % m = (y1 - y2)/(x1 - x2)
100 cc = dc1 - mc*lc1; % c = y - m*x
101 yc = @(xc) mc*xc + cc; % yc = -0.013374610524288*xc +
    3.103580364947039e+02
102
103 D1 = [da1, db1, dc1; da2, db2, dc2]
104
105 %% Dead distances from theoretical probe
106 da1 = norm(pa1 - probe2) - la1 - 60;
107 da2 = norm(pa2 - probe2) - la2 - 60;
108 db1 = norm(pb1 - probe2) - lb1 - 60;
109 db2 = norm(pb2 - probe2) - lb2 - 60;
110 dc1 = norm(pc1 - probe2) - lc1 - 60;
111 dc2 = norm(pc2 - probe2) - lc2 - 60;
112

```

```

113 D2 = [da1, db1, dc1; da2, db2, dc2]
114
115 %% calculate angles between tetrahedron with actual
    probe point
116 va1 = (pa1 - probe)/norm(pa1 - probe);
117 vb1 = (pb1 - probe)/norm(pb1 - probe);
118 vc1 = (pc1 - probe)/norm(pc1 - probe);
119
120 betaab1 = angles(va1, vb1);
121 betabc1 = angles(vb1, vc1);
122 betaca1 = angles(vc1, va1);
123
124
125 va2 = (pa2 - probe)/norm(pa2 - probe);
126 vb2 = (pb2 - probe)/norm(pb2 - probe);
127 vc2 = (pc2 - probe)/norm(pc2 - probe);
128
129 betaab2 = angles(va2, vb2);
130 betabc2 = angles(vb2, vc2);
131 betaca2 = angles(vc2, va2);
132
133 % Equations used for variable distances
134 ma = (betaca1 - betaca2)/(la1 - la2) % m = (y1 - y2)/(x1
    - x2)
135 ca = betaca1 - ma*la1 % c = y - m*x
136 ya = @(xa) ma*xa + ca; % yab = 0.014048554826179*xa +
    68.257395830445560
137
138 mb = (betaab1 - betaab2)/(lb1 - lb2) % m = (y1 - y2)/(x1
    - x2)
139 cb = betaab1 - mb*lb1 % c = y - m*x
140 yb = @(xb) mb*xb + cb; % ybc = -0.002311030128832*xb +
    68.407001290625388
141
142 mc = (betabc1 - betabc2)/(lc1 - lc2) % m = (y1 - y2)/(x1
    - x2)
143 cc = betabc1 - mc*lc1 % c = y - m*x
144 yc = @(xc) mc*xc + cc; % yca = 0.010869972519353*xc +
    68.266075710813610
145
146 beta1 = [betaab1, betabc1, betaca1];
147 beta2 = [betaab2, betabc2, betaca2];
148 %% calculate angles between tetrahedron with theoretical
    probe

```

```

149
150 va1t = (pa1 - probe2)/norm(pa1 - probe2);
151 vb1t = (pb1 - probe2)/norm(pb1 - probe2);
152 vc1t = (pc1 - probe2)/norm(pc1 - probe2);
153
154 betaabt = angles(va1t, vb1t);
155 betabct = angles(vb1t, vc1t);
156 betacat = angles(vc1t, va1t);
157
158 va2t = (pa2 - probe2)/norm(pa2 - probe2);
159 vb2t = (pb2 - probe2)/norm(pb2 - probe2);
160 vc2t = (pc2 - probe2)/norm(pc2 - probe2);
161
162 betaab2t = angles(va2t, vb2t);
163 betabc2t = angles(vb2t, vc2t);
164 betaca2t = angles(vc2t, va2t);
165
166 betat = [betaabt, betabct, betacat];
167 beta2t = [betaab2t, betabc2t, betaca2t];
168
169 beta = [beta1; beta2; (beta1 + beta2)/2;...
170         betat; beta2t; (betat + beta2t)/2]
171 %% Variables
172
173 deadLa = D1(1,1);
174 deadLb = D1(1,2);
175 deadLc = D1(1,3);
176
177 for j = 1:s
178     la(j) = la(j) + som + deadLa;
179     lb(j) = lb(j) + som + deadLb;
180     lc(j) = lc(j) + som + deadLc;
181 end
182
183 betaab = beta(1,1);
184 betabc = beta(1,2);
185 betaca = beta(1,3);
186
187 %% Functions
188
189 % Calculate b distance
190 function B = b_dist(l1, l2, beta)
191 B = sqrt(l1.^2 + l2.^2 - 2.*l1.*l2.*cosd(beta));
192 end

```

```
193
194 % calculate z height
195 function [f] = zcalc(li,ln,beta,B,zi)
196 din = li.^2 + ln.^2 - 2.*li.*ln.*cosd(beta);
197 dzin = sqrt(din - B^2);
198 zn = zi + dzin;
199 f = zn;
200 end
201
202 % Calculate beta angle
203 function B = angles(P1,P2)
204 B = acosd(dot(P1,P2)/(norm(P1)*norm(P2)));
205 end
```

Appendix D

Post-processing code

```
1 %% Post-processing
2
3 close all
4 clear all
5 clc
6
7 [st, p, m, d, la, lb, lc] = textread('Gauge2.csv', '%d, %
    d, %d, %d, %f, %f, %f');
8 x = [st, p, m, d, la, lb, lc] ;
9 s = size(x);
10
11 %% Original calibrated parameters from Rugbani (2014)
12 % Original dead distance
13 % deadLa = 294.0235;
14 % deadLb = 289.9038;
15 % deadLc = 289.3955;
16
17 % Original beta
18 % betaab = 68.50922383;
19 % betabc = 68.73239673;
20 % betaca = 68.35507659;
21
22 %% Calibrated parameters
23 D1 = [314.9879 309.7521 308.4986;...
24       313.9641 315.7335 310.3345];
25 D2 = [313.7082 281.4081 279.0626;...
26       312.4648 287.1042 280.8018];
27 Beta = [68.0915 69.7773 68.9710;...
28         68.4014 68.2852 68.4308;...
```



```

29     68.2465    69.0312    68.7009;...
30     71.0336    74.6816    72.0658;...
31     71.4543    74.9495    71.5526;...
32     71.2440    74.8155    71.8092];
33 v1 =    [-0.6578    0.0019    -0.7532;...
34     0.3050    0.5733    -0.7604;...
35     0.3193    -0.5705    -0.7567];
36
37 som = 60; % start of measuring range for laser
38
39
40 deadLa = D1(1,1);
41 deadLb = D1(1,2);
42 deadLc = D1(1,3);
43
44 betaab = Beta(1,1);
45 betabc = Beta(1,2);
46 betaca = Beta(1,3);
47
48 % resetting original data from labview where incorrect
   dead distances were
49 % added
50
51 % adding the correct dead distances
52 for j = 1:s
53     la(j) = la(j) + som + deadLa;
54     lb(j) = lb(j) + som + deadLb;
55     lc(j) = lc(j) + som + deadLc;
56 end
57
58 % calculating b distance
59 Bab = b_dist(la ,lb ,betaab);
60 Bbc = b_dist(lb ,lc ,betabc);
61 Bca = b_dist(lc ,la ,betaca);
62
63 %% section for finding the smallest b distance during
   the measurement
64 % process. This is to get rid of complex values.
65 [Yab,Iab] = min(Bab);
66 [Ybc,Ibc] = min(Bbc);
67 [Yca,Ica] = min(Bca);
68 Bab = Bab(Iab);
69 Bbc = Bbc(Ibc);
70 Bca = Bca(Ica);

```

```

71
72 % save the final laser lengths
73 laBab = la(Iab);
74 lbBab = lb(Iab);
75 lbBbc = lb(Ibc);
76 lcBbc = lc(Ibc);
77 lcBca = lc(Ica);
78 laBca = la(Ica);
79
80 %% Post processing
81 % s = start or stop
82 % p = probe touch
83 % m = motor number
84
85 d12 = zeros(s(1),1);
86 dz12 = zeros(s(1),1);
87 z2 = zeros(s(1),1);
88
89 d23 = zeros(s(1),1);
90 dz23 = zeros(s(1),1);
91 z3 = zeros(s(1),1);
92
93 d31 = zeros(s(1),1);
94 dz31 = zeros(s(1),1);
95 z1 = zeros(s(1),1);
96
97 %% Post processing of laser measurement to determine the
    z-value after every movement based on which motor
    has moved and in which direction
98 for n = 1:s
99     if n == 1
100         d12(n) = la(n)^2 + lb(n)^2 - 2*la(n)*lb(n)*cosd(
                betaab); % b
101         d23(n) = lb(n)^2 + lc(n)^2 - 2*lb(n)*lc(n)*cosd(
                betabc); % c
102         d31(n) = lc(n)^2 + la(n)^2 - 2*lc(n)*la(n)*cosd(
                betaca); % a
103
104         dz12(n) = sqrt(d12(n) - Bab^2);
105         dz23(n) = sqrt(d23(n) - Bbc^2);
106         dz31(n) = sqrt(d31(n) - Bca^2);
107
108         z2(n) = z1(n) + dz12(n);
109         z3(n) = z1(n) + dz23(n);

```

```

110         z1(n) = z1(n) + dz31(n);
111     end
112
113     if st(n) == 0 % stop
114         if m(n) == 1.
115             z2(n) = z2(n-1);
116             z3(n) = z3(n-1);
117             d31(n) = lc(n)^2 + la(n)^2 - 2*lc(n)*la(n)*
                cosd(beta ca);
118             dz31(n) = sqrt(d31(n) - Bca^2);
119             if d(n) == 1
120                 z1(n) = z3(n) + dz31(n);
121             end
122             if d(n) == -1
123                 z1(n) = z3(n) - dz31(n);
124             end
125             if d(n) == 0 && z1(n) >= z1(n-1)
126                 z1(n) = z3(n) + dz31(n);
127             end
128             if d(n) == 0 && z1(n) < z1(n-1)
129                 z1(n) = z3(n) - dz31(n);
130             end
131         end
132         if m(n) == 2
133             z1(n) = z1(n-1);
134             z3(n) = z3(n-1);
135             d12(n) = la(n)^2 + lb(n)^2 - 2*la(n)*lb(n)*
                cosd(beta ab);
136             dz12(n) = sqrt(d12(n) - (Bab)^2);
137             if d(n) == 1
138                 z2(n) = z1(n) + dz12(n);
139             end
140             if d(n) == -1
141                 z2(n) = z1(n) - dz12(n);
142             end
143             if d(n) == 0
144                 z2(n) = z1(n) + dz12(n);
145             end
146             if d(n) == 0 && z2(n) >= z2(n-1)
147                 z2(n) = z1(n) + dz12(n);
148             end
149             if d(n) == 0 && z2(n) < z2(n-1)
150                 z2(n) = z1(n) - dz12(n);
151             end

```

```

152         end
153         if m(n) == 3
154             z1(n) = z1(n-1);
155             z2(n) = z2(n-1);
156             d23(n) = lb(n)^2 + lc(n)^2 - 2*lb(n)*lc(n)*
                cosd(betabc);
157             dz23(n) = sqrt(d23(n) - Bbc^2);
158             if d(n) == 1
159                 z3(n) = z2(n) + dz23(n);
160             end
161             if d(n) == -1
162                 z3(n) = z2(n) - dz23(n);
163             end
164             if d(n) == 0
165                 z3(n) = z2(n) + dz23(n);
166             end
167             if d(n) == 0 && z3(n) >= z3(n-1)
168                 z3(n) = z2(n) + dz23(n);
169             end
170             if d(n) == 0 && z3(n) < z3(n-1)
171                 z3(n) = z2(n) - dz23(n);
172             end
173         end
174     end
175
176
177     if st(n) > 0 && n > 1
178         if (la(n) == la(n-1)) && (lb(n) == lb(n-1)) &&
            (lc(n) == lc(n-1))
179             z1(n) = z1(n-1);
180             z2(n) = z2(n-1);
181             z3(n) = z3(n-1);
182         else
183             if m(n-1) == 1
184                 z2(n) = z2(n-1);
185                 z3(n) = z3(n-1);
186                 d31(n) = lc(n)^2 + la(n)^2 - 2*lc(n)*la(
                    n)*cosd(betaca);
187                 dz31(n) = sqrt(d31(n) - Bca^2);
188                 if d(n-1) == 1
189                     z1(n) = z3(n) + dz31(n);
190                 end
191                 if d(n-1) == -1
192                     z1(n) = z3(n) - dz31(n);

```

```

193         end
194         if d(n-1) == 0 && z1(n) >= z1(n-1)
195             z1(n) = z3(n) + dz31(n);
196         end
197         if d(n-1) == 0 && z1(n) < z1(n-1)
198             z1(n) = z3(n) - dz31(n);
199         end
200         if p(n) == 1 && d(n) == 1
201             z1(n) = z3(n) + dz31(n);
202         end
203         if p(n) == 1 && d(n) == -1
204             z1(n) = z3(n) - dz31(n);
205         end
206     end
207     if m(n-1) == 2
208         z1(n) = z1(n-1);
209         z3(n) = z3(n-1);
210         d12(n) = la(n)^2 + lb(n)^2 - 2*la(n)*lb(
                n)*cosd(betaab);
211         dz12(n) = sqrt(d12(n) - (Bab)^2);
212         if d(n-1) == 1
213             z2(n) = z1(n) + dz12(n);
214         end
215         if d(n-1) == -1
216             z2(n) = z1(n) - dz12(n);
217         end
218         if d(n-1) == 0 && z2(n) >= z2(n-1)
219             z2(n) = z1(n) + dz12(n);
220         end
221         if d(n-1) == 0 && z2(n) < z2(n-1)
222             z2(n) = z1(n) - dz12(n);
223         end
224         if p(n) == 1 && d(n) == 1
225             z2(n) = z1(n) + dz12(n);
226         end
227         if p(n) == 1 && d(n) == -1
228             z2(n) = z1(n) - dz12(n);
229         end
230     end
231     if m(n-1) == 3
232         z1(n) = z1(n-1);
233         z2(n) = z2(n-1);
234         d23(n) = lb(n)^2 + lc(n)^2 - 2*lb(n)*lc(
                n)*cosd(betabc);

```

```

235         dz23(n) = sqrt(d23(n) - Bbc^2);
236         if d(n-1) == 1
237             z3(n) = z2(n) + dz23(n);
238         end
239         if d(n-1) == -1
240             z3(n) = z2(n) - dz23(n);
241         end
242         if p(n) == 1 && d(n) == 1
243             z3(n) = z2(n) + dz23(n);
244         end
245         if p(n) == 1 && d(n) == -1
246             z3(n) = z2(n) - dz23(n);
247         end
248         if d(n-1) == 0 && z3(n) >= z3(n-1)
249             z3(n) = z2(n) + dz23(n);
250         end
251         if d(n-1) == 0 && z3(n) < z3(n-1)
252             z3(n) = z2(n) - dz23(n);
253         end
254     end
255 end
256 end
257 end
258
259
260 Z = [z1, z2, z3];
261 D = zeros(s(1,1),3);
262
263 for i = 1:s(1,1)
264     D(i,:) = [sqrt(d12(i)) sqrt(d23(i)) sqrt(d31(i))];
265     if D(i,1) == 0
266         D(i,1) = D(i-1,1);
267     end
268
269     if D(i,2) == 0
270         D(i,2) = D(i-1,2);
271     end
272
273     if D(i,3) == 0
274         D(i,3) = D(i-1,3);
275     end
276 end
277
278 MD = [min(D(:,1)), min(D(:,2)), min(D(:,3))];

```

```

279 D;
280
281 M = zeros(s(1,1),4);
282 k = 1;
283 for j = 1:s(1,1)
284     M(j,:) = pp(la(j),lb(j),lc(j),z1(j),z2(j),z3(j),p(j)
285                );
286     if(M(j,4) == 1)
287         P(k) = j;
288         k = k + 1;
289     end
290 end
291 [NT,dT] = plot_line4(M(P(1),1:3),M(P(2),1:3),M(P(3),1:3)
292                    ,M(P(5),1:3));
293 [NS,dS] = plot_line(M(P(6),1:3),M(P(8),1:3),M(P(9),1:3))
294 ;
295 % normal vector points for top plane
296 AT = NT(1,1);
297 BT = NT(1,2);
298 CT = NT(1,3);
299
300 %normal vector points for side plane
301 AS = NS(1,1);
302 BS = NS(1,2);
303 CS = NS(1,3);
304
305 %x,y,z coordinates for points on either length of block
306 x1 = M(P(10),1);
307 y1 = M(P(10),2);
308 z1 = M(P(10),3);
309 x2 = M(P(11),1);
310 y2 = M(P(11),2);
311 z2 = M(P(11),3);
312
313 % angle between planes
314 phi = atan2d(norm(cross(NT,NS)),dot(NT,NS))
315
316
317 %angle between vectors and planes
318 theta1 = asin(abs(AT*x1 + BT*y1 + CT*z1)/(sqrt(AT^2 + BT
319          ^2 + CT^2))*...

```

```

319     sqrt(x1^2 + y1^2 + z1^2)));
320 theta1 = rad2deg(theta1)
321
322 theta2 = asin(abs(AS*x2 + BS*y2 + CS*z2)/(sqrt(AS^2 + BS
      ^2 + CS^2)*...
323     sqrt(x2^2 + y2^2 + z2^2)));
324 theta2 = rad2deg(theta2)
325
326 hyp = sqrt((x1 - x2)^2 + (y1 - y2)^2 ...
327     + (z1 - z2)^2);
328
329 L1 = hyp*cosd(theta1);
330 L = L1*cosd(theta2)
331
332
333 function C = pp(la ,lb ,lc ,za ,zb ,zc ,p)
334 r1 = la ;
335 r2 = lb ;
336 r3 = lc ;
337 Z1 = za ;
338 z2 = zb ;
339 z3 = zc ;
340
341 % x- and y-coordinates of each probe pivot point
      determined from using the
342 % b distance and vector slider paths
343 x1 = -2.681534609325000E2;
344 y1 = 0.007745387287500E2;
345 x2 = 1.233141658982500E2;
346 y2 = 2.317902010146450E2;
347 x3 = 1.295628087224600E2;
348 y3 = -2.314925849551000E2;
349
350 % this is to form the new base length i.e. b_in/b_ip
      used in equations (6)
351 % & (7) as the hypotenuse
352 x12 = x1-x2;
353 y12 = y1-y2;
354 x13 = x1-x3;
355 y13 = y1-y3;
356
357 % Equation (8) & (9) where here l is replaced with r
358 % r12 = (r1^2+r2^2-2*r1*r2*cosd(beta));
359 % r13 = (r3^2+r1^2-2*r3*r1*cosd(beta));

```



```

360
361 % Forms part of the first equation after (14) where ci
      is substituted for
362 % the left hand sides of (12), (13) and (14) for la, lb,
      lc
363 c1 = r1^2-x1^2-y1^2-Z1^2;
364 c2 = r2^2-x2^2-y2^2-z2^2;
365 c3 = r3^2-x3^2-y3^2-z3^2;
366
367 % For equation (15) & (16)
368 c21 = c2-c1;
369 c31 = c3-c1;
370
371 % A, B, D and F for equations (19) & (20)
372 A = ((c21/(2*z2))-(c31/(2*z3)))/(x12/z2-x13/z3);
373 B = (y12/z2-y13/z3)/(x12/z2-x13/z3);
374 D = (y12/z2)-B*(x12/z2);
375 F = A*(x12/z2) - c21/(2*z2);
376
377
378 % Actually u, v and w in Rugbani (2014). This is for
      equation (22)
379 k = 1 + B^2 + D^2;
380 l = 2*D*F + 2*x1*B - 2*A*B - 2*y1;
381 m = A^2 + F^2 - 2*x1*A - c1;
382
383 % Equation (22) can be either + or - sqrt(...)
384 Y1 = (-1 +sqrt(l^2-4*k*m))/(2*k);
385 Y2 = (-1 -sqrt(l^2-4*k*m))/(2*k);
386 X1 = A - B*Y1;
387 X2 = A - B*Y2;
388 Z1 = F + D*Y1;
389 Z2 = F + D*Y2;
390
391 x = X1;
392 y = Y1;
393 z = Z1;
394 C = [x, y, z, p]
395 end
396
397 % Plot a long given four points
398 function [normal, d] = plot_line4(p1, p2, p3, p4)
399 normal = cross(p2 - p1, p4 - p3);
400 d = p1(1)*normal(1) + p1(2)*normal(2) + p1(3)*normal(3);

```

```
401 d = -d;
402 x = -300:300; y = -300:300;
403 [X,Y] = meshgrid(x,y);
404 Z = (-d - (normal(1)*X) - (normal(2)*Y))/normal(3);
405 mesh(X,Y,Z)
406 end
407
408 %Plot a line given three points
409 function [normal, d] = plot_line(p1, p2, p3)
410
411 % This function plots a line from three points.
412 % Input arguments:
413 %   p1, p2, p3 eg, p1 = [x y z]
414 %
415 %
416 % Output arguments:
417 % normal: it contains a,b,c coeff , normal = [a b c]
418 % d : coeff
419
420 normal = cross(p1 - p2, p1 - p3);
421 d = p1(1)*normal(1) + p1(2)*normal(2) + p1(3)*normal(3);
422 d = -d;
423 x = -300:300; y = -300:300;
424 [X,Y] = meshgrid(x,y);
425 Z = (-d - (normal(1)*X) - (normal(2)*Y))/normal(3);
426 mesh(X,Y,Z)
427 end
428
429 function B = b_dist(l1, l2, beta)
430 B = sqrt(l1.^2 + l2.^2 - 2.*l1.*l2.*cosd(beta));
431 end
```

Appendix E

Thermal error model code

```
1 %% Thermal error model
2
3 close all
4 clear all
5 clc
6
7 % Here is where the values for the z-coordinates of each
   spherical joint on
8 % the motors is received (z1,z2,z3). r1,r2,r3 are the
   lengths measured by
9 % the laser.
10
11 r1 = 411.344316;
12 r2=    406.3561545;
13 r3=    407.8386567;
14 z1=    400.9875133;
15 z2=    387.2532227;
16 z3=    381.4428078;
17 pz2=1;
18 pz3=1;
19 bita1=  68.5638916;
20 bita2=  68.67709737;
21 bita3=  68.42587616;
22 erx1 = -0.1601968;
23 erx2 = -0.12512656;
24 erx3 = 0.271252905109806
25 ery1 = 0.095144219864;
26 ery2 = -0.059912;
27 ery3 = 0.031501548041;
28
```

```

29 % This section is used to determine the new x and y
    coordinates as the
30 % joint moves up or down. This is based on pg. 81 & 82
    on Ali Rugbani's
31 % report. It shows that the columns of the tetrahedron
    aren't exactly
32 % upright. These equations are also based off eq (10)
    and (11) in Rugbani's
33 % report.
34 x1 = ((0.0075 *z1 + 148.59 -erx1));
35 x2 = ((0.0082 *z2 + -253.75 -erx2));
36 x3 = ((0.0069 *z3 + 140.74 -erx3));
37 y1 = (((0.0201 *z1 + 3.3691) -ery1));
38 y2 = ((0.0192 *z2 + -220.11 -ery2));
39 y3 = ((0.0153 *z3 + -454.53 -ery3));
40
41 % This is to form the new base length i.e. b_in/b_ip
    used in equations (6)
42 % & (7) as the hypotenuse
43 x12 = x1-x2;
44 y12 = y1-y2;
45 x13 = x1-x3;
46 y13 = y1-y3;
47
48 % This is the actual base length used in equations (6) &
    (7)
49 b12 = (x12^2+y12^2);
50 b13 = (x13^2+y13^2);
51
52 % Equation (8) & (9) where here l is replaced with r
53 r12 = (r1^2+r2^2-2*r1*r2*cos(75.5224878140701*pi/180));
54 r13 = (r3^2+r1^2-2*r3*r1*cos(75.5224878140701*pi/180));
55
56 % Forms part of the first equation after (14) where ci
    is substituted for
57 % the left hand sides of (12), (13) and (14) for la, lb,
    lc
58 c1 = r1^2-x1^2-y1^2-z1^2;
59 c2 = r2^2-x2^2-y2^2-z2^2;
60 c3 = r3^2-x3^2-y3^2-z3^2;
61
62 % For equation (15) & (16)
63 c21 = c2-c1;
64 c31 = c3-c1;

```

```

65
66 % A, B, D and F for equations (19) & (20)
67 A = ((c21/(2*z2))-(c31/(2*z3)))/(x12/z2-x13/z3);
68 B = (y12/z2-y13/z3)/(x12/z2-x13/z3);
69 D = (y12/z2)-B*(x12/z2);
70 F = A*(x12/z2) - c21/(2*z2);
71
72 % Actually u, v and w in Rugbani (2014). This is for
    equation (22)
73 k = 1 + B^2 + D^2;
74 l = 2*D*F + 2*x1*B - 2*A*B - 2*y1;
75 m = A^2 + F^2 - 2*x1*A - c1;
76
77 % Equation (22) can be either + or - sqrt(...)
78 Y1 = (-1 +sqrt(l^2-4*k*m))/(2*k);
79 Y2 = (-1 -sqrt(l^2-4*k*m))/(2*k);
80 X1 = A - B*Y1;
81 X2 = A - B*Y2;
82 Z1 = F + D*Y1;
83 Z2 = F + D*Y2;
84 fprintf ('1st point x = %f, y = %f, z = %f\n', X1,Y1,Z1)
85 fprintf ('2nd point x = %f, y = %f, z = %f\n', X2,Y2,Z2)
86 x = real(X1);
87 y = real(Y1);
88 z = real(Z1);
89
90 %% Simple Error Model
91 % Variables
92 format short
93 Lab = 650;
94 Lac = 650;
95 Lbc = 650;
96 ax = -375.277;
97 ay = 0;
98 % Using ... with digits to increase accuracy. Important
    to set the
99 % values to type "double" afterwards because when using
    vpa they are of
100 % type "sym"
101 bx0 = -375.277 + (650*cosd(30));
102 by = 325;
103 cx0 = bx0;
104 cy0 = -325;
105

```

```

106 % User input values to test temperature effects
107 Tab = input('Temperature for side AB\n') + 273.15;
108 Tac = input('Temperature for side AC\n')+ 273.15;
109 Tbc = input('Temperature for side BC\n')+ 273.15;
110 Ta = input('Temperature for upright column a\n')+
    273.15;
111 Tb = input('Temperature for upright column b\n')+
    273.15;
112 Tc = input('Temperature for upright column c\n')+
    273.15;
113 Tla = input('Temperature for arm a\n')+ 273.15;
114 Tlb = input('Temperature for arm b\n')+ 273.15;
115 Tlc = input('Temperature for arm c\n')+ 273.15;
116
117 var = 21;
118 Tab = var + 273.15;
119 Tac = var + 273.15;
120 Tbc = var + 273.15;
121 Ta = var + 273.15;
122 Tb = var + 273.15;
123 Tc = var + 273.15;
124 Tla = var + 273.15;
125 Tlb = var + 273.15;
126 Tlc = var + 273.15;
127
128 % Structure expansion
129 Lab_p = Lab + delta_l(Lab,T0,Tab);
130 Lac_p = Lac + delta_l(Lac,T0,Tac);
131 Lbc_p = Lbc + delta_l(Lbc,T0,Tbc);
132 fprintf('Original:\tLab = %f\tLac = %f\tLbc = %f\nPrime
    :\tLab = %f\tLac = %f\tLbc = %f\n',Lab,Lac,Lbc,Lab_p,
    Lac_p,Lbc_p);
133
134 % Calculate bx from expansion
135 bx = sqrt((Lab_p)^2 - (by-ay)^2) + ax
136 beta = acosd((Lac_p^2 - Lab_p^2 - Lbc_p^2)/(-2*Lab_p*
    Lbc_p))
137 phi = asind((by - ay)/Lab_p)
138
139 theta = phi + beta - 90;
140
141 cy = by - Lab_p*cosd(theta)
142 cx = bx + Lbc_p*sind(theta)
143

```

```

144 fprintf('Original:\tbody = %f\tbody = %f\tbody = %f\nPrime:\tbody = %f\tbody = %f\tbody = %f\n',bx0 , cy0 , cx0 , bx , cy , cx );
145 bx0 = bx0
146 cx0 = cx0
147 xt0 = [ax; bx0; cx0];
148 yt0 = [ay; by; cy0];
149 dt0 = delaunayTriangulation(xt0 , yt0);
150 figure ()
151 triplot(dt0 , 'b');
152 grid on
153 xlabel('x-axis [mm]')
154 ylabel('y-axis [mm]')
155 title('CMM Structure')
156 hold on
157
158 xt = [ax; bx; cx];
159 yt = [ay; by; cy];
160 dt = delaunayTriangulation(double(xt) , double(yt));
161 triplot(dt , 'r');
162 legend('original' , 'deformed')
163 % Vertical error
164 z1_p = delta_l(z1 , T0 , Ta);
165 z2_p = delta_l(z2 , T0 , Tb);
166 z3_p = delta_l(z3 , T0 , Tc);
167
168 z_p = z + (z1_p + z2_p + z3_p)/3;
169
170 %%
171 % Other simple model for planar error just with
    structure expansion
172 x_p = x + (cx-cx0 + bx-bx0)/2;
173 y_p = y + (cy - cy0);
174
175 %%
176 % Planar error with strucure
177 alpha_a = vpa(acosd(sqrt(sym(ay - y)^2)/(sqrt(sym(ax - x)^2 + sym(ay - y)^2))));
178 alpha_b = vpa(acosd(sqrt(sym(by - y)^2)/(sqrt(sym(bx - x)^2 + sym(by - y)^2))));
179 alpha_c = vpa(acosd(sqrt(sym(cy - y)^2)/(sqrt(sym(cx - x)^2 + sym(cy - y)^2))));
180
181 % alpha_a = double(alpha_a32);
182 % alpha_b = double(alpha_b32);

```

```

183 % alpha_c = double(alpha_c32);
184
185 da = (z1_p) - z;
186 db = (z2_p) - z;
187 dc = (z3_p) - z;
188
189 la_p = r1 + delta_l(r1, T0, T1a);
190 lb_p = r2 + delta_l(r2, T0, T1b);
191 lc_p = r3 + delta_l(r3, T0, T1c);
192
193 delta_a = vpa(sqrt(sym(la_p^2 - da^2)));
194 delta_b = vpa(sqrt(sym(lb_p^2 - db^2)));
195 delta_c = vpa(sqrt(sym(lc_p^2 - dc^2)));
196
197 xa_p = vpa(sym((da/la_p)*sind(alpha_a)) + ax);
198 ya_p = vpa(sym((da/la_p)*cosd(alpha_a)) + ay);
199
200 xb_p = vpa(sym((db/lb_p)*sind(alpha_b)));
201 yb_p = vpa(sym((db/lb_p)*cosd(alpha_b)));
202
203 xc_p = vpa(sym((dc/lc_p)*sind(alpha_c)));
204 yc_p = vpa(sym((dc/lc_p)*cosd(alpha_c) + cy));
205
206 x_p = x + (xa_p + xb_p + xc_p)/3 + (cx-cx0 + bx-bx0)/2 ;
207 y_p = y + (ya_p + yb_p + yc_p)/3 + (cy - cy0);
208
209 figure()
210 plot(double(x), double(y), '.b', 'LineWidth', 100)
211 hold on
212 plot(double(x_p), double(y_p), '.r', 'LineWidth', 5)
213 legend('original', 'Final', 'error a', 'error b', 'error c')
214
215 hold off
216 % Print final result
217 fprintf('%f\t%f\t%f\n%f\t%f\t%f\n', x, y, z, x_p, y_p, z_p)
218
219 %% Functions
220
221 function dL = delta_l(l0, T0, Tf)
222 alpha = 23.1e-6;
223 if ((Tf - T0) ~= 0)
224     dL = vpa(sym(alpha*10*(Tf - T0)));
225 else
226     dL = 0;

```


227 **end**

228 **end**

Appendix F

Brute force calibration code

```
1 %% brute force
2 clear all
3 close all
4 clc
5
6 %% DeadOptimGaV4
7 clear all
8 close all
9 clc
10
11 %% Variables
12 [la, lb, lc] = textread('zt10m.csv', '%f', '%f', '%f');
13 s = size(la);
14 dm = [0 5.44 8.17 11.26 15.81 19.54 22.51 26.54 29.58
15       32.50 37.77 39.66...
16       46.75 49.18 51.58; 0 3.49 6.13 10.66 12.45 15.85
17       20.53 27.58 31.37...
18       34.10 39.38 41.08 43.37 46.26 49.18; 0 2.66 6.19
19       8.96 11.75 16.04...
20       19.54 23.19 26.23 30.77 35.64 39.23 43.53 47.87
21       52.04]';
22 sd = size(dm);
23 dm1 = dm(:,1);
24 dm2 = dm(:,2);
25 dm3 = dm(:,3);
26 vec = ones(size(la));
27
28 tic;
29 %% Generating random variables
30 N = 8000000;
```

```

27 rng('shuffle');
28 Da = 350 + (385-350).*rand(N,1);
29 rng('shuffle')
30 Db = 350 + (385-350).*rand(N,1);
31 rng('shuffle')
32 Dc = 350 + (385-350).*rand(N,1);
33 rng('shuffle')
34 betaCA = 65 + (80 - 65).*rand(N,1);
35 rng('shuffle')
36 betaAB = 65 + (80 - 65).*rand(N,1);
37 rng('shuffle')
38 betaBC = 65 + (80 - 65).*rand(N,1);
39 rng('shuffle')
40 bCA = 440 + (490 - 440).*rand(N,1);
41 rng('shuffle')
42 bAB = 440 + (490 - 440).*rand(N,1);
43 rng('shuffle')
44 bBC = 440 + (490 - 440).*rand(N,1);
45 toc
46
47 s1 = size(dm1);
48 s2 = size(dm2);
49 s3 = size(dm3);
50 E = zeros(N,1);
51 Ea = zeros(s1(1),N);
52 Eb = zeros(s2(1),N);
53 Ec = zeros(s3(1),N);
54 X = zeros(N,4);
55 k = 1;
56
57 %% Da, Dc, betaCA, bCA
58 tic;
59 for i = 1:N
60     i
61     [A, A1, B, C, D] = bf3(lc ,la ,Dc(i),Da(i),betaCA(i),
        bCA(i),dm1);
62     if D ~= 0
63         E(k) = A;
64         Ea(:,k) = A1;
65         X(k,:) = C;
66         k = k + 1;
67     end
68 end
69

```

```

70 k = k-1;
71 E = E(1:k);
72 X = X(1:k,:);
73 [Y,Ia] = min(E)
74 X(Ia,:);
75 Dc = X(Ia,1); Da = X(Ia,2); betaCA = X(Ia,3); bCA = X(Ia
    ,4);
76 %% Db, betaAB, bAB
77 [la, lb, lc] = textread('zt9m.csv', '%f', '%f', '%f');
78 % la = la(1:n); lb = lb(1:n); lc = lc(1:n);
79
80 E = zeros(N,1);
81 X = zeros(N,4);
82 k = 1;
83 for i = 1:N
84     i
85     [A, A1, B, C, D] = bf3(la, lb, Da, Db(i), betaAB(i), bAB(
        i), dm2);
86     if D ~ = 0
87         E(k) = A;
88         Eb(:,k) = A1;
89         X(k,:) = C;
90         k = k + 1;
91     end
92 end
93
94 k = k-1;
95 E = E(1:k);
96 X = X(1:k,:);
97 [Y,Ib] = min(E)
98 X(Ib,:);
99 Db = X(Ib,2); betaAB = X(Ib,3); bAB = X(Ib,4);
100
101 %% betaBC, bBC
102 [la, lb, lc] = textread('zt11m.csv', '%f', '%f', '%f');
103 % la = la(1:n); lb = lb(1:n); lc = lc(1:n);
104 E = zeros(N,1);
105 X = zeros(N,4);
106 k = 1;
107 for i = 1:N
108     i
109     [A, A1, B, C, D] = bf3(lb, lc, Db, Dc, betaBC(i), bBC(i),
        dm3);
110     if D ~ = 0

```

```

111         E(k) = A;
112         Ec(:,k) = A1;
113         X(k,:) = C;
114         k = k + 1;
115     end
116 end
117
118 k = k-1;
119 E = E(1:k);
120 X = X(1:k,:);
121 [Y,Ic] = min(E)
122 X(Ic,:);
123 betaBC = X(Ic,3); bBC = X(Ic,4);
124 toc
125 %% Final Variables
126 dc = [Ea(:,Ia) Eb(:,Ib) Ec(:,Ic)]
127 d = [Da Db Dc];
128 beta = [betaCA betaAB betaBC];
129 b = [bCA bAB bBC];
130
131 M = [d beta b]
132
133 %% Funcitons
134 function [eps, err, d, x, r] = bf3(li,lin,Di,Din,beta,b
    ,dzm)
135 r = 1;
136 vec = ones(size(li));
137 dc = (li + vec.*Di).^2 + (lin + vec.*Din).^2 - 2.*(li +
    vec.*Di).*(lin + vec.*Din).*cosd(beta);
138 dzc = sqrt(dc - vec.*b^2);
139 err = dzm - dzc;
140 eps = sum(abs(err));
141 ri = isreal(dzc);
142 if ri == 0
143     r = 0;
144 end
145 d = dzc; % i would like to replace this with (dzm - sqrt
    (dzc)).^2
146 x = [Di, Din, beta, b];
147 end

```

List of References

- Barakat, N., Elbestawi, M. and Spence, A. (2000). Kinematic and geometric error compensation of a coordinate measuring machine. *International Journal of Machine Tools and Manufacture*, vol. 40, no. 6, pp. 833–850.
- Bommakanti, P. (2012). Destined to grow-the portable cmm market. *Quality*, vol. 51, no. 7, p. 22.
- Chanal, H., Duc, E., Ray, P. and Hascoet, J. (2007). A new approach for the geometrical calibration of parallel kinematics machines tools based on the machining of a dedicated part. *International Journal of Machine Tools and Manufacture*, vol. 47, no. 7-8, pp. 1151–1163.
- Dai, G., Bütefisch, S., Pohlenz, F. and Danzebrink, H.-U. (2009). A high precision micro/nano cmm using piezoresistive tactile probes. *Measurement Science and Technology*, vol. 20, no. 8, pp. 1–9.
- Harris, J. and Spence, A. (2004). Geometric and quasi-static thermal error compensation for a laser digitizer equipped coordinate measuring machine. *International Journal of Machine Tools and Manufacture*, vol. 44, no. 1, pp. 65–77.
- Hughes, T.A. (1995). *Measurement and control basics*. Isa.
- ISO 10360-1:2000 (2000). Geometrical product specification - vocabulary. standard, International Organization for Standardization.
- ISO 10360-2:2009 (2009). Geometrical product specification - acceptance and re-verification tests for coordinate measuring machines. Standard, International Organization for Standardization.
- ISO 10360-6:2014 (2014). Geometrical product specification - estimation of errors in computing gaussian associated features. Standard, International Organization for Standardization.
- Jansen, M., te Voert, M. and Haitjema, H. (2010). Calibration of cmm reference spheres using stitching interferometry. In: *Proceedings of euspen International Conference-Delft-June 2010 79V1*.
- Jeon, D., Kim, K., Jeong, J. and Kim, J. (2010). A calibration method of redundantly actuated parallel mechanism machines based on projection technique. *CIRP annals*, vol. 59, no. 1, pp. 413–416.

- Kerlin, T.W. and Johnson, M. (2012). *Practical thermocouple thermometry*. 7th edn. Instrument Society of America.
- Kruth, J.-P., Vanherck, P., Van den Bergh, C. and Schacht, B. (2002). Interaction between workpiece and cmm during geometrical quality control in non-standard thermal conditions. *Precision Engineering*, vol. 26, no. 1, pp. 93–98.
- Küng, A., Meli, F. and Thalmann, R. (2007). Ultraprecision micro-cmm using a low force 3d touch probe. *Measurement Science and Technology*, vol. 18, no. 2, p. 319.
- Lin, Z.-C. and Chen, C.-C. (2001). Collision-free path planning for coordinate measurement machine probe. *International Journal of Production Research*, vol. 39, no. 9, pp. 1969–1992.
- Oiwa, T. (2000). Coordinate measuring machine using parallel mechanism. In: *Proc. 16th IMEKO World Congress*, vol. 8, pp. 211–214.
- Okafor, A.C. and Ertekin, Y.M. (2000). Vertical machining center accuracy characterization using laser interferometer: part 1. linear positional errors. *Journal of Materials Processing Technology*, vol. 105, no. 3, pp. 394–406.
- Pahk, H. and Lee, S. (2002). Thermal error measurement and real time compensation system for the cnc machine tools incorporating the spindle thermal error and the feed axis thermal error. *The International Journal of Advanced Manufacturing Technology*, vol. 20, no. 7, pp. 487–494.
- Park, J.-j., Kwon, K. and Cho, N. (2006). Development of a coordinate measuring machine (cmm) touch probe using a multi-axis force sensor. *Measurement Science and Technology*, vol. 17, no. 9, p. 2380.
- Patnaik, D. and Raxit, K. (2014). An economics of nano technology: The relative importance of governmental funding and filing of patent applications. *Int. Journal of Engineering Research and Applications*, vol. 4, no. 10, pp. 23–43.
- Renishaw[®] (2016). Data sheet - equator[™] 300 versatile gauge. <http://resources.renishaw.com/en/download/data-sheet-equator-300-versatile-gauge-93768>.
- Renishaw[®] (2018). Equator[™] gauging explained.
- Rugbani, A.M. (2014). *The design, kinematics and error modelling of a novel micro-CMM parallel manipulator*. Ph.D. thesis, Stellenbosch: Stellenbosch University.
- Salsbury, J. (2016). Unleash your cmm's potential. *Quality*, vol. 55, no. 13, p. 22.
- Saputra, V., Ong, S. and Nee, A. (2014). Optimum calibration of a parallel kinematic manipulator using digital indicators. *Advances in Manufacturing*, vol. 2, no. 3, pp. 222–230.
- Snyman, J.A. (2005). Practical mathematical optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms. *Interfaces*, vol. 36, no. 6, pp. 613+.

- Sudatham, W., Matsumoto, H., Takahashi, S. and Takamasu, K. (2016). Diagonal in space of coordinate measuring machine verification using an optical-comb pulsed interferometer with a ball-lens target. *Precision Engineering*, vol. 43, pp. 486–492.
- Takamasu, K., Murui, I., Sato, O., Olea, G. and Furutani, R. (2002). Calibration of three dimensional mechanism-novel calibration method for 3 dof parallel mechanism. In: *Industrial Technology, 2002. IEEE ICIT'02. 2002 IEEE International Conference on*, vol. 1, pp. 394–398. IEEE.
- van Zyl, P. (2016). Manual control unit for a coordinate measurement machine.
- Verl, A., Boye, T. and Pott, A. (2008). Measurement pose selection and calibration forecast for manipulators with complex kinematic structures. *CIRP Annals-Manufacturing Technology*, vol. 57, no. 1, pp. 425–428.
- Vermeulen, M., Rosielle, P. and Schellekens, P. (1998). Design of a high-precision 3d-coordinate measuring machine. *CIRP Annals-Manufacturing Technology*, vol. 47, no. 1, pp. 447–450.
- Wampler, C.W., Hollerbach, J.M. and Arai, T. (1995). An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. *IEEE Transactions on Robotics and Automation*, vol. 11, no. 5, pp. 710–724.
- Wu, C.-W., Tang, C.-H., Chang, C.-F. and Shiao, Y.-S. (2012). Thermal error compensation method for machine center. *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 5-8, pp. 681–689.