

# **Investigation and Development of an Advanced Virtual Coordinate Measuring Machine**

A thesis submitted for the degree of Doctor of Philosophy

**Yang Hu**

School of Engineering and Design  
Brunel University

May 2010

## **Abstract**

Dimensional measurement plays a critical role in product development and quality control. With the continuously increasing demand for tighter tolerances and more complex workpiece shapes in the industry, dimensional metrology often becomes the bottleneck of taking the quality and performance of manufacturing to the next level. As one kind of the most useful and powerful measuring instruments, coordinate measuring machines (CMMs) are widely employed in manufacturing industries. Since the accuracy and efficiency of a CMM have a vital impact on the product quality, productivity and manufacturing cost, the evaluation and improvement of CMM performance have always been important research topics since the invention of CMM.

A novel Advanced Virtual Coordinate Measuring Machine (AVCMM) is proposed against such a background. The proposed AVCMM is a software package that provides an integrated virtual environment, in which user can plan inspection strategy for a given task, carry out virtual measurement, and evaluate the uncertainty associated with the measurement result, all without the need of using a physical machine. The obtained estimate of uncertainty can serve as a rapid feedback for user to optimize the inspection plan in the AVCMM before actual measurement, or as an evaluation of the result of a performed measurement. Without involving a physical CMM in the inspection planning or evaluation of uncertainty, the AVCMM can greatly reduce the time and cost needed for such processes. Furthermore, as the package offers vivid 3D visual representation of the virtual environment and supports operations similar to a physical CMM, it does not only allow the user to easily plan and optimise the inspection strategy, but also provide a cost-effective, risk-free solution for training CMM operators.

A modular, multitier architecture has been adopted to develop the AVCMM system,

which incorporates a number of functional components covering CMM and workpiece modelling, error simulation, inspection simulation, feature calculation, uncertainty evaluation and 3D representation. A new engine for detecting collision/contact has been developed and utilized, which is suitable for the virtual environment of simulated CMM inspections. A novel approach has been established to calculate errors required for the error simulation, where the data are obtained from FEA simulations in addition to conventional experimental method. Monte Carlo method has been adopted for uncertainty evaluation and has been implemented with multiple options available to meet different requirements.

A prototype of the proposed AVCMM system has been developed in this research. Its validity, usability and performance have been verified and evaluated through a set of experiments. The principles for utilising the AVCMM in practical use have also been established and demonstrated.

The results have indicated that the proposed AVCMM system has great potentials to improve the functionalities and overall performance of CMMs.

## **Acknowledgements**

I would first like to express my deepest gratitude to my PhD supervisor, Dr Qingping Yang, for being supportive, inspiring and encouraging throughout the progress of this research whilst allowing me the freedom to work in my own way. The completion of this work would not have been possible without his invaluable advice and guidance.

Many thanks go to my colleagues and friends in Brunel University and especially in the AMEE research group, who have kindly shared their knowledge and experience with me during my research. Special thanks go to Dr Xizhi Sun, who has greatly helped me with her intelligence and patience, and has given me many valuable and useful suggestions.

I am very grateful for the financial support to this PhD project provided by ORSAS and the School of Engineering and Design of Brunel University. My appreciation also goes to NPL, for their professional and generous assistance in my experiments.

I would like to deeply thank my parents, for their unconditional love, support and understanding. And last but definitely not least, there is simply no word that can adequately express my gratefulness and thankfulness to Na Tan, for her selfless love, motivating encouragement and for being the most wonderful wife.

## Table of Contents

Abstract .....	ii
Acknowledgements.....	iv
Table of Contents .....	v
Acronyms .....	x
Nomenclature .....	xiii
List of Figures.....	xv
List of Tables .....	xix
Chapter 1      Introduction.....	1
1.1    Research background .....	1
1.1.1    CMM and their applications .....	2
1.1.2    Overview of VCMM .....	3
1.1.3    Characteristics and advantages of VCMM.....	4
1.1.4    Applications and limitations of existing VCMMs .....	5
1.2    Research motivation, aim and objectives .....	6
1.3    Thesis structure .....	7
Chapter 2      Literature Review.....	9
2.1    Introduction.....	9
2.2    Coordinate measuring machines .....	9
2.2.1    Introduction.....	9
2.2.2    CMM systems and configurations .....	12
2.2.3    CMM measuring systems and probing systems .....	16

2.2.4	CMM inspection planning .....	21
2.2.4.1	Accessibility analysis.....	23
2.2.4.2	Workpiece and probe setup.....	27
2.2.4.3	Path planning.....	30
2.2.4.4	Automation and intelligent planning system .....	35
2.2.5	Evaluation of measurement uncertainty .....	39
2.2.5.1	GUM uncertainty framework.....	40
2.2.5.2	Supplement 1 to the GUM: A Monte Carlo method approach .....	47
2.2.5.3	Uncertainty evaluation for CMM measurements .....	50
2.3	Virtual coordinate measuring machine.....	57
2.3.1	Introduction.....	57
2.3.2	Uncertainty evaluation oriented VCMMs .....	58
2.3.3	Inspection oriented VCMMs .....	61
2.4	Virtual Reality Modelling Language .....	63
2.4.1	VRML features .....	64
2.4.2	X3D.....	64
2.4.3	Collision detection in VRML .....	65
2.5	Summary.....	66
Chapter 3	Proposed AVCMM and Methodology.....	68
3.1	Introduction.....	68
3.2	Overview of AVCMM .....	69
3.3	Work flow of AVCMM.....	71
3.4	Architecture of AVCMM .....	76
3.4.1	Multitier architecture.....	76

3.4.2	AVCMM system architecture .....	78
3.5	Summary.....	80
Chapter 4	Design of AVCMM .....	82
4.1	Introduction.....	82
4.2	Geometrical and kinematical design of CMM model .....	82
4.3	Collision detection .....	89
4.3.1	3D collision detection algorithms .....	90
4.3.2	Object-to-object collision detection interface.....	92
4.3.3	Design of a collision detection engine for AVCMM.....	94
4.4	Error modelling .....	98
4.4.1	CMM geometric error modelling .....	99
4.4.1.1	Volumetric error modelling .....	99
4.4.1.2	Geometric error modelling using FEM.....	106
4.4.2	Probe error modelling.....	107
4.4.3	Error arising from random effects.....	111
4.5	Data processing and surface fitting .....	111
4.6	Monte Carlo simulation based uncertainty evaluation .....	112
4.7	Summary.....	116
Chapter 5	Implementation of a Prototype of AVCMM .....	117
5.1	Introduction.....	117
5.2	Development environment and runtime platforms .....	117
5.3	VRML modelling .....	118
5.3.1	CMM modelling.....	118
5.3.1.1	Model structure.....	119

5.3.1.2	Interface abstraction.....	121
5.3.2	Workpiece modelling .....	122
5.4	VRML parser .....	125
5.5	Virtual CMM module .....	129
5.5.1	Workpiece loader .....	129
5.5.2	Movement control.....	130
5.5.3	Collision and contact detection.....	132
5.5.4	Error simulation .....	133
5.5.4.1	CMM geometric error simulation.....	133
5.5.4.2	Probe error simulation .....	138
5.5.4.3	Simulation of error arising from random effects.....	145
5.6	Uncertainty evaluation .....	147
5.7	User interface.....	150
5.7.1	User interface design.....	150
5.7.2	3D representation .....	151
5.7.3	Virtual Console .....	152
5.7.3.1	Message window .....	152
5.7.3.2	Virtual controllers.....	152
5.7.4	Uncertainty evaluation UI .....	153
5.8	Web based user interface .....	153
5.9	Extension interface.....	155
5.10	Applications of AVCMM.....	157
5.11	Summary.....	158
Chapter 6	Validation, Results and Discussions.....	159



6.1	Introduction.....	159
6.2	Experiment objectives .....	159
6.3	Experiment configurations and setups .....	160
6.4	Experiment procedure .....	164
6.5	Results and discussions .....	170
6.6	Summary.....	183
Chapter 7	Conclusions and Future Work.....	184
7.1	Conclusions.....	184
7.2	Contributions to knowledge.....	186
7.3	Recommendations for future work.....	187
	References .....	189
	Appendices .....	202
Appendix A	Measurement results .....	203
Appendix B	Segments of programs .....	219
Appendix C	List of publications .....	226

## Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
AABB	Axis-Aligned Bounding Box
AI	Artificial Intelligence
API	Application Programming Interface
AVCMM	Advanced Virtual Coordinate Measuring Machine
BIPM	Bureau International des Poids et Mesures
BLL	Business Logic Layer
B-Rep	Boundary Representation
CAD	Computer-Aided Design
CAI	Computer Aided Inspection
CAM	Computer Aided Manufacturing
CCD	Charge-Coupled Device
CIM	Computer Integrated Manufacturing
CIPM	Comité International des Poids et Mesures
CLI	Common Language Infrastructure
CLR	Common Language Runtime
CMM	Coordinate Measuring Machine
DAL	Data Access Layer
DEA	Digital Electric Automation

DMIS	Dimensional Measurement Interface Specification
DOF	Degree of Freedom
DRO	Digital Read Out
EXINS	Expert Inspection System
FEA	Finite Element Analysis
FEM	Finite Element Method
GA	Genetic Algorithm
GAC	Global Accessibility Cone
GUF	GUM Uncertainty Framework
GUM	Guide to the Expression of Uncertainty in Measurement
HVCMM	Haptic Virtual Coordinate Measuring Machine
IEC	International Electrotechnical Commission
IFCC	International Federation for Clinical Chemistry and Laboratory Medicine, formerly International Federation of Clinical Chemistry
ILAC	International Laboratory Accreditation Cooperation
IMS	International Metrology Systems
IUPAC	International Organization for Pure and Applied Chemistry, formerly International Union of Pure and Applied Chemistry
IUPAP	International Organization for Pure and Applied Physics, formerly International Union of Pure and Applied Physics
LAC	Local Accessibility Cone
MCM	Monte Carlo Method
MCR	MATLAB Compiler Runtime
MV	Machine Vision
MVC	Model, View and Controller

MZM	Minimum Zone Method
NML	Neutral Messaging Language
NPL	National Physical Laboratory
OBB	Oriented Bounding Box
OIML	International Organization of Legal Metrology
PDF	Probability Density Function
PL	Presentation Layer
PTB	Physikalisch-Technische Bundesanstalt
QTC	Quick Turnaround Cell
SBC	Simulation by Constraints
THAIP&P	Tsinghua Automated Inspection Planning and Programming System
UI	User Interface
UML	Unified Modeling Language
VCMM	Virtual Coordinate Measuring Machine
VRML	Virtual Reality Modelling Language
VT	Virtual Table
XML	eXtensible Markup Language

## Nomenclature

$A$	tilting angle of probe head
$B$	rolling angle of probe head
$L$	translational matrix, stylus length
$M$	number of Monte Carlo trials
$O$	algorithm time complexity
$R$	rotational matrix
$S_{xy}, S_{xz}, S_{yz}$	squareness errors
$T$	rotational matrix, transformation matrix
$U$	expanded uncertainty
$(X, Y, Z)$	coordinates
$c_i$	sensitivity coefficients
$f$	functional relationship
$k$	coverage factor
$n_{dig}$	number of significant decimal digits
$p$	coverage probability, level of confidence
$r(y_i, y_j)$	correlation coefficient
$s(X_i)$	experimental standard deviation
$s^2(X_i)$	experimental variance
$u(x_i)$	standard uncertainty
$u(x_i, x_j)$	covariance

$u^2(x_i)$	variance
$u_c(y)$	combined standard uncertainty
$\hat{v}$	probing direction
$v_m$	measurement speed
$(x, y, z)$	coordinate system
$[y_{low}, y_{high}]$	coverage interval
$\delta$	numerical tolerance
$\delta_x(x), \delta_y(y), \delta_z(z)$	positional errors
$\delta_y(x), \delta_z(x), \delta_x(y), \delta_z(y),$ $\delta_x(z), \delta_y(z)$	straightness errors
$\varepsilon_x(x), \varepsilon_y(x), \varepsilon_z(x), \varepsilon_x(y),$ $\varepsilon_y(y), \varepsilon_z(y), \varepsilon_x(z), \varepsilon_y(z),$ $\varepsilon_z(z)$	rotational errors
$\theta$	latitude
$\phi$	longitude

## List of Figures

Figure 1.1 Examples of typical CMM applications .....	3
Figure 2.1 A typical Bridge CMM .....	10
Figure 2.2 Examples of CMM types .....	13
Figure 2.3 Ten common CMM configurations .....	15
Figure 2.4 Different types of probes .....	20
Figure 2.5 Accessibility cone in three dimensions (Lim & Menq, 1994) .....	25
Figure 2.6 The HVCMM system configuration (Chen et al., 2004) .....	26
Figure 2.7 Functional combination of configurable fixture (Cheng & Cai, 1995) .	28
Figure 2.8 Logic of the probe and workpiece setup planner utilizing feature accessibility algorithm (Ziemian & Medeiros, 1998).....	29
Figure 2.9 Structure of a simple genetic algorithm.....	32
Figure 2.10 Variation of flatness error with respect to sample size for: (a) Ra 9.0 µm; (b) Ra 3.5 µm; (c) Ra 0.2 µm.....	34
Figure 2.11 Probe path generated for inspecting a workpiece in an experiment (Furutani et al., 1994) .....	35
Figure 2.12 Flow chart of the generic algorithm (Lin et al., 2001) .....	36
Figure 2.13 Conceptual CMM probe path generation system (Lin et al., 2001) ....	37
Figure 2.14 Vision guided CMM (Qin et al., 2008).....	38
Figure 2.15 Overall scheme of ECMM (Balsamo et al., 1999) .....	52
Figure 2.16 Relationship between number of data and uncertainty of diameter and X coordinate of center in uniform measurment .....	53

Figure 2.17 Space frame .....	54
Figure 2.18 Scheme of the error simulation method.....	56
Figure 2.19 MVC model for VCMM .....	61
Figure 3.1 Conceptual model of proposed AVCMM .....	69
Figure 3.2 Work flow of the proposed AVCMM .....	73
Figure 3.3 A typical three-tier architecture.....	77
Figure 3.4 N-tier architecture of AVCMM .....	79
Figure 4.1 Five models of CMM .....	84
Figure 4.2 An example of bad VRML model design .....	85
Figure 4.3 VRML model designs for five types of CMM.....	88
Figure 4.4 A probe model .....	88
Figure 4.5 VRML model design for moving bridge CMM with 2-axis probe .....	89
Figure 4.6 AABB for a probe at different orientations.....	96
Figure 4.7 OBB for a probe at a tilted orientation .....	96
Figure 4.8 Tip correction vector.....	97
Figure 4.9 Determination of more accurate contact position .....	98
Figure 4.10 Six geometric errors along X axis .....	100
Figure 4.11 Latitude $\theta$ and longitude $\phi$ .....	108
Figure 4.12 Machine and probe coordinate systems .....	108
Figure 4.13 Flow chart of the adaptive Monte Carlo procedure.....	114
Figure 5.1 (a) FN503 CMM; (b) PH9 probe head and TP2 probe.....	119
Figure 5.2 VRML model design for AVCMM prototype .....	120
Figure 5.3 A minimal VRML model for CMM .....	120



Figure 5.4 An improved VRML model for CMM .....	121
Figure 5.5 An instantiated reference ball.....	125
Figure 5.6 Workflow of VRML parser .....	126
Figure 5.7 Virtual controller axes .....	130
Figure 5.8 FEA models for two locations.....	134
Figure 5.9 Meshing for FEA at two locations.....	135
Figure 5.10 FEA results for two locations.....	136
Figure 5.11 Geometric error map.....	138
Figure 5.12 Probe error measurement .....	139
Figure 5.13 Probe error at $A = 0^\circ$ .....	140
Figure 5.14 Probe error for stylus length $L=80$ mm .....	141
Figure 5.15 Measured probe errors ( $\mu\text{m}$ ) for stylus length $L = 30$ mm.....	142
Figure 5.16 Measured probe errors ( $\mu\text{m}$ ) for stylus length $L = 60$ mm.....	143
Figure 5.17 Measured probe errors ( $\mu\text{m}$ ) for stylus length $L = 80$ mm.....	144
Figure 5.18 Distribution of the variations of positions for one point .....	145
Figure 5.19 Distribution of the variations of positions for all 37 points.....	146
Figure 5.20 Running time for different number of points .....	149
Figure 5.21 Main UI of AVCMM .....	150
Figure 5.22 3D representation of virtual CMM world.....	151
Figure 5.23 Predefined viewpoints .....	152
Figure 5.24 Web user interface .....	154
Figure 6.1 Measurement locations (bird's-eye view).....	162
Figure 6.2 Two example setups .....	163

Figure 6.3 Two example setups in virtual inspections .....	164
Figure 6.4 Illustration of reduced measuring length with tilted probe.....	166
Figure 6.5 Part coordinate system for ring gauge .....	167
Figure 6.6 Sampling path (bird's-eye view) .....	168
Figure 6.7 Probe setup in AVCMM.....	168
Figure 6.8 (a) Simplified alignment; (b) part coordinate system for ring gauge in AVCMM .....	169
Figure 6.9 Mean values of measurement results versus AVCMM estimation.....	171
Figure 6.10 Measurement results versus coverage interval predicted by AVCMM .....	175
Figure 6.11 Comparison of simulation performance .....	182

## List of Tables

Table 2.1 Typical CMM accuracy .....	16
Table 2.2 V-COLLIDE API .....	66
Table 3.1 Measurement uncertainty contributors for CMMs .....	74
Table 4.1 Objects Collidee and Collision .....	93
Table 5.1 Materials used in FEA.....	137
Table 5.2 Running time comparison for simulation methods.....	148
Table 5.3 AVCMM API (part 1).....	155
Table 5.4 AVCMM API (part 2).....	156
Table 6.1 Calibration results of the ring gauge .....	161
Table 6.2 Hardware configurations of three computer systems.....	163
Table 6.3 Order of the measurements at 20 setups .....	165
Table 6.4 Recorded temperatures for CMM experiments .....	170
Table 6.5 Plausibility test result for mean values of measurements .....	177
Table 6.6 Plausibility test result for all measurements.....	179
Table 6.7 Plausibility test result after applying adjustment.....	180
Table A-1 Ring gauge bore diameter measurement results .....	218

# **Chapter 1 Introduction**

## **1.1 Research background**

In manufacturing industry, dimensional measurement plays a vital role as it generally provides dimensional information and vital quality assurance about products. The past five decades have seen great development in dimensional metrology. Especially since the invention of Coordinate Measuring Machine (CMM) in 1960s, dimensional measurement has been revolutionised, and CMMs have quickly evolved and become one of the most powerful tools for dimensional inspection. Nowadays, various kinds of advanced CMMs are widely employed in industry. Comparing to conventional inspection methods with complicated and inefficient procedures, CMM greatly simplifies the procedure yet improves the accuracy and precision.

Although the CMMs are vastly useful, in today's rapidly developing manufacturing industry, sometimes its speed and accuracy of inspection can still become the bottleneck of further improving manufacturing technology. Enormous efforts have been made to help CMM keep in pace with the modern industry. Among others, Virtual Coordinate Measuring Machine (VCMM) is the general designation of a range of computer applications that simulate certain aspects of CMM characteristics in one way or another, and typically facilitate the inspection planning on CMMs, or help evaluate the measurement uncertainty associated with the CMM inspection. The work reported in this thesis aims to improve the existing VCMMs significantly, by integrating the different functionalities offered by current solutions into a complete, full-featured system. The proposed system utilises many advanced technologies and innovations, and provides not only comprehensive, but also fast and reliable services covering various aspects of CMM. This newly designed system is named Advanced Virtual Coordinate

Measuring Machine (AVCMM) and is demonstrated and validated by the implementation of a prototype and experimental results.

### **1.1.1 CMM and their applications**

CMMs are metrology instruments designed to measure geometrical characteristics of workpieces. They normally consist of a probe and moving axes to which the probe is attached. CMMs can either be manually controlled by operator or numerically controlled by computer. To date, a number of CMM structures with different forms have been developed to suite various situations, offering various types like free-standing, benchtop or handheld. And in addition to the commonly used contact probes, newer probes are equipped with a variety of different probing technologies, such as laser and vision probes.

To fulfil the industrial needs of accurate inspection in many fields, there is a wide range of CMMs to choose from, ranging from portable models driven by hand to heavy gantry models capable of measuring large objects like cars, providing measuring capacities from smaller than  $200\text{ mm}^3$  to greater than  $10\text{ m}^3$ , and offering accuracies ranging from better than  $1\text{ }\mu\text{m}$  to around  $100\text{ }\mu\text{m}$ . The choice of CMM configurations and size for specific tasks depends on several factors such as accuracy requirement, operation modes, size of the workpiece, and loading requirement (Yang, 1992).

CMMs are widely employed in most areas of modern advanced manufacturing industry, as well as in many other fields that require high quality dimensional inspection, as illustrated in Figure 1.1. Common applications of CMM system include dimensional measurement, angularity or orientation measurement, profile measurement, depth mapping, shaft measurement, digitizing, reverse engineering and rapid prototyping. Powered by fast advancing probing and computing technologies, CMM systems now

can measure almost any type of workpieces, and are especially competent for the inspection of both regular and freeform surfaces of, for example, mould and cast workpieces, vehicle body parts, tool parts, engine components, printed circuit boards, etc.

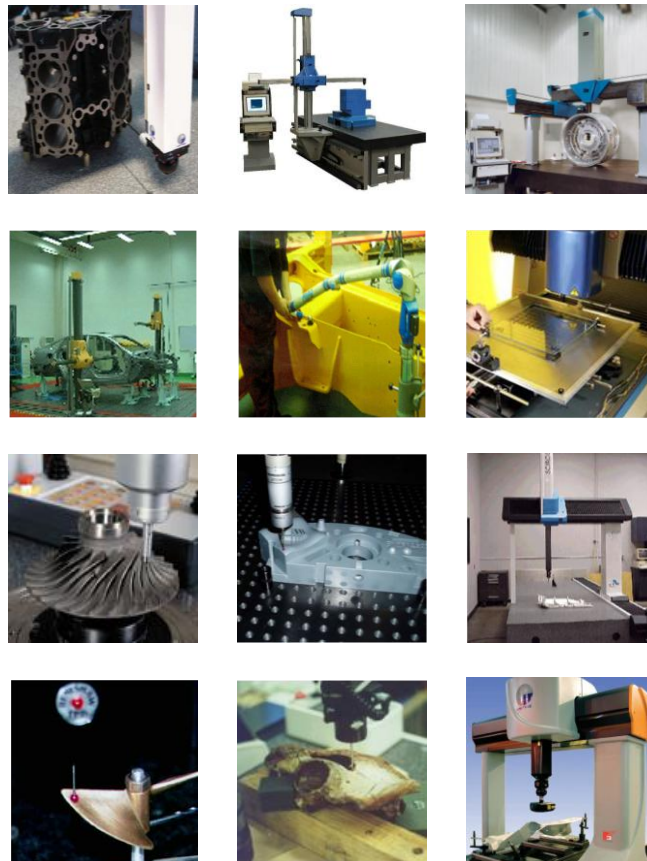


Figure 1.1 Examples of typical CMM applications

### 1.1.2 Overview of VCMM

The concept of Virtual Coordinate Measuring Machine emerged in 1990s as computer programs realizing mathematical models of CMM errors. By constructing measured error components into an error map – which can be considered as an indexed table of

combined errors – these programs are able to predict the volumetric error of an inspection. As time went by, more and more computer applications have been developed to study and aid in various aspects of CMM, and the VCMM concept has evolved to generally refer to computer programs that simulate the behaviours of CMM in one way or another, and provide useful assistance in CMM related tasks.

Existing VCMMs may be roughly categorized into two groups. The first group of VCMMs are basically expansions of the initial VCMM concept, which are tools implemented to analyse errors and uncertainties associated with CMM inspection. The other group of VCMMs focus on the simulation and representation of CMM inspection in cyber space, and allow applications like operation training, intelligent inspection planning, off-line programming, remote monitoring and controlling, etc., to be implemented. Both types of VCMMs have progressed over time and new ideas are still actively being contributed to the concept.

### **1.1.3 Characteristics and advantages of VCMM**

As a computer program, a VCMM typically creates a virtual replica of a physical CMM by modelling certain aspects of the characteristics of the CMM. Results of interest and useful functionalities are then provided based on the model and simulation. Comparing to other experimental methods, VCMMs are normally faster and safer. Experiments on physical CMMs can be time-consuming and expensive, sometimes even risky when manually controlled by new operator, whereas in VCMMs all operations and CMM behaviours are virtualized in computer and therefore the same task can be performed vastly faster and risk-free. The reliability of the results generated by a VCMM can normally be validated by experiment on physical machine, and can be improved by perfecting the model and simulation methods.

### 1.1.4 Applications and limitations of existing VCMMs

As mentioned before, almost all existing VCMMs can be classified as either uncertainty evaluation oriented or inspection oriented. Some of their important applications are listed below:

- Error simulation and prediction, for CMM geometry error, probe error, etc.
- Uncertainty evaluation for CMM measurement result, by taking into consideration CMM and probe errors, workpiece form deviations, inspection strategy and other contributors.
- Online analysis of measurement uncertainty for CMM, by building VCMM into the CMM software package.
- Offline accessibility analysis
- Offline programming of CMM inspection.
- Optimization of CMM inspection plan.
- Remote monitoring of CMM inspection.
- Remote control of CMM inspection.

VCMMs are useful tools and provide various kinds of assistance to CMM related tasks. However the development of VCMM is still at its early stage and inevitably there are many limitations. Most VCMMs only provide functions for certain applications to a



very limited extent. For example, most uncertainty evaluation oriented VCMMs can only perform evaluation of uncertainties caused by a small proportion of possible sources, sometimes only one. Furthermore, as the functions are provided in dribs and drabs by different VCMMs, to accomplish a range of tasks one must install and configure multiple VCMMs and manually convert and reconstruct data and models for each of them. This critically reduces the efficiency of this methodology and potentially hampers the possible utilizations of VCMMs.

## **1.2 Research motivation, aim and objectives**

Urged by the market need and supported by the technology progress, CMM is constantly being improved in terms of reliability, efficiency and usability, all of which can benefit from the applications of VCMMs. This research has recognized the limitations of the existing VCMMs, especially the absence of a comprehensive, integrated system that covers functions related to multiple aspects of the CMM and allows different functions to seamlessly share data and interact with each other.

Against such a background, this research has proposed an Advanced Virtual Coordinate Measuring Machine (AVCMM) system, which organically combines inspection simulation with uncertainty evaluation features, while improves the usability and efficiency of both. User may carry out inspection planning in AVCMM by operating a visualized CMM model, and then have the results of planned tasks including the associated uncertainties quickly calculated and evaluated. The proposed system uses universal model and data sets to support both aspects of functionalities, so that the system components can easily communicate with and provide feedback to each other. AVCMM utilizes a layered design and offers compatibility with various types of CMMs, extendibility of functions and reusability of important components.

The usability of the system is improved by providing 3D graphic representation of the

virtual CMM and supporting operations similar to real CMM. To support inspection planning and point sampling within the AVCMM, an efficient collision detection mechanism is needed for the virtual environment. Therefore a novel collision detection engine that is fast and accurate has been designed and implemented. The AVCMM supports the simulation and combination of errors from different sources. To meet different requirements, AVCMM provides multiple modes for uncertainty evaluation.

While the overall aim of this research is to design, develop and evaluate the proposed AVCMM system that can generally improve the performance of CMM measurements, the distinct objectives include:

- To critically review the current state of CMM related researches with respect to inspection planning and uncertainty evaluation, focusing on the development of VCMMs.
- To establish methodology and to propose a sound architecture for the AVCMM system.
- To design the AVCMM system and its important components, with many novel approaches.
- To implement a prototype of the proposed AVCMM system, using data obtained from experiments and finite element analysis (FEA).
- To validate the implemented AVCMM prototype with experiments.

### **1.3 Thesis structure**

This thesis is composed of the following seven chapters:

Chapter 1: Introduction. This chapter introduces the research background, describes the research aim and outlines the research objectives.

Chapter 2: Literature Review. This chapter mainly investigates literature in CMM related research areas, particularly regarding inspection planning and uncertainty evaluation. The state of the art of VCMM is critically reviewed, followed by a brief review of Virtual Reality Modelling Language (VRML).

Chapter 3: Proposed Advanced Virtual Coordinate Measuring Machine and Methodology. This chapter provides an overview of AVCMM, establishes methodology and sketches out the architecture for the proposed system.

Chapter 4: Design of AVCMM. This chapter details the theories and design of the AVCMM system, focusing on several important components.

Chapter 5: Implementation of a Prototype of AVCMM. This chapter presents the development of an AVCMM prototype.

Chapter 6: Validation, Results and Discussion. This chapter reports and discusses the results of experiments that were designed and conducted to verify and evaluate the implemented AVCMM prototype.

Chapter 7: Conclusions and Future Work. This chapter summarises the research project, points out the contributions to knowledge, and makes recommendations for future work.

## Chapter 2 Literature Review

### 2.1 Introduction

In this chapter, CMM related technologies and researches are reviewed, emphasizing inspection planning and uncertainty evaluation, from where the motivation of this research derives. The GUM uncertainty framework and the Monte Carlo approach of uncertainty evaluation are closely examined and compared. The existing proposals of VCMMs are critically reviewed and categorised, from which the absence of a comprehensive solution is revealed. The VRML technologies, especially the researches on the collision detection for VRML, are also briefly reviewed, as VRML with innovatory collision detection method is used to build the proposed AVCMM.

### 2.2 Coordinate measuring machines

#### 2.2.1 Introduction

A coordinate measuring machine (CMM) is an instrument for measuring geometrical features of an object. A useful definition can be found in ISO10360-1: A CMM is a “measuring system with the means to move a probing system and capability to determine spatial coordinates on a workpiece surface” (International Organization for Standardization, 2000). A typical CMM usually possesses three orthogonal axes X, Y, Z, and a probe that is attached to the third moving axis, as shown in Figure 2.1. The three axes are stacked together, and each axis may move along its direction. The combination of the movements of three axes allows the probe to travel in a three dimensional space. Along with the rotation of the probe head and tilting of the probe stylus, CMM probe has a great degree of access to the object being inspected.

Normally a CMM can be controlled manually by an operator, or it may be driven by a computer. Modern CMMs of CNC type are highly computerized so that they may be automated and have very powerful computational abilities. CMMs can collect positional information from a set of points, and based on that, calculate various geometrical features such as radius of a sphere, and even determine complicated relationships between features, for example, angles between surfaces, distance between the centres of two spheres, and so on.



Figure 2.1 A typical Bridge CMM

Comparing to conventional inspection approaches, CMM inspection has many advantages:

- **Universality.** CMMs are essentially very precise Cartesian robots operating in universally compatible Cartesian coordinate system (Wang et al., 2006). They can theoretically measure anything although in reality there are still limitations (Kurfess, 2006).

- High efficiency. In contrast with conventional geometry inspection, in which the workpiece alignment is difficult and the establishment of appropriate reference points are labour intensive and time-consuming, CMM greatly improved inspection efficiency by introducing a coordinate system and data-fitting algorithms to allow operators to set up workpiece “in a convenient orientation for the inspection and all coordinate data measured is subsequently corrected according to the calibration” of the workpiece and the probe (Gu, 1994).
- High accuracy and precision. In CMM inspection accumulation of the errors resulting from hard-gauging inspection approaches is eliminated. Measurements are taken in temperature controlled environment, by using common procedure guidelines. The operator’s influence can also be kept to a minimum (Gu, 1994).
- Automation. Modern CMMs are normally integrated with computer controlling systems. The automation of CMM inspection has great potential (Gu, 1994) and has become one of the most exciting research areas in CMM metrology. There is a definitive standard and programming language called DMIS (Dimensional Measurement Interface Specification) available for controlling CMM and for communication between a CAD (computer-aided design) system and the CMM.

The first CMM was designed and developed in 1963 by Italian company Digital Electric Automation (DEA) which introduced a portal frame CMM with three axes and a hard probe (Ferranti Metrology from Scotland also claim to be first, although its machine only had 2 axes). Ferranti (now International Metrology Systems or IMS) is believed to have introduced the first Direct Computer Control (DCC) CMM while DEA claimed to have developed the first Computer Numerical Control (CNC) CMM.

The invention of CMM was urged by the need of high precision measurement in the

manufacturing industry at that time. Its emergence was partly promoted by “the ‘technology push’ associated with the rapid development of micro-electronics, computers and precision machining technology, and partly due to the ‘market pull’ of the need for fast, reliable and flexible measurement” (Yang, 1992). Although the CMM industry was born immediately after the invention of CMM, it did not achieve big commercial success until the late 1970s. However, along with the fast development of advanced manufacturing industry, the amount of workpieces with complicated and aesthetic shape is rapidly increasing while higher product quality and greater automation is continuously being pursued. This industrial trend called for better measuring capability that CMMs possess. The reliable inspection and automation became indispensable. Nowadays CMMs are extensively used as one of the most powerful metrological instruments in manufacturing industry.

### **2.2.2 CMM systems and configurations**

Early CMMs were equipped with direct digital display or digital read out (DRO), or a computer for data processing, but can only be operated manually. Now many CMMs are of CNC type and can be driven automatically from the CMM program. Following features became common in modern CMMs: crash protection, offline programming, CAD model import capability, temperature compensation, etc. The appearance and architecture of CMMs also advanced immensely over the past decades. Most designs of CMM are based on a Cartesian coordinate system, i.e. three mutually orthogonal axes, due to the simplicity and universality of such an arrangement (Yang, 1992).



(a) Articulated arm CMM



(b) Cantilever CMM



(c) Fixed bridge CMM



(d) Moving horizontal arm CMM



(e) Gantry CMM

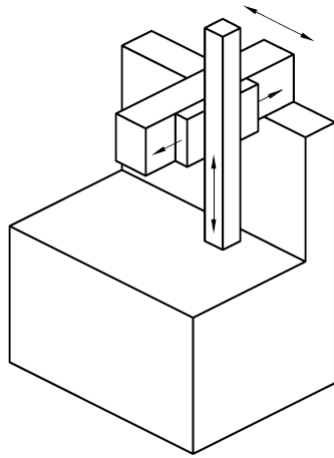


(f) Portal (moving bridge) CMM

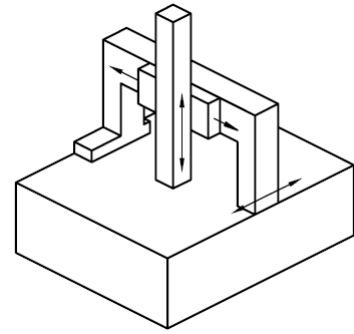
Figure 2.2 Examples of CMM types

Figure 2.2 illustrates some examples of commonly available CMM types. The configurations and types of CMMs officially listed and described in ISO 10360-1 are: (a) Fixed table cantilever CMM, (b) Moving bridge CMM, (c) Gantry CMM, (d) L-shaped bridge CMM, (e) Fixed bridge CMM, (f) Moving table cantilever CMM, (g) Column CMM, (h) Moving ram horizontal-arm CMM, (i) Fixed table horizontal-arm CMM, and (j) Moving table horizontal-arm CMM, as demonstrated in Figure 2.3 (International Organization for Standardization, 2000). This is not an exhaustive list but a sample of commonly used configurations.

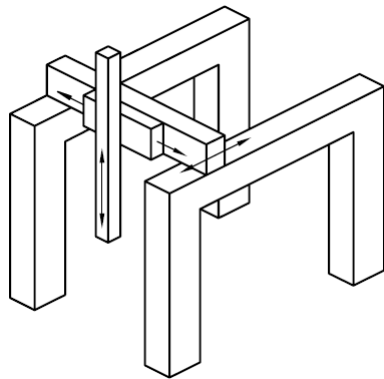




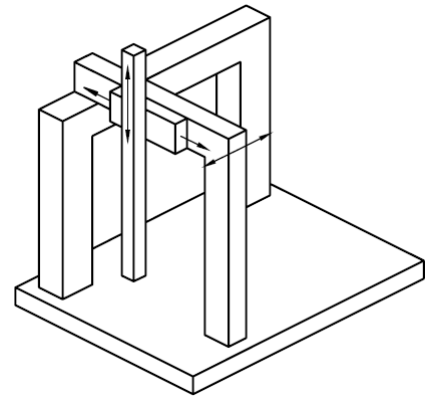
(a) Fixed table cantilever CMM



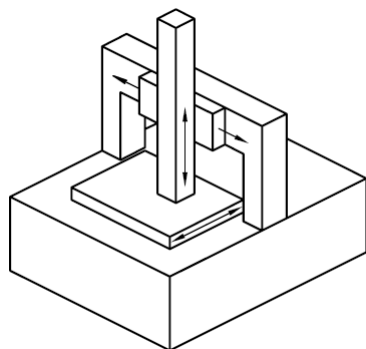
(b) Moving bridge CMM



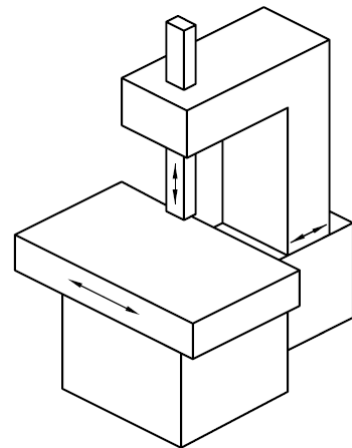
(c) Gantry CMM



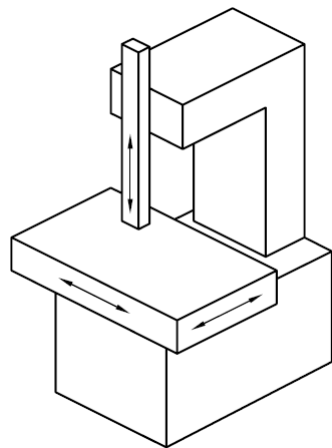
(d) L-shaped bridge CMM



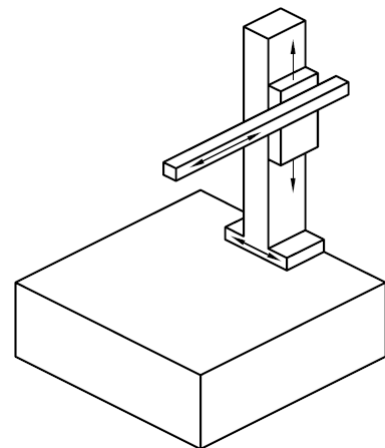
(e) Fixed bridge CMM



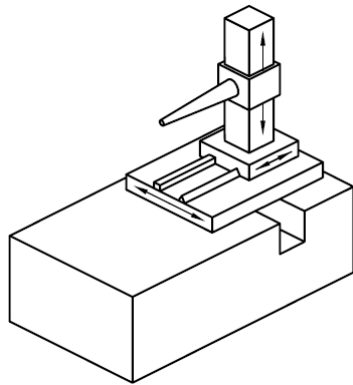
(f) Moving table cantilever CMM



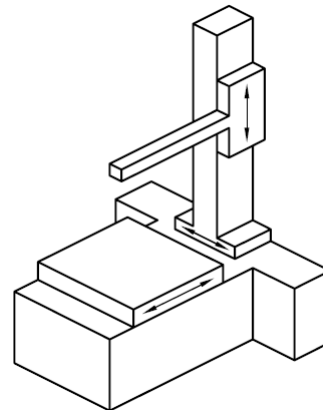
(g) Column CMM



(h) Moving ram horizontal-arm CMM



(i) Fixed table horizontal-arm CMM



(j) Moving table horizontal-arm CMM

Figure 2.3 Ten common CMM configurations

Typical accuracies for some of the configurations are listed in Table 2.1 (Rodger et al., 2007).

Configuration	Indicative Errors
Bridge	$\pm < 5 \mu m$
Gantry	$\pm 10s \mu m$
Column	$\pm 10s \mu m$
Cantilever	$\pm 10 - 50 \mu m$
Articulated Arm	$\pm 50 - 250 \mu m$

Table 2.1 Typical CMM accuracy

### 2.2.3 CMM measuring systems and probing systems

CMMs usually utilize electromechanical, electronic or optoelectronic measuring systems such as linear or rotary encoders, inductive or capacitive transducers and interferometers (Yang, 1992).

In the early days the probes of CMMs were normally attached onto the end of the spindle with a probe holder. These probes usually consisted of a hard ball tip mounted at the end of a stem, although some probes may have tips of other shape, e.g. a quadrant, to measure special features. These probes were physically held by operator and were brought into contact with points on the surfaces of features to be measured by hand. The spatial positions of these points were read from a 3-axis DRO or were stored in a computer. In practise, measurements taken by this method were normally considered to be unreliable due to that these manual operations differed hugely from operator to operator in many ways, for example the force they applied to the probe, the measurement techniques they were used to, etc. Hence the measurement speed and the precision and accuracy of measurements depended heavily on the skill and experience of an operator. This method was also prone to probe damage or workpiece deformation

due to accidental collision or excessive force applied by operators. These problems were partially solved by the inclusion of additional motors that drive each axis. This way the operators no longer need to move the probe physically by hand but instead can control the movement of probe by using joysticks.

One of the most important developments in probing technology so far was the introduction of electronic touch trigger probe invented by David McMurtry, founder of Renishaw PLC, in the 1970s (McMurtry, 1979). This probe has a spring loaded steel ball (replaced by a ruby ball in newer models) stylus and is equipped with a highly precise micro-switch that responds to displacements of the order of  $1\ \mu\text{m}$  or smaller. When probe contacts the surface being measured, the stylus deflects and triggers an electronic signal to record its current coordinates. Touch trigger probe dramatically improved accuracy and precision of CMM, and reduced the difference of measurements taken by individual operators.

Many new types of probing systems have been developed since then. Depending on whether the probe contacts the surface being measured, probes can be categorized as tactile (contact) or non-contact types. Most commonly used probes today are still of contact types. Contact types are normally slower than non-contact types, but much more accurate, suitable for measuring rigid workpieces consisting of primitive geometrical shapes. Apart from touch trigger type, another kind of contact type probes is analogue probe, which generates continuous displacement information of the probe from its free position. Newer models of contact types may feature probes that can drag along the surface being measured and collect information of points at specified intervals. These are also referred to as 'scanning probes'. Scanning probes are usually more accurate and faster than conventional contact types. Non-contact types of probes, also known as 'optical probes', utilize technologies such as CCD (charge-coupled device) systems, machine vision (MV) systems, and 'non-contact scanning' including laser scanning and

white light scanning. Advanced non-contact scanning technologies make it possible to collect large amount of high-density points' information in very short time, normally many thousands of points per second or millions of points per inspection. These 3D 'point-cloud data' can be used to not only check geometrical features but also recreate 3D image of the workpiece. With integration or support of CAD software, it is convenient to create 3D model of the workpiece. This makes non-contact scanning probes ideal for reverse engineering and rapid prototyping, especially for objects with complex shapes or free-form surfaces. And since there is no contact, measurements of soft, deformable or delicate objects become feasible with non-contact probes. Although non-contact types are normally considered to be less accurate than contact types, the gap is closing due to rapid advancing technologies.

Due to the distinct characteristics of tactile and non-contact types of probes, there are emerging research interests in combining different probing technologies in one integrated CMM system to either fit wider range of application (Qin et al., 2008) or improve inspection speed and still retain high precision (Shen et al., 2000).

In terms of size and measurement scope, apart from regular probes, microprobe is another emerging research direction. Microscale probing technologies include scaling down conventional probes, optical probes, standing wave probe (Bauza et al., 2005) etc. Although there are already some commercially available applications of microprobe, it still faces critical challenges including lack of a high aspect ratio probe with ability to access deep, narrow features, meanwhile without contacting surface being measured or maintaining mild contact forces to prevent damaging the high precision surface (nanometre level). Another challenge is that current microprobes are very unreliable and vulnerable to influences of environmental factors such as humidity and surface interactions.

The definitions found in BS6808-1 regarding probes are as following (British Standards Institution, 1987):

- Touch-trigger probe: “A probe that gives a binary signal as a result of contact with a workpiece”;
- Analogue contacting probe: “A probe that gives a signal, or signals, proportional to the displacement of the probe from its free position”;
- Nulling contacting probe: “A probe that, when referenced to a workpiece, gives a signal which causes the coordinate measuring machine to be driven to a position, giving a constant probe reading, usually (but not necessarily), zero or near zero output”;
- Passive (solid) probe: “A probe that mechanically locates the movable components of the coordinate measuring machine relative to the workpiece”;
- Non-contacting probe: “A probe which has no material contact with the surface being measured”.

In ISO 10360-1, probing systems are simply classified into following categories (International Organization for Standardization, 2000):

- Contacting probing system: “probing system which needs material contact with a surface being measured in order to function”;
- Non-contacting probing system: “probing system which needs no material contact with a surface being measured in order to function”;

- Optical probing system: “non-contacting probing system which creates a corrected measured point by probing using an optical system”;
- Multi-probe system: “probing system with more than one probe”;
- Articulating probing system: “probing system which can be oriented in various spatial angular positions by means of a manual or motorized positioning device”.

Figure 2.4 shows the pictures of several types of probes.



(a) Touch-trigger probe

(b) Contact-sensing rigid probe

(c) Analogue scanning probe



(d) Laser scanning probe



(e) Vision probe

Figure 2.4 Different types of probes

## 2.2.4 CMM inspection planning

In today's fast developing manufacturing industry, workpieces that have more complicated geometrical form and aesthetic shape are being produced more and more to meet both functional and aesthetical requirements. The ability to rapidly design and manufacture these products with high accuracy and precision became essential to thrive in this competitive environment (Qin et al., 2008), and measurement with accurate coordinate data is one of the most important procedure, sometimes even the bottleneck in the process to achieve short production cycle. It is required that CMM inspection must be conducted "with more accurate and more efficient operating procedures in order to meet increasingly higher quality assurance standards with shorter cycle times" (Lu et al., 1999). As CMM has become one of the most accepted and widely adopted metrological instruments, it is very highly desired to improve CMM's performance which includes two principal goals, increasing inspection speed and improving measurement accuracy (Lu et al., 1999). The CMM inspection procedure can be divided into three steps which are inspection planning, measuring program generation and actual inspection. The key to achieve better CMM performance is inspection planning.

Inspection planning plays a critical role in CMM inspection procedure. The strategy and methodology used in inspection planning not only directly affect the efficiency of measurement and the accuracy of inspection results, but also are responsible for the safety of equipment and workpiece during the inspection.

In recent years, the study of inspection planning has grown into a continuously developing research area and has attracted more and more attention in research literature (Ziemian & Medeiros, 1998). Typically the CMM inspection planning involves two aspects or levels of planning (Yang & Chen, 2005). Low-level planning includes the selection of measurement points on each surface, accessibility analysis for measurement



points, grouping and sequencing of measurement points and generation of collision-free inspection path. High-level planning focuses on the setups of workpiece on the measurement table, determination of features to measure in certain setup, selection and orientation of the probes (Chen et al., 2004), etc. In each above specific area many researches have been done and new approaches have been proposed.

Since low-volume high-variety production and close tolerance high-quality products have become the characteristics of the modern manufacturing industry (Chen et al., 2004), measurement strategies and programs are forced to change frequently depending on different configurations and requirements of workpieces. In this situation the automation of inspection planning becomes very desirable due to the following reasons:

- Inspection needs to be re-planned every time the workpieces change or measured features and requirements change. This process is very time consuming especially when it is carried out manually on CMMs. It increases the inspection cycle and encumbers the efficiency of entire manufacturing chain. Automated inspection planning greatly reduces human workload and improves inspection speed.
- CMMs are very expensive instruments and traditional manual inspection planning on physical machine causes additional wear and risks of collision damage to CMMs. Automatic and off-line generation of measurement program can avoid manual operation on physical CMM and hence reduces cost and improves safety.
- The effectiveness and validity of inspection relies hugely on the skills and experience of the operators when the inspection planning is conducted manually. And the reliability of inspection differs from one operator to another. By automating the planning process using computer it eliminates the difference caused by individual operators.

- In advanced computer integrated manufacturing (CIM) environment, CMMs become more involved in in-process inspection and are required to be more closely integrated with computer aided design and computer aided manufacturing (CAM). Automation of inspection planning is crucial for efficient and computer managed interactivity between CMM and other components in the CIM environment, making it possible to form a complete automated manufacturing system.

The following sections review recent studies and researches on improving CMM inspection planning in different aspects.

#### **2.2.4.1 Accessibility analysis**

Accessibility analysis is an essential and pivotal step in CMM inspection planning. It determines a bounded space associated with the feature of interest, which defines the range of possible probe orientation (Ziemian & Medeiros, 1997). The goal of accessibility analysis is to determine a set of feasible orientations that probes can be set to inspect specific measurement points without collision or interference with the workpiece. By maximizing the number of features inspected with the same probe orientation, it minimizing unnecessary changes of probe orientation during the inspection process (Wang et al., 2006). This is considered to be the most critical factor that affects the effectiveness of inspection planning, because the change and re-orientation of probes and associated re-calibration and setup are the most expensive operations (Limaiem & ElMaraghy, 2000).

In the context of general CMM inspection planning, accessibility analysis is closely associated with the tasks of workpiece setup and probe selection and acts as the determinant of the decision of these two tasks. Generally there are two approaches for accessibility analysis, relative approach and absolute approach (Limaiem & ElMaraghy, 2000). The orientation of the workpiece is fixed in the relative approach, while in

contrast the absolute approach is independent of the workpiece orientation, which is determined after the generation of optimal sequence of measurement points and inspection path. Using relative approach, two different methods can be chosen from to determine probe orientation. One is to select a probe orientation based on some strategies or operator's experience, and then assess whether the probe orientation is feasible. The other is to generate all possible probe orientations and choose the best one (Wang et al., 2009). In absolute approach, the core concept is to group inspection points that can be accessed using the same probe orientation in the same setup (Limaïem & ElMaraghy, 2000).

Accessibility cones (Spyridi & Requicha, 1990), i.e. sets of directions, can be used to quantitatively characterize accessibility of surface features. When defining the range of probe orientations that are suitable for inspecting the feature of interest, the local accessibility cone (LAC) considers only the characteristics of the desired feature itself, while the global accessibility cone (GAC) can define the range of probes that are able to accomplish collision-free inspection by taking into consideration potential collision with all features of the workpiece. This inspection strategy employs algorithms that are primarily based on the computation of Gaussian images and Minkowski operations, to calculate and cluster LAC and GAC. These direction cone clusters are then processed to determine a minimal set of directions for inspecting a workpiece.

Accessibility cones, as shown in Figure 2.5, can be simplified into an accessibility map (Lim & Menq, 1994). By doing so, it reduces the time cost to find the optimal angle, because the accessibility map contains less information than the accessibility cones. And since there is no attribute of position the accessibility map also simplifies the grouping of points. The computation of accessibility map is based on ray tracing algorithm as it is competent to handle complex and concave surfaces. Although the ray tracing algorithm has the drawback of that it can only be done to a certain resolution,

the CMM probe head only has a rotational resolution.

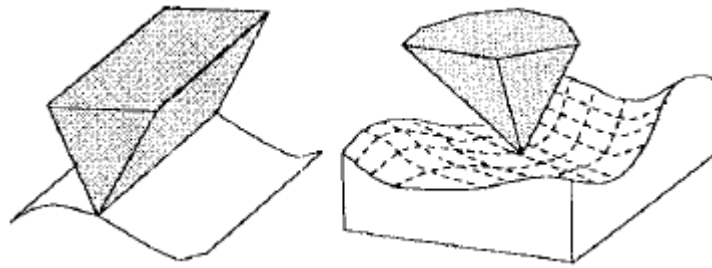


Figure 2.5 Accessibility cone in three dimensions (Lim & Menq, 1994)

Ziemian and Medeiros (1997) proposed an automated method that takes a CAD model of the workpiece and a list of tolerance information as input, and for each feature of interest, the algorithm outputs a set of feasible probe orientations by calculating potential collisions between probe and workpiece. This feature accessibility algorithm addresses the problem by assessing three levels of accessibilities: local feature accessibility, global point accessibility, and global feature accessibility. The result of the algorithm is the percentage of accessible area of each feature of interest. This result is then analysed to generate the optimal workpiece orientation for the inspection of all desired features.

Apart from these traditional approaches which are computational algorithms based on CAD model of workpieces and probes, recently a serial of researches on accessibility analysis using haptic device (Wang et al., 2006) and STL file format has been reported (Wang et al., 2009). In this approach, the accessibility analysis is carried out in a virtual environment called HVCMM (haptic virtual coordinate measuring machine), as pictured in Figure 2.6. HVCMM utilises a PHANToM desktop haptic device as a user interface to the virtual environment (Chen et al., 2004). PHANToM desktop haptic device has six degree of freedom (DOF) of position sensing and three DOF of force feedback. This haptic interface allows users to manipulate the digital probe in the virtual environment with their hands, in the similar way as they do in real world. And with force feedback it

provides user with intuitive experience of feeling the contact and collision between the virtual probe and virtual workpiece (Yang & Chen, 2005). Both the probe units and workpiece to be measured are represented using STL file format, which is widely used for rapid prototyping and CAM. Collision detection algorithm is implemented to detect the intersection between probe units and workpiece. If the intersection is between the workpiece and the probe stylus tip, then the desired measurement point is accessed normally without collision, and a small force is fed back to user to notify the contact. If the intersection occurs between the workpiece and anywhere else on the probe, then a collision is detected and a large and constant force is fed back to user to alert the collision. By simulating the operation and collision of physical CMMs, this approach allows users to carry out manual accessibility analysis off-line, without the need of operating a real CMM. In their work (Wang et al., 2009), the workpiece orientation is considered to be fixed, so the relative approach for accessibility analysis is adopted. They select probe orientation based on human planner's experience and then verify whether the probe orientation is feasible by operating the HVCMM to touch each selected measurement points and detect potential collisions.

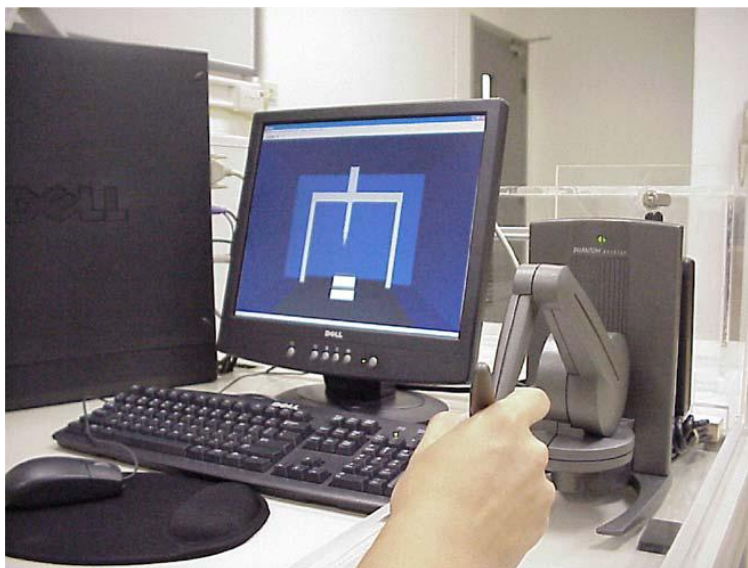


Figure 2.6 The HVCMM system configuration (Chen et al., 2004)

#### **2.2.4.2 Workpiece and probe setup**

As the reason for accessibility analysis, in the process of inspection planning, both workpiece and the probe need to be well setup before inspection. Good setup can reduce the measurement error as well as improve inspection efficiency. The setup tasks include determining the workpiece orientation, planning the fixturing, selecting the suitable probe and choosing the optimal probe orientation. Motivated by the cost and time associated with these tasks, most recent researches mainly focused on the improvement of efficiency and automation.

View planning algorithm is implemented to determine the best probe orientation that enables “an improved tactile measurement in terms of measurement error and inspection time while maximizing sampling point measurements per probe position” (Gerhardt & Hyun, 1995). Using this algorithm, the measurement error is modelled mathematically as a function of the probe approaching direction and the surface normal. The best probe orientation is closely associated with the measurement points distribution, i.e. different measurement points distributions yield different corresponding optimal probe orientation. The results of their experiments reveal that the generated best probe orientation produces the minimal standard deviation of measurement error globally as well as a reduced measurement error. It also reduces the distance of probe movement hence improves the inspection speed.

Cheng and Cai (1995) presented an approach to automate the process of fixturing planning for CMM inspection in the context of automatic manufacturing system environment. A modular, configurable fixture system was established which is “suitable for not only measuring process but also automated fixturing planning” (Cheng & Cai, 1995). Figure 2.7 demonstrates the functional components of the developed fixture system. They analysed the difficulties in fixturing planning and proposed a planning

strategy that makes use of the shape body based part description method, the shape body elements relating technology and the characteristics of the measurement fixturing to process complicated workpiece structure and reduce the requirement of strength, clamping force and accuracy etc. The fixture system carries out fixturing planning in two steps. First, it determines the orientation of workpiece according to the accessibility of features to be inspected, and decides the style of fixing based on the fixing stability of the bottom side surface. It then generates detailed fixture structure parameters and actual workpiece position. The fixturing planning system has been integrated in THAIP&P (Tsinghua Automated Inspection Planning and Programming system) and could determine appropriate workpiece orientation and position, and conduct planning of fixing points and fixture parameters. This application shows that the fixturing planning system is suitable for CMM inspection in an automatic manufacturing system environment and is practical for inspection of complicated non-rotational workpieces.

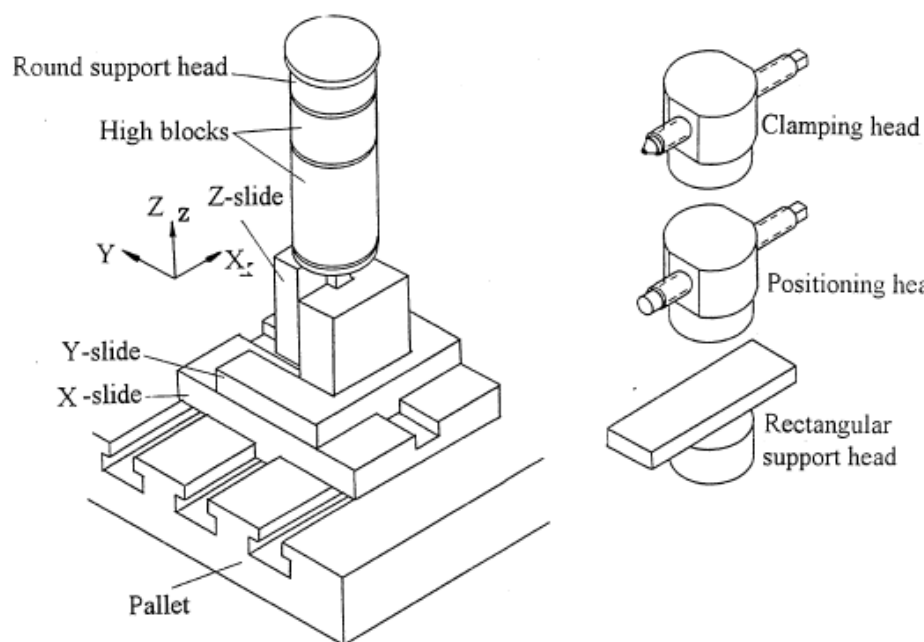


Figure 2.7 Functional combination of configurable fixture (Cheng & Cai, 1995)

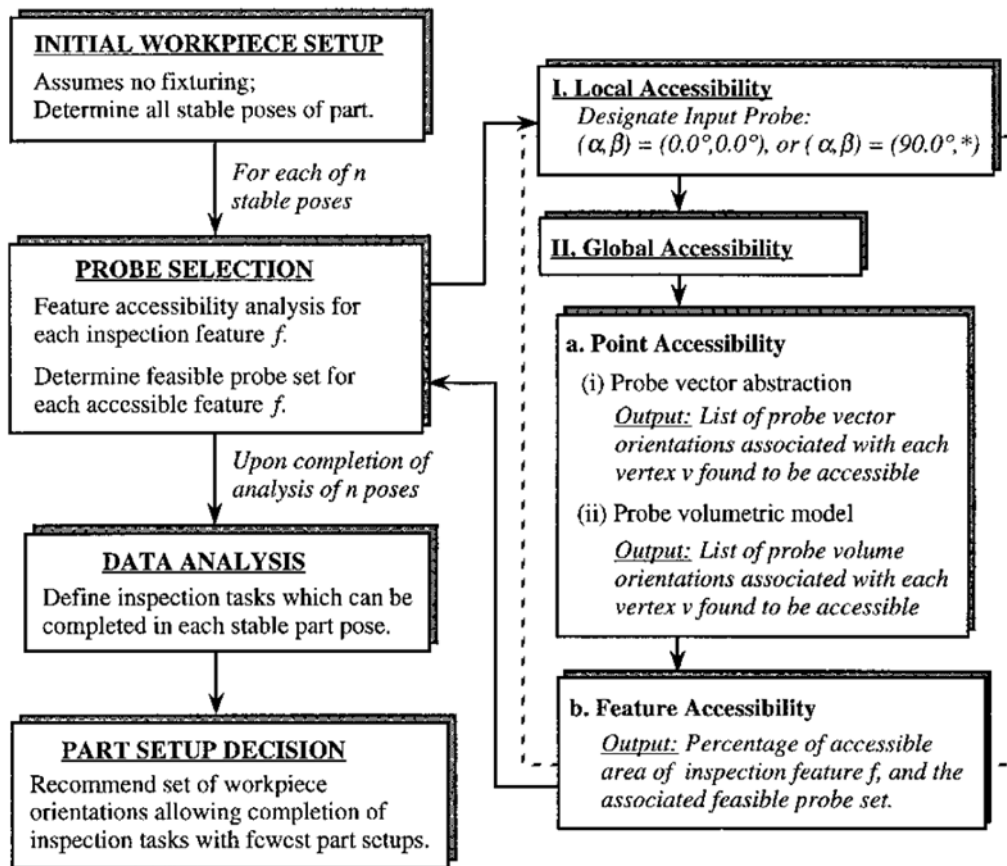


Figure 2.8 Logic of the probe and workpiece setup planner utilizing feature accessibility algorithm (Ziemian & Medeiros, 1998)

Aiming to minimize the number of workpiece setups required for a full inspection and to verify defined geometrical and dimensional constraints, Ziemian and Medeiros described a probe and workpiece setup planner utilizing their feature accessibility algorithm (Ziemian & Medeiros, 1997; 1998). This planner, as sketched in Figure 2.8, takes the CAD description of the workpiece, as well as its associated geometrical dimensions and tolerances as input, and automatically generates probe and workpiece setup recommendations by determining the stable workpiece orientation according to its CAD model, assessing feature accessibility to determine measurable features and



successively determine the feasible probe orientations, analysing accessibility data together with input data indicating which tolerances can be fully verified in each stable orientation, and finally recommending the workpiece orientation sets that require the minimal number of workpiece setups needed for inspection of all desired features. Although the outputted probe orientations cannot be guaranteed to be optimal, each result is a potential option to conduct a collision-free inspection for certain portion of the desired feature. This planner simplifies the CMM inspection process by reducing human decision on workpiece and probe orientations, and it shows the potential of becoming an integral component for a completely automated system.

#### **2.2.4.3 Path planning**

Path planning, or the generation of optimal collision-free inspection programs, is the most important purpose and output of CMM inspection planning. Its tasks include the selection, clustering and sequencing of measurement points, and the generation of collision-free inspection paths.

Traditionally, the CMM's inspection paths are programmed manually by operating the probe of a physical CMM, to travel through a sequence of movements which are recorded in computer system and are then repeated automatically in the subsequent inspections. This method is normally referred to as 'learning mode' or 'teaching method', similar to teaching a robot in robot assembly (Gu, 1994). However the 'teaching method' has many drawbacks, such as:

- As operated by human planner, the CMM probe moves, almost unavoidably, through some repeated routes and travels some unnecessary and non-optimal path. This results in a slower measurement speed for each repeated inspection hence sometimes can considerably increase the overall inspection cycle time in batch production.

- Depending on the skills and experiences of operators, the efficiency of the inspection program, as well as the validity of the resulting inspections, differ largely between individuals.
- Due to human nature, misoperations and accidents cannot always be avoided while manually planning the inspection path. Potentially the resulting collision may seriously damage the expensive CMM and the workpiece. This risk is particularly high with a new operator.

Many new approaches have been proposed and studied in the past few years, in the effort to overcome above limitations and to further improve the efficiency.

Genetic algorithm (GA) inspection path planning optimiser (Lu et al., 1999) was developed to help to generate an optimal collision-free inspection path, with no repeated routes, in the inspection of multiple components. Lu et al. compared genetic algorithms with linear integer programming and in order to overcome many of the disadvantages of the latter they chose GA to implement the planning optimiser. Genetic algorithms are a class of evolutionary algorithms that are based on techniques and concepts derived from evolutionary biology such as inheritance, crossover, mutation and selection. Figure 2.9 shows the structure of a simple genetic algorithm (Lu et al., 1999). Normally used in machine learning and solving search and optimization problems, in the planning optimiser GA is used to model the path planning of multi-component inspection. A tree structure is used to represent the workpiece. The deeper a level is on the tree, the more detailed and specific information it contains. At the same time a relationship matrix is created to handle the relationship between measurement points and ‘dummy points’ which are extra points added to the inspection path to avoid collision. Each element  $(i, j)$  in the relationship matrix contains a Boolean value indicating whether a route exists from point  $i$  to point  $j$ . Ordinal encoding is chosen to be the encoding scheme

for the GA optimiser. The results shows reasonable reduction of path length compared to teaching method.

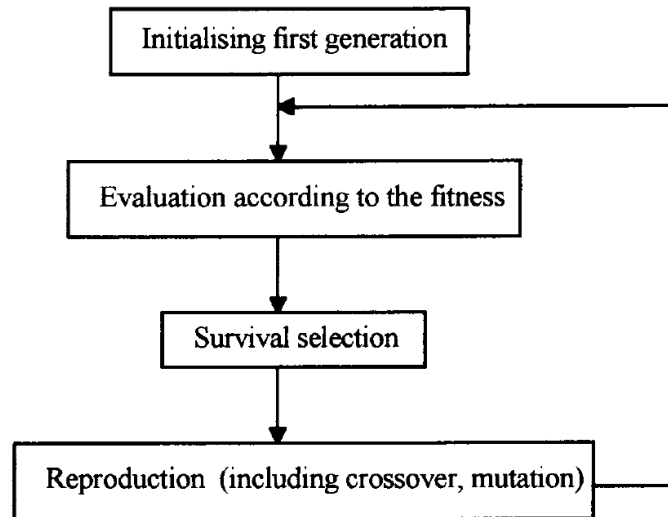


Figure 2.9 Structure of a simple genetic algorithm

Albuquerque et al. (2000) presented algorithms for automatic inspection point placement and path planning that may be used in the integration of CAD and CMM inspection. These algorithms make use of iterative subdivision of surface to determine inspection point placement, together with 3D collision avoidance, to generate optimal collision-free inspection path for geometrically complicated workpieces that have multiple intersecting features. These algorithms have been utilized as part of an overall automated CMM inspection planner, which is used to improve the inspection capabilities of the quick turnaround cell (QTC) rapid design and manufacturing system (Chang, 1990).

Jiang and Chiu (2002) developed a feature-based technique to determine the optimal number of measurement points needed for the inspection of a specific feature, according to form tolerance. They developed a computer aided inspection (CAI) system to calculate the required number of measurement points. The concept behind this CAI

system is that the determined number of measurement points should keep a balance between minimal cost (minimal points to be measured), and acceptable measurement uncertainty. Since for online measurement the errors caused by measurement are usually not separated from the workpiece dimension deviation from the norm, they chose form tolerance to be the acceptable error amount as “it best represents the limit of the sum of all possible error sources” (Jiang & Chiu, 2002). The inspections of circular and cylindrical features are demonstrated, showing that the proposed method produces considerably lower measurement error as compared to traditional method i.e. determining the measurement points by human inspectors.

Based on the developed HVCMM, path planning can be carried out manually in the virtual environment in the way similar to operating a physical CMM (Chen et al., 2004; 2005; Yang & Chen, 2005). Although this is not an automatic method for inspection planning, HVCMM provides an interactive environment with intuitive operations and force feedback to make the process simpler and less risky for human operators.

Raghunandan and Venkateswara Rao (2008) investigated the relationship between the surface roughness of workpiece and the sampling strategy i.e. the determination of the number and locations of measurement points, for best determination of flatness error when using CMM for inspection. This research used Hammersley sequence sampling to identify the location of measurement point and minimum zone method (MZM) fitting to evaluate the flatness error. Figure 2.10 compares the variation of flatness error with respect to the sample sizes for three specimens with different roughness (Raghunandan & Venkateswara Rao, 2008). Results show that the surface quality has influence to the accuracy of inspection and hence can be used as a criterion to determine the initial sample size when evaluating flatness error.

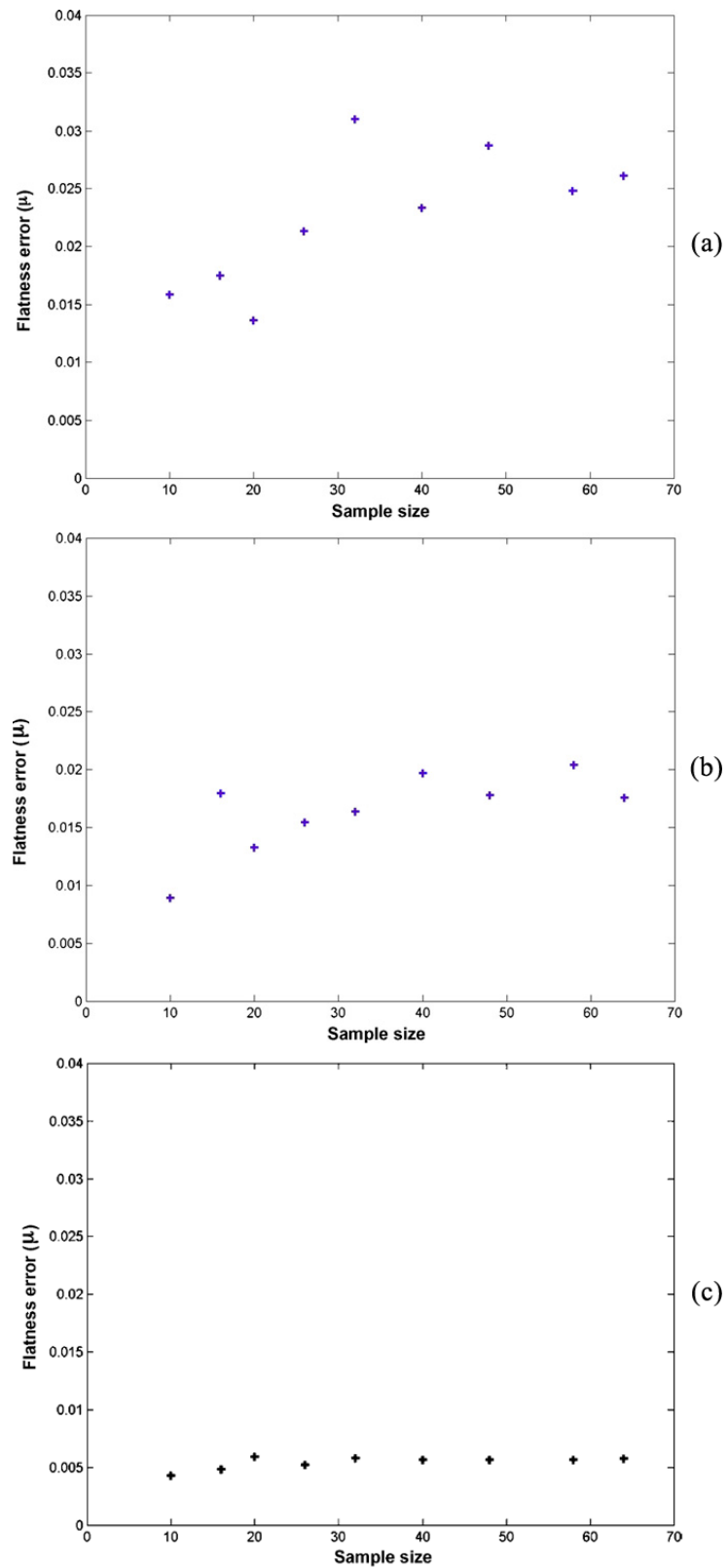


Figure 2.10 Variation of flatness error with respect to sample size for: (a) Ra 9.0  $\mu\text{m}$ ; (b) Ra 3.5  $\mu\text{m}$ ; (c) Ra 0.2  $\mu\text{m}$

#### 2.2.4.4 Automation and intelligent planning system

Automation has been one of the most desired objectives of many of the above mentioned researches. Some other works have been conducted focusing on the development of automation.

Furutani et al. (1994) briefly described an autonomous inspection planning system developed for CMM. This system takes various kinds of information as input data, such as form and dimension data of the workpieces, tolerances, parameters of the probing system, alignment accuracy, etc. It was reported that this system can automatically generate the inspection path based on sufficient input, such as shown in Figure 2.11.

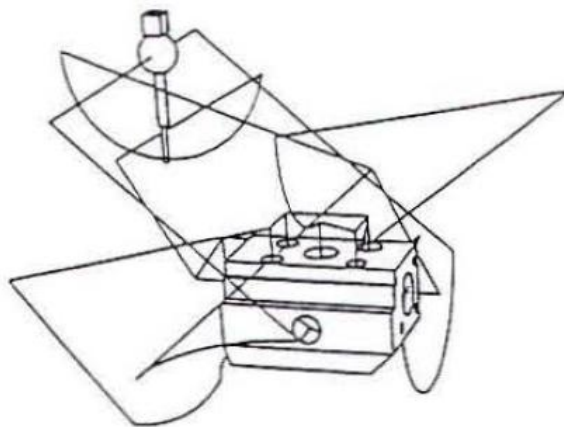


Figure 2.11 Probe path generated for inspecting a workpiece in an experiment (Furutani et al., 1994)

Lin et al. developed a generic algorithm for CAD-directed CMM inspection planning, based on the modified ray tracing technique which uses B-Rep (boundary representation) data from any geometric modelling systems (Lin & Mahabaleshwarkar, 1999; Lin & Murugappan, 2000; Lin et al., 2001). The concept is to establish an imaginary ray between start point and target point, and if intersection with workpiece is encountered then an optimal detour path must be created to avoid collision between probe and

workpiece. The flow chart of this algorithm is shown in Figure 2.12. This algorithm was implemented to demonstrate its ability to determine optimal collision-free inspection path for prismatic workpiece models. The algorithm was designed to serve as a principal component of an automated inspection path planning system as shown in Figure 2.13.

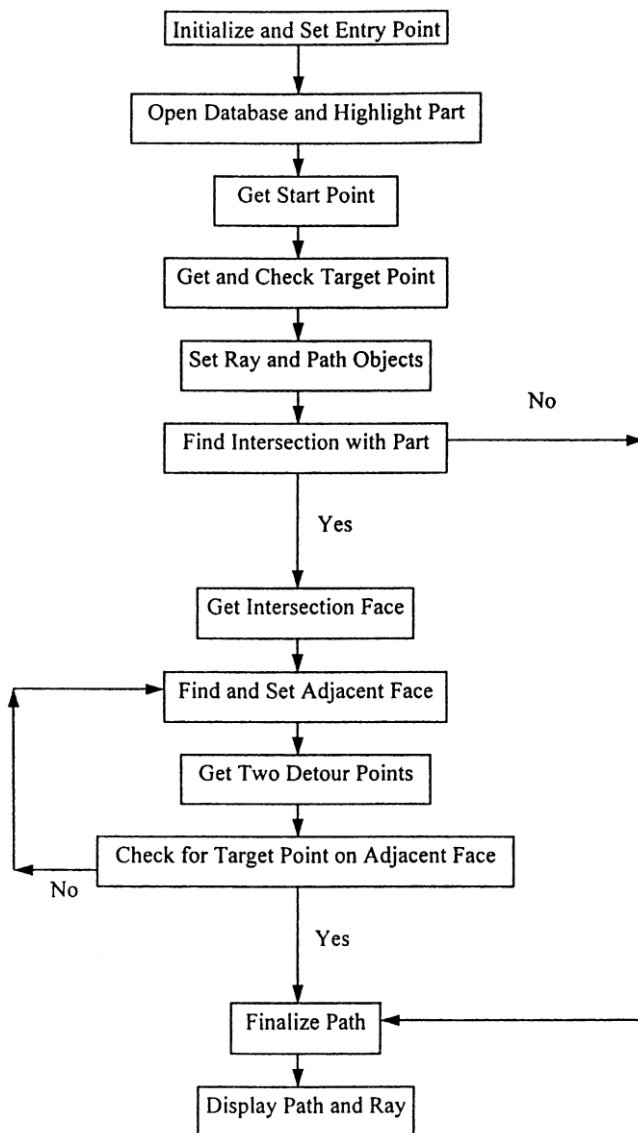


Figure 2.12 Flow chart of the generic algorithm (Lin et al., 2001)

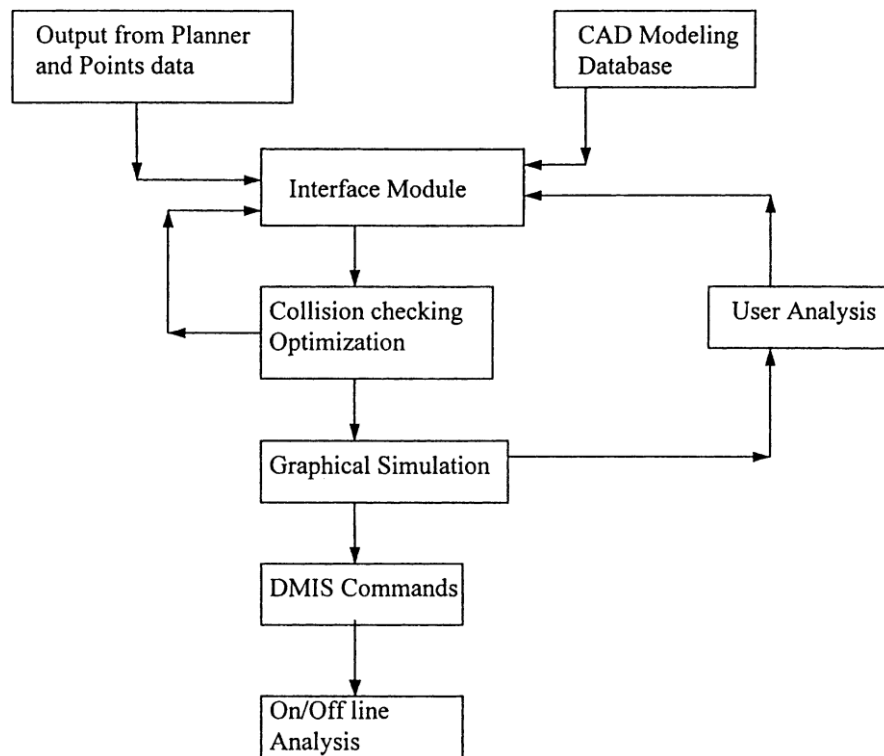


Figure 2.13 Conceptual CMM probe path generation system (Lin et al., 2001)

Qin et al. (2008) presented an automatic inspection planning method utilizing computer vision. They established a vision guided CMM by attaching a camera to CMM probe, as shown in Figure 2.14. When the camera moves along with the probe, it captures images of the workpiece from different positions, hence constructs a stereo vision system. These images are then processed by using symmetric multi-baseline matching to acquire 3D edge characters, based on which the inspection planning is then conducted.

Towards the goal of automatic inspection planning, some proposed works intended to enhance the inspection planning process with simulation of human intelligence, i.e. artificial intelligence (AI) technologies.





Figure 2.14 Vision guided CMM (Qin et al., 2008)

Gu (1994) developed a prototype of a knowledge-based planning system. This prototype has a modular design and consists of eight modules that correspond to components of the model representing CMM inspection planning process. Each functional module handles different sub-problems or tasks in the planning system, including: interfacing with CAD models, conducting non-CMM inspection planning for features not suitable for CMM inspection, assessing feature accessibility, identifying datum feature, selecting probe, grouping features according to accessibility and probe selection, checking datum feature tolerances and then grouping features based on the tolerance type and datum, etc. These modules are knowledge-based sub-systems that make use of a feature base and are coordinated by a main module. Both input design description and planning knowledge are represented in the planning system. ‘The knowledge base contains detailed planning procedures and factual knowledge’ (Gu, 1994) that cover each of the planning steps/tasks mentioned above. The implemented prototype ignored many of the

planning elements but demonstrated the feasibility of some AI planning concepts and showed its potential towards automated inspection planning.

Among many efforts to develop and utilize intelligent technique in CMM inspection planning, some researches include: expert system for inspection planning - EXINS (Expert Inspection System) (Majstorovic et al., 1995), relative information model for LINDO software (Lu et al., 1995), artificial intelligence path planning system for multi-components inspection on CMM (Lu et al., 1995), etc.

### **2.2.5 Evaluation of measurement uncertainty**

To assess the quality of a result of a measurement, the concept of error and error analysis have long been used in the practice of metrology historically. However the application of error involves two idealized concepts – the ‘true value’ of the measurand and the ‘error’ of a measurement, both impractical to be known exactly. Even when all the known or suspected components of error have been well evaluated and properly corrected, there still remains an uncertainty about the correctness of the result (International Organization for Standardization, 2008). Since early 1970s, more and more metrologists realized that it is more appropriate to replace ‘error’ with ‘uncertainty’, while characterizing the reliability of measurement results. Instead of trying to describe the unknowable ‘true value’ and ‘error’, ‘uncertainty of measurement’ focuses on the dispersion of the values that could reasonably be attributed to the measurand.

In 1978, having realized the lack of consensus on the estimation and expression of measurement uncertainty, the CIPM (Comité International des Poids et Mesures) commissioned the BIPM (Bureau International des Poids et Mesures) to produce a widely acceptable recommendation. The BIPM convened the Working Group on the

Statement of Uncertainties that later issued the general Recommendation INC-1 in 1980. Based on the Recommendation INC-1 and Recommendation 1 (CI-1981) of the CIPM, in 1993 the International Organization for Standardization (ISO) published the Guide to the Expression of Uncertainty in Measurement (GUM) (International Organization for Standardization, 1993, corrected and reprinted in 1995), after more than ten years of development and transformation. The GUM was corrected and reprinted in 1995 and then revised in 2008, under the title ‘Uncertainty of measurement – Part 3: Guide to the expression of uncertainty in measurement (GUM:1995)’. This 2008 version is the current version of GUM and is a currently active ISO guide (ISO/IEC Guide 98-3:2008). The development and preparation of the GUM are credited to the Joint Committee for Guides in Metrology (JCGM) consisting of experts nominated by the BIPM, the IEC (International Electrotechnical Commission), the ISO and the OIML (International Organization of Legal Metrology). Other organizations involved in the JCGM are: IFCC (International Federation for Clinical Chemistry and Laboratory Medicine, formerly International Federation of Clinical Chemistry), IUPAC (International Organization for Pure and Applied Chemistry, formerly International Union of Pure and Applied Chemistry), IUPAP (International Organization for Pure and Applied Physics, formerly International Union of Pure and Applied Physics) and ILAC (International Laboratory Accreditation Cooperation, joined other seven founding organizations in 2005).

The following sections introduce the GUM uncertainty framework and the Monte Carlo method described in GUM supplement, and review a range of recent researches related to the uncertainty evaluation of CMM measurement.

#### **2.2.5.1 GUM uncertainty framework**

The formal definition of the term ‘uncertainty of measurement’ as given in the GUM (International Organization for Standardization, 2008) and the International vocabulary of basic and general terms in metrology (VIM) (International Organization for

Standardization, 1993) is: “parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand”. GUM takes great care to distinguish between ‘error’ and ‘uncertainty’ and indicates that the uncertainty of measurement is on account of the lack of exact knowledge of the value of the measurand. Even after correction for recognized systematic effects, a result of measurement is still an estimate of the value of the measurand, due to the uncertainty resulting from random effects and from imperfection of the correction for systematic effects. GUM lists some of the possible sources of uncertainty in a measurement as following:

- Incomplete definition of the measurand;
- Imperfect realization of the definition of the measurand;
- Nonrepresentative sampling — the sample measured may not represent the defined measurand;
- Inadequate knowledge of the effects of environmental conditions on the measurement or imperfect measurement of environmental conditions;
- Personal bias in reading analogue instruments;
- Finite instrument resolution or discrimination threshold;
- Inexact values of measurement standards and reference materials;
- Inexact values of constants and other parameters obtained from external sources and used in the data-reduction algorithm;

- Approximations and assumptions incorporated in the measurement method and procedure;
- Variations in repeated observations of the measurand under apparently identical conditions.

Based on the methods of evaluation from which the uncertainty components are obtained, uncertainty components can be grouped into two categories, namely Type A standard uncertainty and Type B standard uncertainty, corresponding to Type A evaluation and Type B evaluation respectively. These two types are only used to classify two different approaches of evaluating uncertainty components, not that there is any difference in the nature of the components obtained from the two types of evaluation. A Type A standard uncertainty is obtained statistically from a probability density function (PDF) derived from an observed frequency distribution, while a Type B standard uncertainty is evaluated from an assumed PDF. Both types of evaluation are based on probability distribution and both types of uncertainty components are quantified by variances or standard deviations.

To evaluate and express the uncertainty of the result of a measurement in the GUM Uncertainty Framework (GUF), the steps below should be followed.

Step 1. In most cases a measurand  $Y$  is not measured directly but is determined from  $N$  other quantities  $X_i$ . So the first step is to model the measurement and express mathematically the relationship between the measurand  $Y$  and the input quantities  $X_i$  through a functional relationship  $f$ :

$$Y = f(X_1, X_2, \dots, X_N) \quad (2.1)$$

The function  $f$  should contain all the quantities that may contribute significant components of uncertainty to the result of the measurement, including all corrections and correction factors.

Step 2. Determine the estimated value  $x_i$  of input quantity  $X_i$ . If  $x_i$  is evaluated by means of  $n$  independent repeated observations  $X_{i,k}$ , then the best available estimate is the arithmetic mean or average of the  $n$  observations:

$$x_i = \bar{X}_i = \frac{1}{n} \sum_{k=1}^n X_{i,k} \quad (2.2)$$

Sometimes certain input quantities may have their values and uncertainties brought into the measurement from external sources, such as quantities associated with certified reference materials, calibrated measurement standards, and reference data obtained from handbooks, etc.

Step 3. Evaluate the standard uncertainty  $u(x_i)$  of the estimate  $x_i$  of each input quantity  $X_i$ . If an input estimate  $x_i$  is determined by means of statistical analysis of series of observations, its standard uncertainty  $u(x_i)$  can be obtained using Type A evaluation. The Type A variance  $u^2(x_i)$  associated with  $x_i$  can be calculated from:

$$u^2(x_i) = s^2(\bar{X}_i) = \frac{s^2(X_i)}{n} = \frac{1}{n(n-1)} \sum_{k=1}^n (X_{i,k} - \bar{X}_i)^2 \quad (2.3)$$

where  $s^2(X_i)$  is the experimental variance of the observations and  $s^2(\bar{X}_i)$  is the experimental variance of the mean. Their positive square roots are termed the experimental standard deviation  $s(X_i)$  and the experimental standard deviation of the

mean  $s(\bar{X}_i)$  respectively. The Type A standard uncertainty  $u(x_i) = s(\bar{X}_i)$  is the positive square root of the Type A variance  $u^2(x_i)$ .

If an estimate  $x_i$  of an input quantity  $X_i$  has not been determined by statistical analysis of repeated observations, then its associated Type B variance  $u^2(x_i)$  and Type B standard uncertainty  $u(x_i)$  may be obtained by scientific judgement based on all available knowledge and information on the possible variability of  $X_i$ . For example, if an estimate  $x_i$  can be taken from a previous measurement, a calibration certificate, a handbook or a manufacturer's specification, and its quoted uncertainty is expressed as a multiple of a standard deviation, or as a confidence interval, then its standard uncertainty  $u(x_i)$  may be obtained by simply dividing the quoted value with the multiplier in the first case, or it may be calculated based on the confidence level and the type of distribution of the input quantity in the second case.

Step 4. If some of the input quantities  $X_i$  are significantly correlated, the correlations should be taken into account when evaluating the combined uncertainty. When two input quantities  $X_i$  and  $X_j$  are estimated by means of their arithmetic means determined from  $n$  independent pairs of repeated simultaneous observations, their associated covariance  $u(x_i, x_j)$  can be calculated from:

$$u(x_i, x_j) = s(\bar{X}_i, \bar{X}_j) = \frac{1}{n(n-1)} \sum_{k=1}^n (X_{i,k} - \bar{X}_i)(X_{j,k} - \bar{X}_j) \quad (2.4)$$

The estimated covariances  $u(x_i, x_j)$  may be expressed as the elements of a covariance matrix with elements  $u_{ij}$ . The diagonal elements  $u_{ii}$  are in fact the variances  $u^2(x_i)$  and other elements  $u_{ij}$  are the covariances  $u(x_i, x_j) = u(x_j, x_i)$ , where  $i \neq j$ . If two input quantities are uncorrelated, their corresponding elements  $u_{ij}$  and  $u_{ji}$  in the

covariance matrix are expected to be near 0.

Step 5. Calculate the estimate  $y$  of the measurand  $Y$ , which is the result of the measurement.  $y$  can be determined from the functional relationship  $f$  modelled in step 1, using estimates  $x_i$  for the input quantities  $X_i$ :

$$y = f(x_1, x_2, \dots, x_n) \quad (2.5)$$

Step 6. Calculate the combined standard uncertainty  $u_c(y)$  of the measurement result  $y$  following the law of propagation of uncertainty. If all the input quantities  $X_i$  are independent or uncorrelated, the combined variance  $u_c^2(y)$ , which is the square of combined standard uncertainty  $u_c(y)$ , is given by:

$$u_c^2(y) = \sum_{i=1}^N \left( \frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) \quad (2.6)$$

where  $f$  is the functional relationship modelled in Step 1 and  $u(x_i)$  is the standard uncertainty of the estimate  $x_i$  of each input quantity  $X_i$  evaluated in step 3. The partial derivatives  $\frac{\partial f}{\partial x_i}$ , often called sensitivity coefficients  $c_i$ , are equal to  $\frac{\partial f}{\partial X_i}$  evaluated at  $X_i = x_i$ . The sensitivity coefficients  $c_i$  can sometimes be obtained experimentally rather than being calculated from  $f$ .

If some or all of the input quantities  $X_i$  are correlated, the combined variance  $u_c^2(y)$  associated with the measurement result  $y$  is given by:



$$\begin{aligned} u_c^2(y) &= \sum_{i=1}^N \sum_{j=1}^N \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j) \\ &= \sum_{i=1}^N \left( \frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j) \end{aligned} \quad (2.7)$$

where  $u(x_i, x_j) = u(x_j, x_i)$  is the estimated covariance associated with the input estimates  $x_i$  and  $x_j$ , obtained in Step 4.

If a measurement produces multiple results  $y_i$  for two or more measurands  $Y_i$  simultaneously, then in addition to  $y_i$  and  $u_c(y_i)$ , the covariance  $u(y_i, y_j)$  associated with outputs  $y_i$  and  $y_j$  should be given, as well as the correlation coefficient  $r(y_i, y_j)$ , which is given by:

$$r(y_i, y_j) = r(y_j, y_i) = \frac{s(y_i, y_j)}{s(y_i)s(y_j)} \quad (2.8)$$

Step 7. Although it is recommended to use combined standard uncertainty  $u_c(y)$  universally to express the uncertainty of a measurement result, in some cases it's often required to provide a measure of uncertainty as an interval  $y - U$  to  $y + U$  that may be expected to encompass a large fraction of the distribution of values that could reasonably be attributed to the measurand  $Y$ .  $U$  is termed the expanded uncertainty and is calculated by multiplying the combined standard uncertainty  $u_c(y)$  by a coverage factor  $k$ :

$$U = ku_c(y) \quad (2.9)$$

The coverage factor  $k$  is chosen according to the required level of confidence  $p$  of the interval  $y - U$  to  $y + U$  and is usually in the range of 2 to 3. For example, if the probability distribution characterized by  $y$  and  $u_c(y)$  is approximately normal and the effective degrees of freedom of  $u_c(y)$  is of significant size (International Organization for Standardization, 2008), then  $k = 2$  yields an interval with a level of confidence  $p \approx 95\%$ , and  $k = 3$  yields an interval with a level of confidence  $p \approx 99\%$ .

Step 8. Report the measurement result  $y$  together with its combined standard uncertainty  $u_c(y)$  and/or expanded uncertainty  $U$ . If more than one measurands are simultaneously determined then their covariances  $u(y_i, y_j)$  and/or correlation coefficient  $r(y_i, y_j)$  should be reported. Report should be in a clear manner and follow one of the recommended formats given in GUM. Additional information on how the result and uncertainty were determined should also be included in the report, to the extent that one can independently repeat the calculation or update the result in the future when new information become available.

### **2.2.5.2 Supplement 1 to the GUM: A Monte Carlo method approach**

Monte Carlo methods are a class of stochastic techniques that heavily rely on repeated random sampling. The name ‘Monte Carlo’, which is a reference to the Monte Carlo Casino in Monaco, was chosen by physicists at Los Alamos Scientific Laboratory in 1940s as a code name for their work in nuclear weapon projects. Some similar ideas and early variants of Monte Carlo method (MCM) may be traced back to nineteenth century, however, the development and usefulness of MCM were highly restricted due to the

lack of computational power in the pre-electronic computing era. After the invention of electronic computer, MCM started to be studied in depth and quickly became popularized in the areas of physical sciences and operational research. Being studied and developed for more than half a century, MCM has now widely spread its applications to many other fields such as mathematics, engineering, design, meteorology, medical science, finance, business, etc. Although the ways and forms in which the Monte Carlo methods are implemented and applied may vary enormously, Monte Carlo methods share a common principle of using randomly generated inputs and probability statistics to investigate problems.

In the Supplement 1 to GUM (International Organization for Standardization, 2008), a Monte Carlo method approach for uncertainty evaluation is provided as a practical alternative to the GUM. This approach is consistent with the general principles of the GUM, but can be used in many situations where the GUM uncertainty framework is difficult or infeasible to apply, such as those in which: the models are very complicated; the partial derivatives of the model are difficult to obtain; the input quantities have asymmetric PDFs; the PDF for the output quantity is not a normal distribution or a scaled and shifted  $t$ -distribution; the uncertainty components are not of approximately the same magnitude; the measurand estimate and its standard uncertainty are of approximately the same magnitude (International Organization for Standardization, 2008).

To implement uncertainty evaluation using MCM as described in Supplement 1 to GUM, following steps are required for the propagation of distributions.

Step 1. Choose the number  $M$  of Monte Carlo trials to be repeated.  $M$  can be pre-assigned, or can be decided using an adaptive procedure in which the number of Monte Carlo trials is increasing until the outputs have stabilized in a statistical sense.

Step 2. Assign input quantities  $X_i$  appropriate PDFs. Sample from these PDFs  $M$  times to obtain  $M$  vectors, each of which is a realization of the set of input quantities.

Step 3. Using the  $M$  vectors generated in Step 2 for input, evaluate  $Y = f(X_1, X_2, \dots, X_N)$ , which is the functional relationship between input quantities  $X_i$  and output quantity  $Y$ , to obtain  $M$  model values  $y_r$  of  $Y$ .

Step 4. Sort the  $M$  model values  $y_r$  of  $Y$  into strictly increasing order  $y_{(r)}$  to provide an approximate numerical representation  $G$  of the distribution function for  $Y$ .

Step 5. Use  $G$  to calculate the estimate  $y$  of measurand  $Y$  and its standard uncertainty  $u(y)$ . The equations below are used:

$$y = \tilde{y} = \frac{1}{M} \sum_{r=1}^M y_r \quad (2.10)$$

and

$$u(y) = u(\tilde{y}) = \sqrt{u^2(\tilde{y})} = \sqrt{\frac{1}{M-1} \sum_{r=1}^M (y_r - \tilde{y})^2} \quad (2.11)$$

Step 6. For a stipulated coverage probability  $p$ , use  $G$  to determine the coverage interval for  $Y$ . Let  $q$  be the integer part of  $pM + 1/2$ . Then a  $100p\%$  coverage interval for  $Y$  can be expressed as  $[y_{low}, y_{high}]$ , where  $y_{low} = y_{(r)}$  and  $y_{high} = y_{(r+q)}$  for any  $r = 1, \dots, M - q$ . Taking the integer part of  $(M - q + 1)/2$  for  $r$  yields the probabilistically symmetric  $100p\%$  coverage interval.

Comparing to GUM uncertainty framework, some differences and advantages of MCM approach are:

- PDFs are assigned to input quantities  $X_i$  instead of their estimates  $x_i$  and associated standard uncertainties  $u(x_i)$ .
- The sensitivity coefficients are not required. Therefore the calculation or experimental acquisition of the partial derivatives of model  $f$  with respect to the  $X_i$  are not needed. For non-linear or complicated models, it is a great reduction of workload required.
- The estimate  $y$  of measurand  $Y$  and its associated standard uncertainty  $u(y)$  are improved for non-linear models.
- A coverage interval for a corresponding stipulated coverage probability can be provided even when the PDF for the measurand  $Y$  cannot be approximated by a Gaussian distribution or a scaled and shifted  $t$ -distribution.
- A coverage factor is not needed for determining a coverage interval.

### 2.2.5.3 Uncertainty evaluation for CMM measurements

ISO/TS 23165:2006 provides a set of general guidelines and simplified equations regarding the evaluation of CMM test uncertainty (International Organization for Standardization, 2006). ISO/TS 15530-4:2008 specifies the requirements for the application of simulation-based uncertainty evaluating software to CMM measurements and also describes testing methods for such software and various general testing procedures (International Organization for Standardization, 2008). In ISO/TS 15530-3:2004 and ISO/DIS 15530-3:2009, an experimental approach for determining

CMM measurement uncertainty is described, which makes use of substitution measurements that are carried on calibrated workpieces with similar shape and size instead of the actual workpiece, and uses the obtained differences between the known calibration values and the measurement results to estimate the measurements uncertainty (International Organization for Standardization, 2004; 2009).

Apart from above standards, many other efforts have been made to improve the quality of uncertainty evaluation for CMM measurements.

Balsamo et al. (1999) proposed a system based on Monte Carlo method simulation called Expert CMM or ECMM. ECMM is built around the standard CMM software, as illustrated in Figure 2.15. It uses a random generator to sample from a known joint probability density function  $JPDF_p$  for the parameters  $p_i$ . An error simulator takes  $p_i$ , the geometrical information  $x_0$  from the part program, and other auxiliary values of influence quantities such as temperature, as its input. The error simulator makes perturbations  $\varepsilon_i$  to instances of  $p_i$ , by mimicking measurement errors in different groups of contributors. The points outputted from the error simulator are compensated by the CMM software and then calculated by the part program to produce final results  $y_i$ . The covariance matrix  $\Psi_y$ , or variance  $u^2(y)$  in case of one measurand, can then be evaluated. In the actual implementation, the covariance matrix  $\Psi_p$  for  $p_i$  is used instead of  $JPDF_p$ , because  $JPDF_p$  is difficult to derive in full. The actual ECMM implementation is focused on the geometric error contributor only, due to its complexity. Preliminary results were compared with measured deviations of calibration values and the obtained expanded uncertainties  $U$  (with coverage factor  $k = 2$ ) produced intervals that encompassed the calibration value in 89 out of 100 measurements, which is quite close to the theoretical value of 95%.

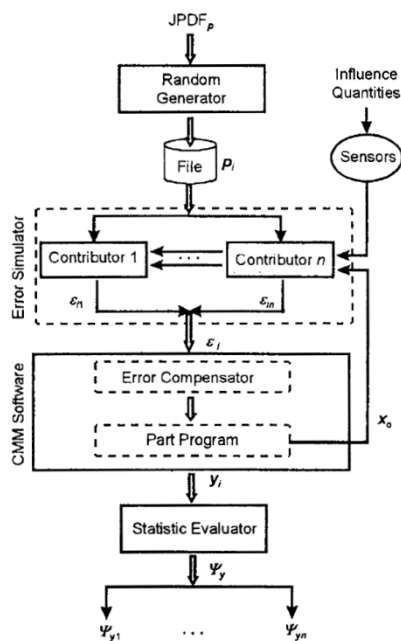


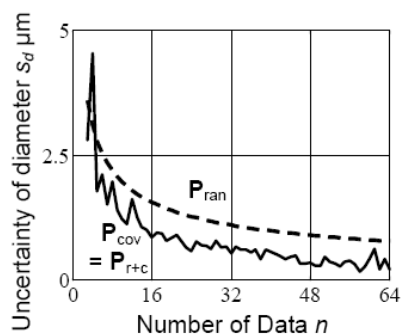
Figure 2.15 Overall scheme of ECMM (Balsamo et al., 1999)

The concept of using virtual instruments for uncertainty evaluation was proposed by Haitjema et al. (2001). A virtual CMM was developed based on Monte Carlo method (Van Dorp et al., 2001). This virtual CMM utilizes a surrogate signal that has the same autocorrelation as the original CMM error signal, to enable the errors of simulated points in one simulation to be correlated in the same way they are correlated on physical CMM. The implementation of this method was evaluated by comparing to real measurements, and the experimental results indicated that the predicted uncertainties were too high due to that the repeatability enters in all calibration components, i.e. the repeatability is included many times in a simulation while in a real measurement it enters only once. Another cause responsible for the pessimistic simulations was that the short-range errors are somewhat present in the larger range too.

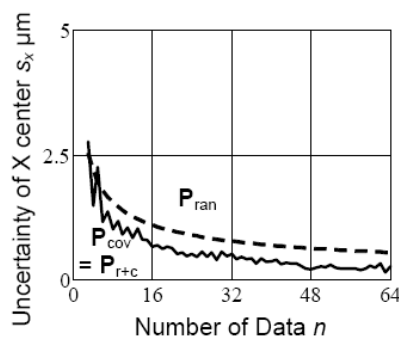
Sprael et al. (2003) proposed a statistical approach of uncertainty evaluation for CMM measurements. The method focuses on the estimation of instantaneous uncertainties, which are calculated from the residue of the least square fitting of the acquired points.

This method is limited to control of ISO 1101. The experimental application showed results that are close to the values obtained by repeated measurements.

Takamasu et al. (2004) theoretically analyzed the effects of unknown systematic errors for the uncertainty evaluation in feature-based metrology. This research was carried out by examining a circular feature with form deviations using a CMM. The form deviations of the circle were treated as the unknown systematic errors that propagate to the uncertainties of measured parameters. Three types of uncertainties are defined:  $P_{ran}$  when the form deviation is considered random function,  $P_{cov}$  when form deviation is correlated and calculated using the autocorrelation function, and  $P_{r+c}$  when form deviation is correlated but calculated without the autocorrelation function. The preliminary results reveal that when the circle is measured uniformly,  $P_{ran}$  and  $P_{cov}$  have same values, and the uncertainty has not only relationship to the number of measured data, but also relationship to the odd-even of measured data (see Figure 2.16), although this relationship differs for different features. When the circle is measured partially,  $P_{cov}$  is the best estimation, and other two are either over or under estimations in specified conditions.



(a) Uncertainty of diameter  $s_d$



(b) Uncertainty of X coordinate of center  $s_x$

Figure 2.16 Relationship between number of data and uncertainty of diameter and X coordinate of center in uniform measurement



Baldo and Donatelli (2004) proposed an improvement to the ISO/DTS 15530-3. The principle of this proposal is to find inconsistencies in the obtained average and standard deviation of each part, before they are used to estimate the Type A standard uncertainty.

A method of evaluating CMM probe uncertainty using FEA (Finite Element Analysis) modelling was proposed by Salleh et al. (2006). A FEA model of the probe was built, consisting of a stylus and probe system, with changeable materials and stylus length. The behaviour of probe was studied, in terms of the relationships between the stylus displacement and the triggering force and the stylus length. The obtained information was analyzed and used to estimate the uncertainty of measurement results.



Figure 2.17 Space frame

De Aquino Silva et al. (2009) designed a new form of space frame, as shown in Figure 2.17, for rapid evaluation of uncertainty of four-axis CMMs. The new space frame consists of a ball plate that incorporates seven high accuracy spheres, three of which are attached on the plate surface and other four on steel stems fixed to the plate. When applied in uncertainty evaluation, the artefact is set up on a rotary table on the CMM being tested. A local coordinate reference system for the space frame is established, with Z-axis being the rotation axis of the artefact. The centre coordinates of the seven

spheres are measured and defined as the nominal coordinates at the initial position. The nominal coordinates of the seven spheres at other rotated positions can then be calculated. Finally the space frame is actually rotated at step of 15 degrees and the spheres are measured at each different position. The differences between the nominal and measured values are then analysed to evaluate the uncertainties. The experimental application proved to be practical and faster than using traditional artefacts.

Cui et al. (2009) conducted a research to compare uncertainties from different evaluation methods of form errors, namely the least square method and the genetic algorithm-based method. The computation uncertainties to flatness and roundness were taken as examples in the research, and the results led to the conclusions: a) the differences between uncertainties from the two evaluation methods were very small but the least square method was considered better because the genetic algorithm-based method was a little unstable; b) the uncertainties to the form errors were mostly affected by CMM sampling hence the measurement strategies are very important.

Kruth et al. (2009) presented a Monte Carlo simulation based method for uncertainty evaluation for feature measurements on CMMs. It was pointed out that the feature form deviations are important and have critical influence to the uncertainties when the number of sampling points is limited. To overcome the difficulty of quantifying the form deviations, a profile simulator was introduced into the simulation scheme. The profile simulator utilizes a reference form database that is based on many actual measured profiles. The profiles are all standardized, for example circles to have roundness 1. When given required position, orientation, size, and magnitude of the form deviation, the profile simulator firstly generates a perfect circle according to the requirements, and then randomly takes a standardized profile from the database and multiplies it by the magnitude of form deviation, and lastly adds it to the perfect circle. The resulted profile is then sampled by a CMM simulator, which also adds random and

systematic hardware errors to the simulated measured points. Figure 2.18 shows the scheme of the error simulation method. Both the simulated measured points and the simulated true profile are fitted by a feature fitting module, and their differences are calculated and recorded. The whole process is repeated sufficient times to form a representative error distribution. Techniques for obtaining reliable coverage intervals for the form deviation and other feature parameters were also presented. The proposed method was fully implemented for circular profiles and was validated by measuring an actual workpiece on CMM. The results also showed that by taking into account the influence of form deviation, the uncertainty estimation became more reliable, especially with limited sampling.

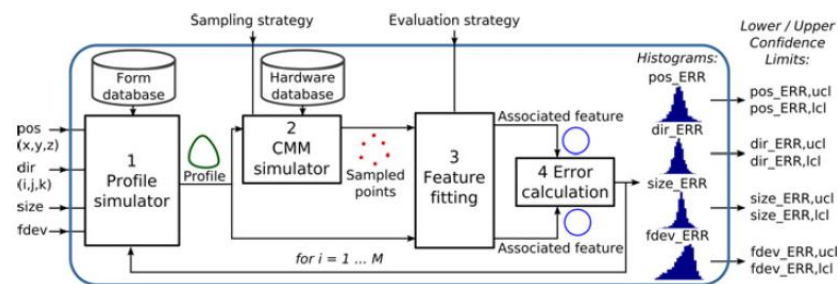


Figure 2.18 Scheme of the error simulation method

PUNDIT/CMM (Summerhays et al., 2002) and Virtual CMM are two commercial packages for uncertainty evaluation and error analysis. While PUNDIT/CMM is a stand-alone application based on SBC (Simulation by Constraints) method (Phillips et al., 1997), Virtual CMM is integrated with the native CMM measuring software to make use of the existing measuring programs. Beaman and Morse (2010) experimentally evaluated these two applications by comparing their prediction of uncertainties to that obtained in actual measurements of calibrated artefacts. Results indicated that the Virtual CMM software is more precise while the PUNDIT package has the ability of generating estimates for an entire class of machines that meet certain specified requirements.

## **2.3 Virtual coordinate measuring machine**

### **2.3.1 Introduction**

The concept of virtual coordinate measuring machine (VCMM) was created to refer to the mathematical models that numerically simulate a CMM, and can often be implemented as a computer program to evaluate or predict the reliability of the measurement results of the CMM. Over the recent years, several types of VCMMs with different functionalities were developed and many new ideas have been contributed to this concept. Generally, now the term VCMM may refer to various kinds of software applications that are designed to simulate or represent some aspects of physical CMMs.

Typically the existing VCMMs fall into two categories according to their purposes and approaches. The first kind of VCMM is basically a software tool intended for the evaluation of measurement uncertainty (or error, if not using an uncertainty framework). This kind of VCMM is essentially a mathematical model of the CMM measurement process. It takes into account various error/uncertainty sources, and based on the knowledge of the influence quantities, these sources of uncertainties can be combined together to generate the expanded uncertainty and also give the distribution of possible results.

The second kind of VCMM is the simulation of the actual measurement process and in particular, the inspection planning of a physical CMM. It focuses on representing the inspection planning methodology in the virtual environment, which enables the off-line programming of the CMM concerned. Through suitable interface, some VCMMs of this kind can be used to control a CMM and allow remote, shared access to a physical machine. As a virtual CMM, it also facilitates the low-cost, accident-free training/learning of complicated CMM operations.

The efforts and achievements in the research of above two categories of VCMMs are summarized in the following two sections. Some of the works are already mentioned or reviewed in previous relevant sections, hence for those works only brief descriptions are listed below.

### **2.3.2 Uncertainty evaluation oriented VCMMs**

Although some VCMMs in this category did not utilize an uncertainty framework but the traditional error system, the name is given due to the similarity of their general purpose and the trend in the current research.

In 1998 Pahk et al. reported their work on the development of a VCMM that can predict volumetric errors from measured parametric errors (Pahk et al., 1998). Ten sets of volumetric equations were derived to cope with ten kinematically different CMM models that cover a wide range of currently available CMM types. These equations abstract the calculation of volumetric errors incorporating the random error components. Several sets of transformation rules were established to transform all the measured machine parametric errors with respect to a specific reference point prior to calculating the volumetric error map. The probe error was also taken into account. A probe error map was constructed for a probe setup by measuring a reference ball. The value of the probe error can then be obtained from the error map according to the probing direction. The probe error and the CMM geometric error were combined into the integrated volumetric error using a proposed integrated volumetric error model. This model was verified by simulations and actual measurements of a ring gauge, and the results showed that the integrated volumetric error has better agreement with the practical experimental measurements than considering only the CMM geometric error.

Haitjema and Van Dorp et al. proposed to use the concept of ‘virtual instruments’ to

evaluate uncertainty in measurement (Haitjema et al., 2001) and developed a ‘virtual CMM’ particularly for CMM measurement (Van Dorp et al., 2001). This is basically a Monte Carlo method based simulation that uses a surrogate signal to mimic the correlations between the errors of all simulated points within one simulation.

Waldele and Schwenke (2002) reported their progress of the development of a ‘virtual CMM’ that may facilitate the automated calculation of CMM measurement uncertainties.

A group of researchers in NEDO VCMM team have been working on the development of international standard for virtual CMM since 1999 (Takamasu, 2002).

Peggs (2003) introduced the virtual metrology frame used to achieve accurate measurement and proposed the use of the concept of virtual systems. As one of the virtual technologies developed for advanced manufacturing and metrology, VCMM was described to be a solution to the profoundly difficult problem of uncertainty evaluation.

EMU is a software application based on the concept of virtual measuring machine (Jakubiec & Starczak, 2004). It evaluates the measurement uncertainties of CMM taking into account the simplifications of measuring strategy and the influence of the geometrical deviations of the measured workpiece.

PTB (Physikalisch-Technische Bundesanstalt) developed a Monte Carlo simulation based method around the concept of VCMM. This VCMM has been built into two CMM software packages from two German CMM manufacturers. As one of the four laboratories that are accredited to calibrate workpieces using this technique within the German Calibration Service (DKD), FEINMESS presented its practical experiences with the software tool (Trenk et al., 2004). This VCMM takes into consideration various

contributors of uncertainties from CMM, environment, probing process and workpiece. PTB provides an analysis tool called KALKOM to obtain the systematic parametric errors based on measurements of calibrated workpieces. The influence of ambient conditions is monitored by multi-channel temperature logging systems. All obtained geometrical parameters, temperatures, gradients, material constants and additional constraints are entered into the VCMM as input, via the software VCMM Tool provided by PTB. When a measurement is made on CMM and the actual result is calculated from the set of recorded points, the VCMM generates more sets of points by adding systematic and random deviations to the nominal coordinates. The CMM software then evaluates the further sets of points generated by VCMM to obtain representative samples of potential measurement results. The measurement uncertainties can then be calculated statistically from the samples. To ensure the input parameters always correspond to the actual conditions, the entire system is verified regularly using monitoring measurements of calibrated artifacts. In the first year as a DKD accredited lab, FEINMESS carried out 25 calibrations for customers with uncertainties calculated by VCMM, and regularly used the VCMM to optimize part programs regarding datum setting and probing strategy.

Calonego et al. (2006) recognized the importance of software fidelity and usability and proposed a user interface design for VCMM software applications. The design, which was developed using UML (Unified Modeling Language), splits the VCMM setup into Model, View and Controller (MVC) software modules, as shown in Figure 2.19. The MVC model improves the usability mainly with the View and Controller layers, and improves simulation fidelity with the Model layer. The simulation quality is expected to be improved by proper user guidance.

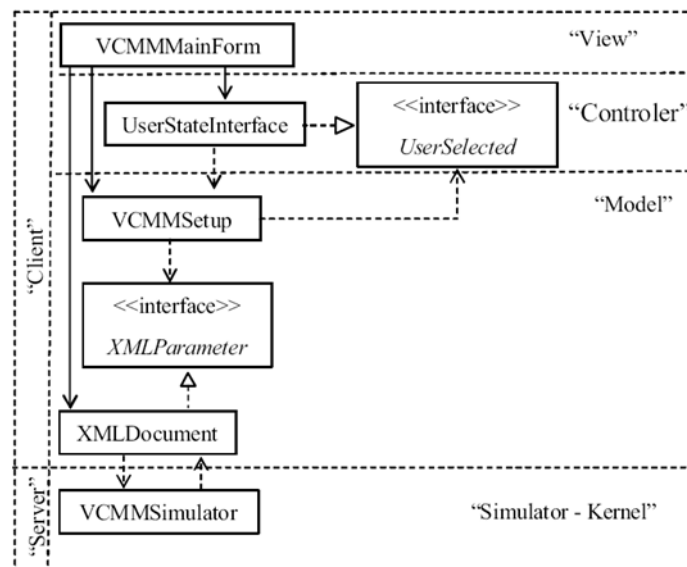


Figure 2.19 MVC model for VCMM

### 2.3.3 Inspection oriented VCMMs

VCMMs in this category may aim at different purposes, but they all have the common ground of simulating the CMM inspection, normally with 2D or 3D graphical representation.

Lin et al. (1999) proposed the idea of creating a VCMM to simulate the CMM part inspection by moving and touching a virtual probe in a virtual environment, in order to help improve the inspecting strategy and inspecting program.

Stouffer and Horst (2000) developed a VRML (Virtual Reality Modelling Language) model of CMM to assist in the remote monitoring of CMM inspection. The VRML CMM can be displayed in a web browser, and it is controlled by an open architecture inspection controller, via a NML (Neutral Messaging Language) socket connection



between the real world controller and the web browser. A Java applet on the remote access web page collects the current probe positions from a world model buffer in the controller, and sends the positions across the socket connection to update the VRML animation, via the External Authoring Interface (EAI) of the VRML plug-in. When a user visits the remote access page, the VRML model is downloaded to the local machine, and after that only current probe positions are sent, hence allowing very low bandwidth usage. Comparing to remote monitoring via only a camera system, VRML CMM provides additional 3D perspective of motions and events with much higher update rates.

Liu et al. (2002) proposed an internet-based measurement system that can lively represent the physical machine in a virtual environment at near real-time speed. To improve the stability and reliability of the system, it utilizes predictive control to decrease the influence of uncertain time delay, and uses autonomous control to prevent incorrect operations.

From 2004 to 2009, Chen, Yang and Wang et al. published a serial of works on their HVCMM, which features a haptic device and haptic modelling representation of the CMM inspection process, and mainly facilitates the accessibility analysis and path planning in the CMM inspection procedure (Chen et al., 2004; 2005; Yang & Chen, 2005; Wang et al., 2006; Wang et al., 2009). They also proposed to extend the haptic modelling to many important aspects of product development (Chen et al., 2005).

Calonego et al. (2004) discussed the implementation of an interactive virtual environment for CMM. The CMM was modelled using VRML, while the scene controlling mechanism was implemented in C++. The implementation utilized the software 'Avalon' which is not just a VRML browser but also provides a set of functionalities that aid in moving avatars, creating or removing scene objects during the

execution of program, etc. The experience with different kinds of user interfaces was also reported, such as virtual table (VT), graphic plate and passive stereoscopic visualizer. The system modules were designed to be able to execute in parallel and the virtual world may be controlled from a textual or graphical interface, or by using optic tracking devices.

## **2.4 Virtual Reality Modelling Language**

Virtual Reality Modelling Language (VRML) is a file format for representing interactive 3D scenes. It was originally termed as Virtual Reality Markup Language and first discussed at the First International Conference on the World-Wide Web in 1994. Urged by the need for a 3D web standard, a group quickly formed around a mailing list called 'www-vrml' and produced the VRML 1.0 specification. The name was changed to Virtual Reality Modelling Language to stress the importance of graphics. VRML is designed to be used on Internet, intranets and local systems, and is also intended to be used as a universal interchange file format for integrated 3D graphics and multimedia.

VRML 1.0 standard was limited to only being able to create static virtual worlds. This limitation severely hampered the potential wider adoption of VRML. Recognizing the necessity of bring life to the static virtual worlds, VRML 2.0 was developed to extend the standard with animation and interactivity support. In 1997 VRML 2.0 became an ISO standard (International Organization for Standardization, 1997), hence it is usually referred to as VRML97. VRML 97 provides a flexible and open platform for the creation of dynamic, interactive 3D scenes, and gained popularity in many domains such as design (Ma & Gao, 2009), interactive 3D animation, technical and scientific visualization, game (Jankovic, 2000), interactive web content, etc. There are a number of VRML authoring tools and browsers/players available for various platforms, and many VRML plug-ins for web browsers to choose from. Furthermore, many traditional graphical software packages (such as CAD software) now provide certain level of

compatibility to VRML and offer VRML import/export features.

Although VRML97 has been superseded by its successor X3D standard, it is backward supported by most X3D-savvy software, and is still actively used and widely supported by many important applications. For example, in MATLAB version R2009b, VRML97 is still the only supported format for virtual reality representation.

### **2.4.1 VRML features**

The content of a VRML97 virtual world may contain multiple shapes, each with properties like geometry, shading, texturing, and transformation. Elements like background, light source, sound, and fog may also be added to the scene. Other contents may include viewpoints, navigation controls, interaction controls and animations. VRML97 has a native event mechanism, and provides a scripting feature that can handle procedural control by using one of the supported scripting languages.

Elements in a VRML97 scene are described and contained in different kinds of nodes. Nodes may be hierarchically grouped and may be named and reused. A prototyping feature is also provided to allow user to extend the set of node types.

Comparing to traditional 3D modelling and animation, VRML has a distinct advantage of being intimate to the Internet since it was designed with the World Wide Web in mind. It facilitates easy online representation of 3D scenes. It is also lightweight, flexible and extensible, and can be integrated with other languages.

### **2.4.2 X3D**

The Web3D Consortium, formerly known as the VRML Consortium, developed the

X3D specification as a successor to VRML97 standard. X3D offers some extensions to VRML features, provides better application programming interfaces (APIs), and can use XML (Extensible Markup Language) for encoding.

X3D maintains very good backward compatibility to VRML. In addition to its XML syntax, it still retains the classic VRML encoding which can play most non-scripted VRML97 worlds with only minor changes. In addition, X3D files and VRML97 files may be converted to each other easily. Currently the latest X3D browsers usually can render VRML97 and most evolved VRML browsers have support to X3D as well.

### **2.4.3 Collision detection in VRML**

When representing and animating a 3D virtual world, inevitably sometimes objects in the virtual world may contact or collide with each other. However, VRML97 itself does not provide a comprehensive native collision detection mechanism. Although VRML97 defines a 'Collision' node, it can only detect the collision between the avatar (viewer) and the scene's geometry, not the collision between objects.

V-COLLIDE is a collision detection library developed by researchers in University of North Carolina. It utilizes a hierarchical approach. Once it is aware of the position of all objects, it carries out a fast sweep-and-prune operation to find out potentially colliding pairs of objects. And then for each potential contact pair, a pairwise test is taken to determine whether the two objects actually collided. V-COLLIDE remembers the positions of all objects and updates can be made to some or all objects' positions by telling V-COLLIDE the new placements. Hudson et al. (1997) attempted to extend the collision detection ability of VRML by interfacing the V-COLLIDE library to VRML browsers with a simple API, as shown in Table 2.2. However due to the lack of access to source code of any VRML97 browser, they were not able to actually integrate the

library to a VRML97 browser. Instead, a stand-alone, multi-body simulation was performed and the experimental results showed that the prototype of the library works at acceptable speed while consumes considerable amounts of memory. The system performance is generally linear in number of collisions.

<b>col_open</b>	initialize collision detection library
<b>col_create_object</b>	add a collidable object
<b>col_add_triangle</b>	add a triangle to an object
<b>col_finish_object</b>	build the OBB (oriented bounding box) hierarchy for an object
<b>col_clear_object</b>	destroy an object's geometry
<b>col_delete_object</b>	delete a collidable object
<b>col_activate</b>	turn on collision detection for an object
<b>col_deactivate</b>	turn off all collision detection for an object
<b>col_activate_pair</b>	turn on collision detection between two objects
<b>col_deactivate_pair</b>	turn off collision detection between two objects
<b>col_update_transform</b>	transform (rigidly) an object
<b>col_test</b>	perform collision detection
<b>col_report_collision</b>	report collisions

Table 2.2 V-COLLIDE API

Cortona3D Viewer (Cortona3D, n.d.) is a Web3D viewer which works as a VRML plug-in for popular Internet browsers. In addition to complete VRML97 support, Cortona3D Viewer provides additional nodes to extend the capabilities of VRML, including a proprietary implementation of an object-to-object collision detection extension to VRML, which can determine whether a given shape would encounter an obstacle in terms of another shape while to undergo certain kind of transformation (Cortona3D, n.d.).

## 2.5 Summary

This chapter reviewed the state of the art of CMM related techniques, focusing on the inspection planning and uncertainty evaluation methodologies and applications.

Particularly, the progress and trend in the research and development of VCMM were reviewed. Other related key technologies and standards were discussed.

As powerful tools as VCMMs may be, this literature review clearly revealed the lack of a comprehensive integrated solution for the desired tasks, i.e. a full featured virtual environment that allows user to simulate CMM operations and carry out virtual inspection (off-line planning and programming by operating virtual probes), then calculates desired geometric parameters of measured feature from the coordinates of points collected from the virtual measurement, whilst at the same time it is also able to evaluate the measurement uncertainty hence predict the actual performance on the physical CMM being simulated. This has confirmed that the proposed Advanced Virtual Coordinate Measuring Machine (AVCMM), which is designed to provide an integrated environment like stated above, is a novel approach with many innovations, such as an original collision detection engine that requires no modification to a standard VRML browser.

## **Chapter 3 Proposed AVCMM and Methodology**

### **3.1 Introduction**

Ever since the invention of CMM, enormous efforts have been put into the studies of increasing its efficiency, and evaluating and improving its accuracy. Along with the wide-spreading acceptance and application in industry, a great number of researches on the inspection planning and uncertainty evaluation have been conducted and many findings and innovations obtained over the past four decades. However these problems remain critical topics in the CMM related research filed and the current solutions are far from perfection, due to the complication of the system.

VCMM (Virtual Coordinate Measuring Machine), as a relatively new concept and a type of rapidly advancing powerful tool, has been proposed and developed to assist in different CMM related tasks. However according to the literature review in the previous chapter, the existing VCMMs only focus on a certain aspect of CMM related issues, i.e. only aiding in the inspection planning or only facilitating uncertainty evaluation. None of them provided a complete solution to these closely related problems due to their complexity. Since the inspection planning, the operation, the measurement, the calculation and the evaluation of uncertainty are a serial of events that happen in a CMM measurement and have substantial influences to each other, it is only natural that a user would want to perform all of them in a consistent environment in order to conveniently evaluate and improve measurement strategy. Urged by this need, an Advanced Virtual Coordinate Measuring Machine (AVCMM) is proposed in this thesis. AVCMM is a software package that aims to provide a comprehensive solution to CMM inspection planning and uncertainty evaluation problems, by creating an interactive virtual environment with organically integrated functionalities.

### 3.2 Overview of AVCMM

Generally the proposed AVCMM is an open architecture, configurable platform that can simulate the behaviours of different types of CMMs and can predict or estimate measurement uncertainties for user defined inspection plans hence allow user to improve or evaluate inspection strategy. Figure 3.1 shows a conceptual model of the AVCMM.

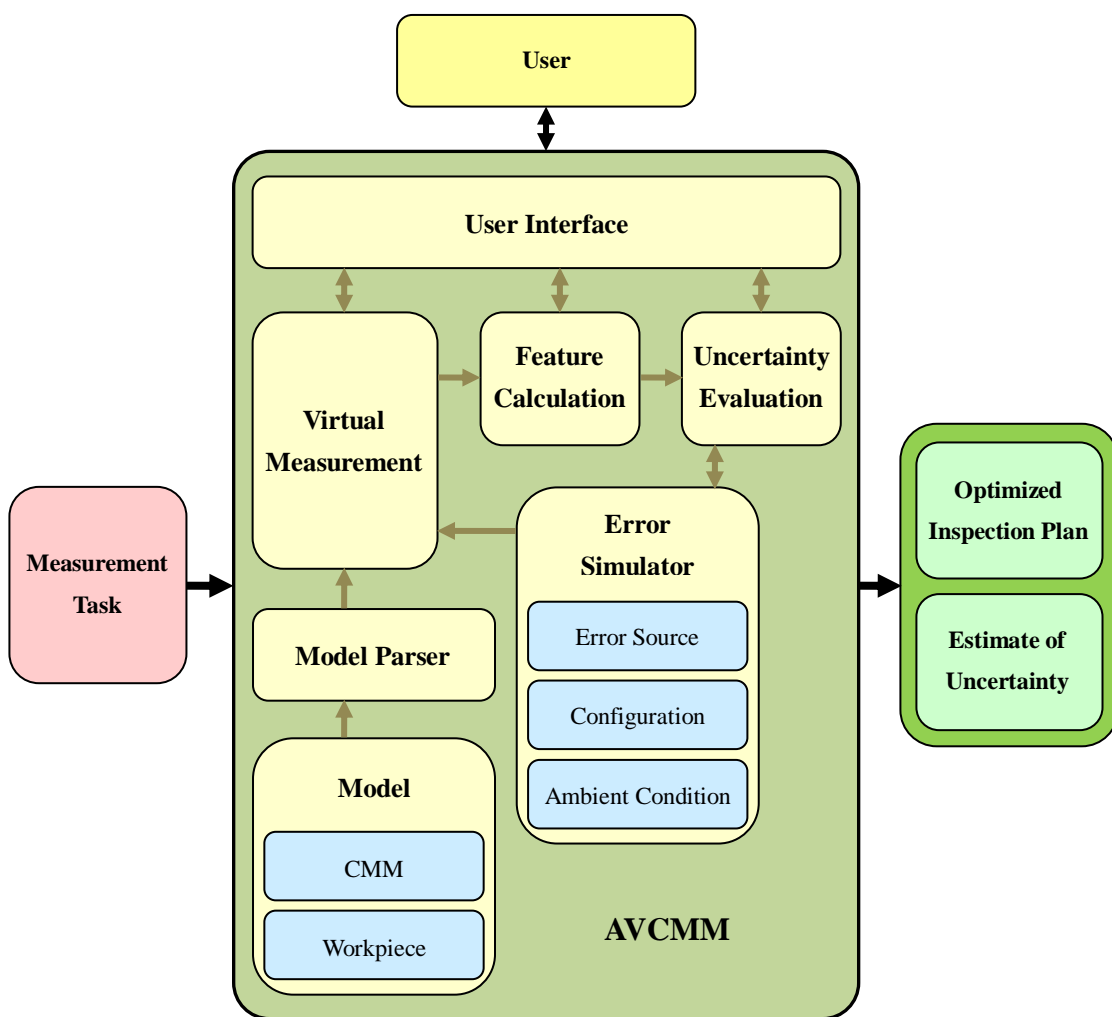


Figure 3.1 Conceptual model of proposed AVCMM



According to the literature review and suggestions from the industry, the intended functionalities and characteristics of the proposed AVCMM design may include the following:

- 3D visual representation of the CMM and the inspection process. Graphical representation is essential for a user-friendly interface. The dynamic 3D illustration of the inspection process provides an intuitive visual feedback of user control. The freely rotatable and movable view points (cameras) eliminate any blind spot of observation hence facilitates an easier user control, especially with such a complicated 3D spatial process. The virtual machine is able to load and visualize different models for various kinds of CMMs.
- Ability to load different workpiece models to the virtual machine. Like a real CMM, the virtual system is able to operate on different objects. Workpiece models can be selected by user and be loaded onto the virtual measurement table.
- Intuitive user control of the virtual machine. To make possible the manual inspection planning and measurement, a controlling interface is provided which allows the user to intuitively operate the virtual CMM and virtual probe in the same manner as on a physical machine. User control can be made with either standard input device, i.e. mouse and keyboard, or with extended input device such as joystick.
- Detection of contact and collision between the virtual probe and virtual workpiece. Collision must be detected in the inspection planning process in order to produce a safe, valid inspection path. Detection of the contact between virtual probe stylus tip and virtual workpiece is essential to obtain coordinates of measured points.

- Simulation of errors from various sources. Different measurement uncertainty contributors are considered, such as the CMM machine, probe, workpiece, inspection strategy, and environment, etc. All available information about the contribution of uncertainty is parameterized as the AVCMM input. Errors are randomly drawn from their estimated distributions and are added to the measurement along the inspection process. The propagation and influence of uncertainty components are modelled and simulated in the AVCMM measurement process, so the results of virtual measurement are expected to carry an approximated combined uncertainty.
- Calculation of desired parameters of measured feature. Parameters such as the diameter of a circle can be chosen by the user and calculated by the AVCMM based on the collected measurement points.
- Evaluation of the measurement uncertainty. Simulation of the measurement process is repeated using Monte Carlo method. This is not a traditional approach of Monte Carlo simulation for uncertainty evaluation. Instead of pure mathematical calculation, the whole process of the inspection including the calculation of feature parameters is repeated sufficient times in order to obtain a discrete representation of the distribution function for the measurand, from which the associated standard uncertainty and coverage interval are then determined.
- Exportation of inspection program. The recorded measurement path can be translated and outputted, either as standard DMIS code or directly to CMM controller console.

### **3.3 Work flow of AVCMM**

Measurements on CMMs are complicated processes and may vary for different types of

configuration. However CMM measurements share some common procedures and follow the same principles. Taking the widely employed Moving Bridge CMM and touch trigger probe as example, the general procedure of manual CMM measurement includes following essential components:

- Machine setup. Both the hardware and software of the CMM need to be started and initialized. A proper probe should be selected and installed and its orientation should be determined. The probe tip must be qualified/calibrated before actual measurement.
- Workpiece setup. Workpiece needs to be properly mounted on the CMM measurement table. A part coordinate system should be established and aligned.
- Inspection. In a manual measurement, user selects the type of desired feature in the CMM software and controls the probe to contact a set of appropriate points on the surface of the datum on the workpiece.
- Calculation of parameters. The desired parameters of the measured features can be calculated by CMM software based on the points collected in the inspection. Parameters of constructed features, such as intersection and angle, can be calculated from a serial of inspections.

As a tool to simulate and evaluate CMM measurements, the proposed AVCMM has a similar work flow with substantial additions. Figure 3.2 demonstrates the general work flow of the proposed AVCMM.

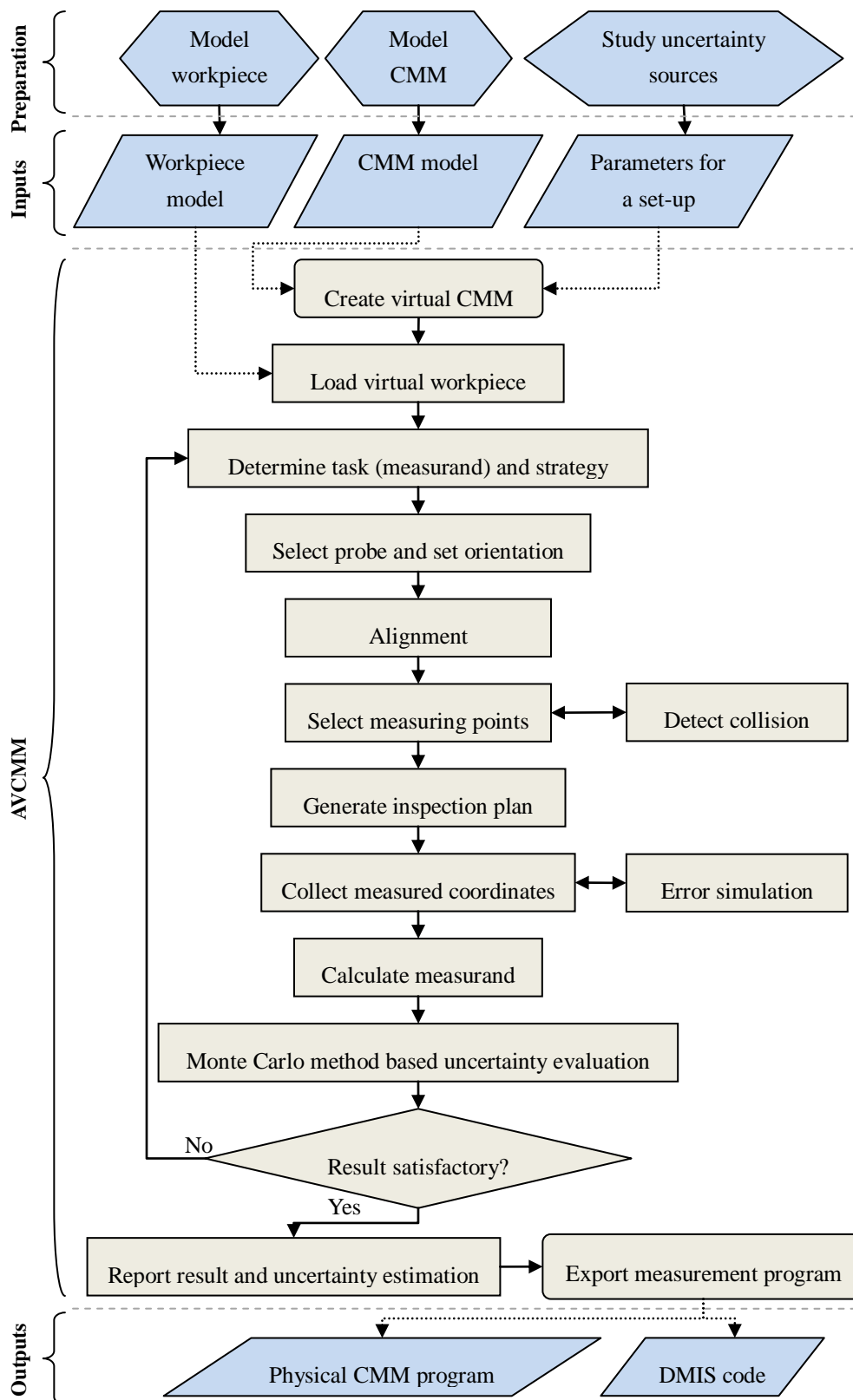


Figure 3.2 Work flow of the proposed AVCMM

In the preparation stage, extensive information should be obtained and modelled or parameterized as inputs to the AVCMM. For the CMM and probe system, the geometrical and kinematical models should be abstracted and represented using VRML, together with parameters describing the degree of freedom, capacity, moving and measurement speed, etc. For the workpiece, its nominal form characteristics (shape and size) are modelled using VRML. And for the most difficult part, the uncertainty sources, the acquisition of information is extremely hard or even impossible to be exhaustive. Table 3.1 shows a common classification of various uncertainty contributors for CMM measurements. Although efforts should be made to obtain as many uncertainty components as possible in order to make better estimation of combined uncertainty, in practical application only a subset of the sources may be feasible to be quantitatively described and thus considered.

<b>CMM/Environment</b>	<b>Workpiece</b>	<b>Sampling strategy</b>	<b>Evaluation strategy</b>
<ul style="list-style-type: none"> <li>• CMM geometric errors</li> <li>• Probe errors</li> <li>• Scale errors</li> <li>• Thermal errors</li> <li>• Calibration errors</li> <li>• Drift effects</li> </ul>	<ul style="list-style-type: none"> <li>• Form deviations</li> <li>• Roughness</li> <li>• Flexibility</li> <li>• Thermal expansion</li> <li>• Fixturing</li> <li>• Weight</li> </ul>	<ul style="list-style-type: none"> <li>• Number of points</li> <li>• Distribution of points</li> <li>• Probing speed</li> <li>• Probing directions</li> </ul>	<ul style="list-style-type: none"> <li>• Alignment algorithm</li> <li>• Filtering algorithm</li> <li>• Fitting algorithm</li> </ul>

Table 3.1 Measurement uncertainty contributors for CMMs

After the preparation of inputs, the proposed AVCMM establishes a virtual CMM environment from a specific CMM model. The virtual CMM should have the same geometrical and kinematical characteristics and conformation as its physical counterpart.

According to user selection, AVCMM loads the workpiece model into the virtual

environment and mounts it on the virtual measurement table. Fixturing process is optional in AVCMM since the virtual workpiece will not move in sampling process. The establishment of part coordinate system, or alignment, is simplified by transforming the workpiece model's local coordinate system obtained directly from the virtual environment. However the errors caused by fixturing and alignment may be simulated and superimposed in the virtual measurement, if such contributors are studied and associated probability distributions are approximated.

User decides the measurement task, i.e. the measurand, based on which a measurement strategy should be determined.

According to the measurand and measurement strategy, appropriate probe should be selected by user and installed on (loaded into) the virtual CMM. Proper orientation can be chosen and set for the probe. The probe tip does not have to be qualified as in the AVCMM, the system 'knows' exactly about the centre and radius of the probe tip. Similarly, the errors caused by calibration may be simulated if sufficient information is obtained.

Following the measurement strategy, user programs the inspection path by controlling the virtual probe to contact each desired points. This is powered by a novel collision detection engine that makes the virtual sampling possible. The potential collisions between the probe and the workpiece are also detected by the same engine, to guarantee a valid, collision-free inspection path.

The inspection path is recorded for later modification, repetition or exportation. During the inspection, various kinds of errors are simulated. Measured points, each with simulated errors, are recorded for later calculation.

The estimation of measurand is calculated from the measured points using filtering and fitting algorithms. This step may introduce a small error caused by the potential differences between the calculation algorithms in AVCMM and the actual CMM software. This is not an intended, simulated error, but a native error of AVCMM due to the lack of access to the source code of CMM software. Ideally if an open architecture CMM software package was available, then these calculations may be directed to the CMM software to eliminate this error.

To evaluate the uncertainty associated with the result, Monte Carlo method based simulation is adopted. In addition to traditional Monte Carlo simulation where pure numeric sampling and calculations are repeated, in the proposed AVCMM, the whole inspection process can also be repeated automatically with errors randomly drawn from their probability distribution and added during each repetition. The uncertainty of the measurand estimation is then analyzed from the obtained collection of potentially possible values of measurement result. Based on the uncertainty estimation, user may choose to adjust or change the inspection strategy to achieve better measurement quality.

Finally, apart from reporting the simulation results, it is very useful that the recorded measurement program may be exported as standard DMIS code, or directly interfaced to physical CMM for actual inspection.

## **3.4 Architecture of AVCMM**

### **3.4.1 Multitier architecture**

Multitier architecture, usually referred to as n-tier architecture, is a type of modular architecture in software engineering. It is a fast developing methodology and widely

adopted architecture in the development of complex software systems. For example, as shown in Figure 3.3, one of its most popular forms is the three-tier, DAL+BLL+PL (Data Access Layer, Business Logic Layer and Presentation Layer) architecture extensively employed in the development of Internet and enterprise applications, such as large-scale MISs (Management Information Systems).

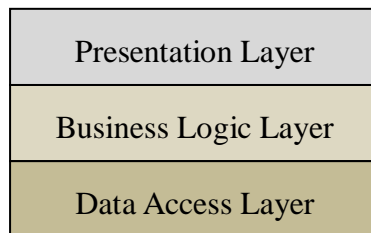


Figure 3.3 A typical three-tier architecture

In general, the concept of multitier architecture is to logically divide and group the modules of a system into ordered layers or tiers, according to their functionalities and responsibilities. The organisation of layers has following principles:

- Except the top layer, each layer is unaware of any higher layers and provides several services for its next higher layer to consume.
- Usually each layer hides its lower layers from the layers above. Except the bottom layer, each layer only knows about its next lower layer and can only use services defined by the next lower layer.

Any architecture design that complies with the above principles may be classified as multitier architecture. The number and composition of layers should be decided according to the requirement and complexity of a particular case.



Generally, multitier architecture aims to achieve high cohesion within each layer and low coupling between layers. The benefits of adopting multitier architecture include:

- Improved manageability of development of complex software system. Since a layer can only invoke services provided by its next lower layer, as long as interfaces are well defined, developers can focus on the implementation inside each layer and do not need to care about other layers.
- High maintainability. When needed, multitier architecture allows any of the layers to be updated or replaced independently.
- High reusability. The same service maybe provided to multiple different higher layers by one lower layer.
- Good robustness. Each layer functions with relative independence and normally the system does not have single point of failure.

As there is no perfect design, the multitier architecture has its drawbacks, for example, reduced performance and efficiency of the system.

### **3.4.2 AVCMM system architecture**

Due to complexity of AVCMM and the consideration for extendibility and scalability, multitier architecture is adopted for its development. Figure 3.4 shows the layout of the n-tier architecture design of AVCMM. It consists of three major layers, a model layer, a logic layer and an interface layer. The logic layer itself is a three-tier structure so the whole system becomes a five-tier architecture.

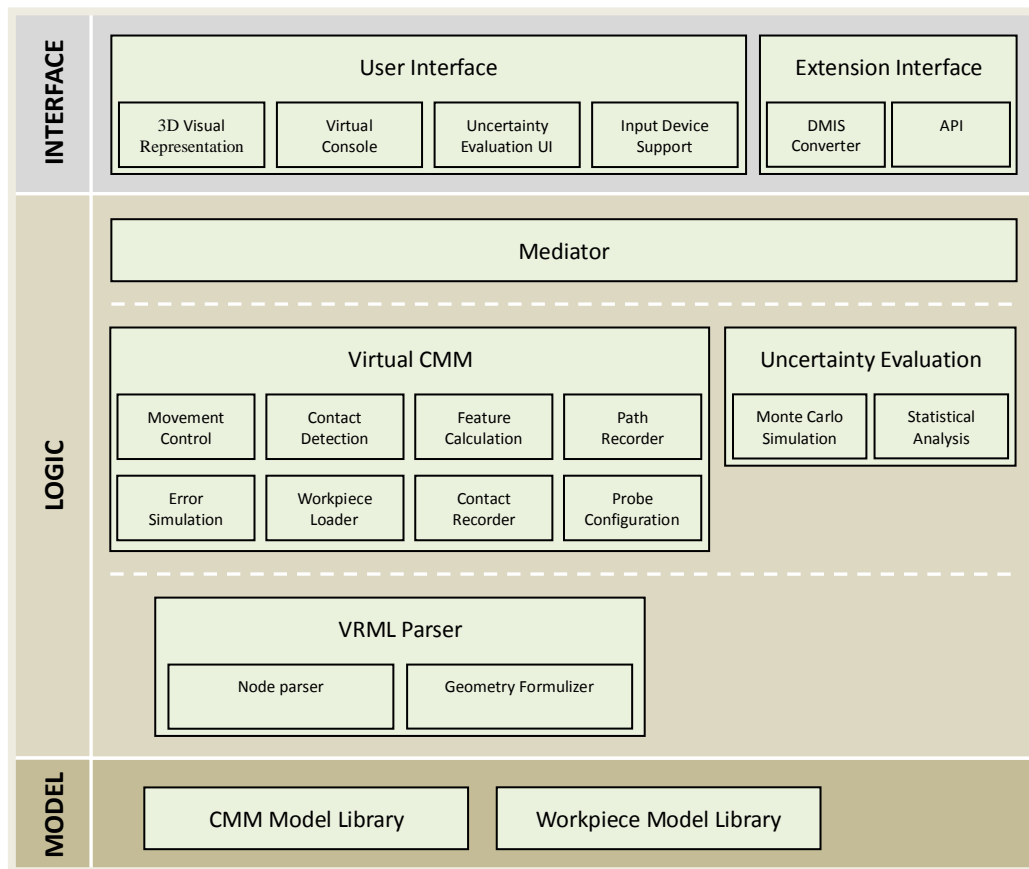


Figure 3.4 N-tier architecture of AVCMM

In the model layer there are two modules that organise all the VRML models used in the system. Different types of CMMs and workpieces are modelled into VRML files in a particular way discussed in Chapter 4, and archived in CMM Model Library and Workpiece Model Library respectively. Model layer is responsible for retrieving any specific model when required.

The logic layer itself has three sub-layers. At the bottom, a VRML Parser is responsible for exacting the geometrical information from the loaded VRML files and transforming the information into geometric equations or formulas. These formulas will later be utilized to enable collision detection in the virtual inspection.

The middle sub-layer of the logic layer is the core of AVCMM system. In this sub-layer the Virtual CMM module is located, and in turn it has several sub-modules that handle the essential tasks respectively, i.e. loading the virtual workpiece, configuring virtual probe, controlling the movement of virtual CMM, simulating various kinds of errors, detecting collision and contact, recording probed points, recording inspection path, and calculating parameters for desired feature. Another important module also located in this sub-layer is the Uncertainty Evaluation module, which carries out Monte Carlo simulation of the recorded inspection plan and calculates the uncertainty of measurement result from the statistical analysis of the simulation results.

In the top sub-layer of the logic layer, a Mediator module acts as a middleman that coordinates the communication and cooperation between the Virtual CMM module and the Uncertainty Evaluation module. It also abstracts and exposes the interface of the whole logic layer to the higher interface layer.

In the interface layer, the graphical user interface is presented, which consists of a Virtual Console that mimics the control console of CMM, a 3D Visual Representation window that displays the virtual environment, and a user interface for controlling the uncertainty evaluation. The support for various kinds of input devices is also implemented here. Apart from the user interface, other interfaces are provided for further extension of the system, for example, a DMIS converter that outputs the inspection path as standard DMIS code so the inspection program can be exported to other DMIS-savvy systems. Furthermore, the API of the whole AVCMM system is abstracted and defined so that further development or integration can be made upon the developed system.

### **3.5 Summary**

The characteristics and general work flow of the proposed AVCMM have been

presented and the underlying methodology has been introduced in this chapter. The proposed AVCMM organically combines the most important functionalities of existing VCMMs, namely the inspection planning and uncertainty evaluation, into an integrated environment and improves the efficiency and usability of both. The general architecture of the system is presented.

The theories and design for the proposed AVCMM system are discussed in detail in next chapter.

## Chapter 4 Design of AVCMM

### 4.1 Introduction

In Chapter 3 the general functionalities, characteristics and conceptual architecture of the proposed AVCMM have been presented. A layered, modular design is adopted to realize such a complicated system. In this chapter, the design and theories for important modules of the system are introduced and discussed in detail.

### 4.2 Geometrical and kinematical design of CMM model

As discussed in the literature review, there are many types of CMM that have kinematically different configurations. As shown in Figure 4.1, most of the commercially available CMMs can be abstracted and grouped into five models: a) The cantilever, gantry, L-shaped and moving bridge CMMs clearly all share the same kinematical model and all can be considered as a moving bridge CMM; b) Fixed bridge CMM; c) Moving horizontal arm CMM; d) Fixed horizontal arm CMM; e) Column CMM.

When modelling the five types of CMM shown in Figure 4.1 in VRML, following constraints should be taken into consideration:

- A machine coordinate system should be defined and its relationship to the native VRML coordinate system should be established. This is normally achieved by constructing a transformation matrix  $T$  and a translation vector  $L$ , so that

$$\begin{bmatrix} X_{CMM} \\ Y_{CMM} \\ Z_{CMM} \end{bmatrix} = T \begin{bmatrix} X_{VRML} \\ Y_{VRML} \\ Z_{VRML} \end{bmatrix} + L \quad (4.1)$$

where subscript ‘CMM’ denotes coordinates in the established machine coordinate system and ‘VRML’ denotes coordinates in the VRML native coordinate system. Matrix  $T$  describes the rotational direction between the two coordinate systems and vector  $L$  is the distance between the two origins.

- The travel range of each axis in the VRML model should comply with the measurement capacity or volume of the CMM being modelled. This is to ensure that location related uncertainty can be properly taken into consideration later in the error simulation.
- The positions of the three axes  $X_{axis}$ ,  $Y_{axis}$  and  $Z_{axis}$  in the machine coordinate system should correspond to the actual coordinates of the CMM  $(X_a, Y_a, Z_a)$  with a simple relationship:

$$\begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = \text{diag} \left( \begin{bmatrix} X_{axis}(X) & X_{axis}(Y) & X_{axis}(Z) \\ Y_{axis}(X) & Y_{axis}(Y) & Y_{axis}(Z) \\ Z_{axis}(X) & Z_{axis}(Y) & Z_{axis}(Z) \end{bmatrix} \right) = \begin{bmatrix} X_{axis}(X) \\ Y_{axis}(Y) \\ Z_{axis}(Z) \end{bmatrix} \quad (4.2)$$

where *diag* means get the diagonal of a matrix. This way, the controlling of the CMM movement and the retrieval of current coordinates are interfaced between the VRML world and other modules with a single vector  $(X_a, Y_a, Z_a)$ , which can be easily transformed into the VRML native coordinate system using Equation (4.1).

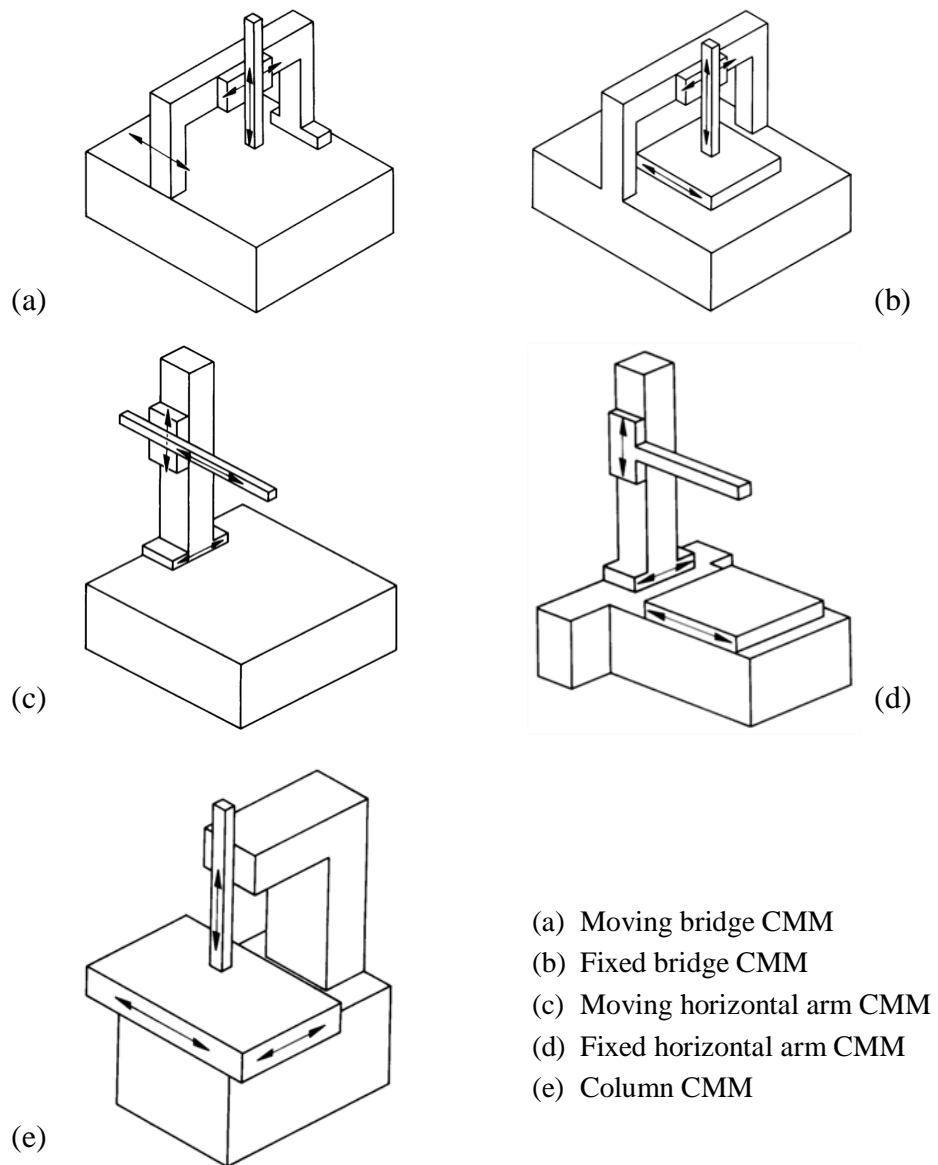


Figure 4.1 Five models of CMM

- The axes X, Y, Z of the CMM model should be able to travel individually and simultaneously. The relationships between each axis must be considered in order to construct an efficient model in terms of using as few parameters as possible to

decide and describe the state of the machine. Figure 4.2 demonstrates an example of bad model design that uses a flat layout and thus requires the three axes to be controlled by three vectors, one for each axis, containing duplicated information. In VRML, an object is composed of a set of basic shapes. Each shape is embedded in a Transform node that has a translation property controlling the position of the node and a rotation property controlling the orientation of the node. Transform nodes can be grouped together, can be children of other Transform node and can have their own children Transform nodes. The translation and rotation properties are all relative to that of the father Transform node. A better design of the modelling of CMM should make use of the hierarchical structure of VRML nodes and implement relative movement using the relative translation between father and children nodes.

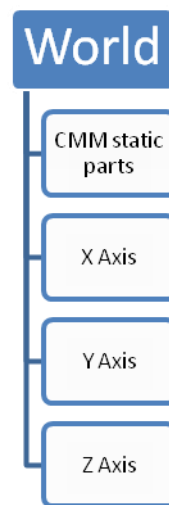


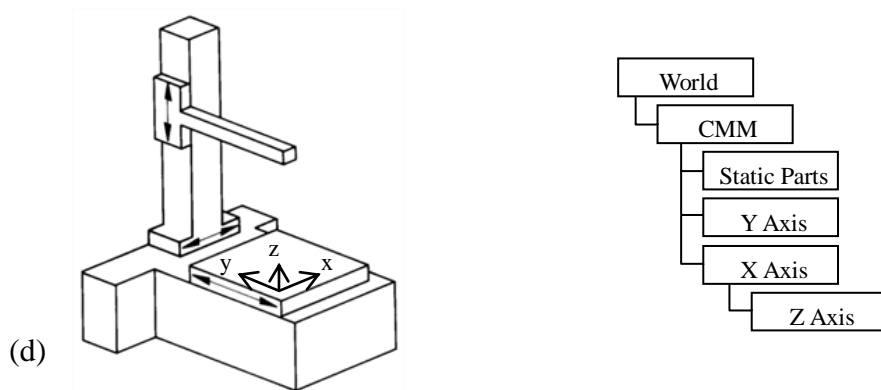
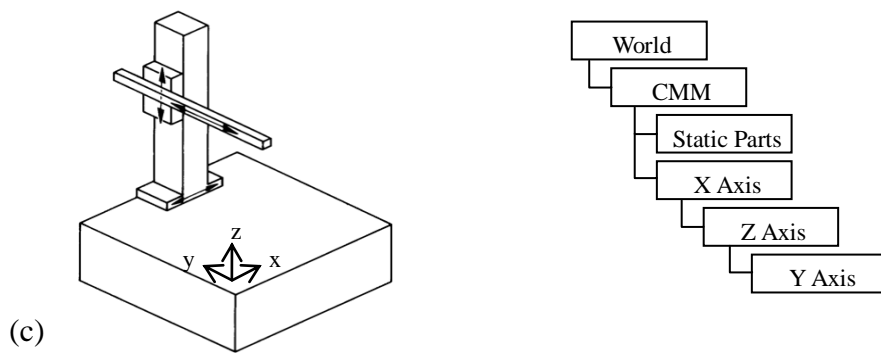
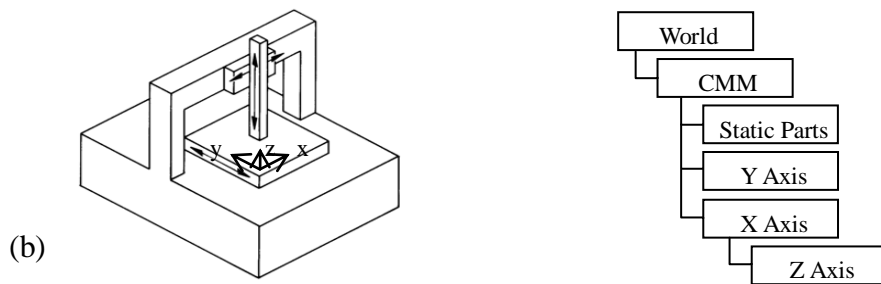
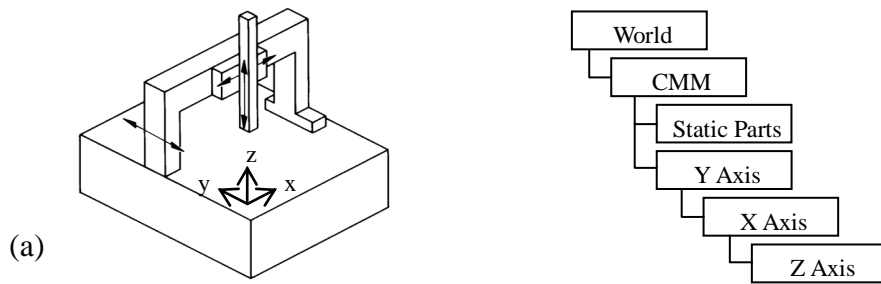
Figure 4.2 An example of bad VRML model design

Based upon the above considerations, VRML models, as shown in Figure 4.3, are designed for each of the five types of CMM listed in Figure 4.1. Although for each type of the CMM there might be multiple different machine coordinate system



configurations exist, they are all equivalent in terms of VRML structure. For each type, different machine coordinate system can be obtained by using different transformation matrix  $T$  and translation vector  $L$  in Equation (4.1). Thus here only one coordinate configuration is used for each type of CMM, for demonstration. However, for different CMM types the structure of VRML model may vary. For example, in a moving bridge CMM (Type a), axis Z is attached to axis X and in turn the axis X is attached to axis Y. So when axis Y moves, both axis X and axis Z move along with it in the Z direction, and similarly when X axis moves in X direction the Z axis moves with it. It is thus very reasonable to model Z axis as child of X axis and X axis as child of Y axis. In a fixed bridge CMM (Type b), however, the situation is different. Z axis is still attached to X axis so Z axis is still the child of X axis, but the Y axis is detached as a moving table, so Y axis becomes the sibling of X axis. In the same way, all five types are modelled as shown in Figure 4.3, and from these abstractions it becomes clear that although fixed bridge (Type b) and fixed horizontal arm (Type d) CMMs look and work very differently, they actually share the same kinematical model and hence same VRML design.

Probing systems are modelled as changeable parts attached to one of the axes, just like in a usual CMM. As reviewed in Chapter 2, various types of probing systems may be installed on a CMM. In this research we focus on the widely used probing system with touch-trigger probes and two rotational axes. Figure 4.4 demonstrates the two rotational axes of a probing system, conventionally named A axis (tilting) and B axis (rolling). The tilting joint is modelled as the child of rolling joint so it rolls with the latter but tilts by its own. The rotation property of VRML nodes is utilized to implement the father-child relative rotational movements.



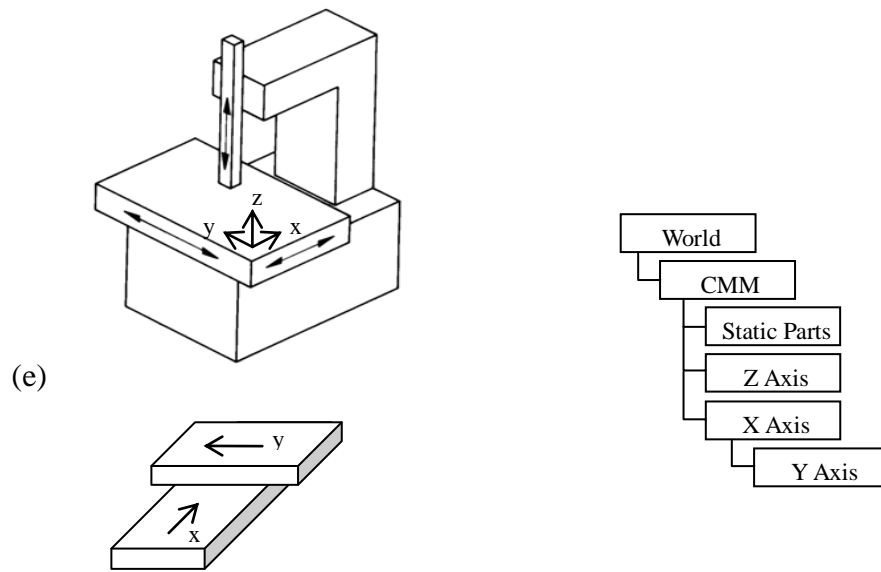


Figure 4.3 VRML model designs for five types of CMM

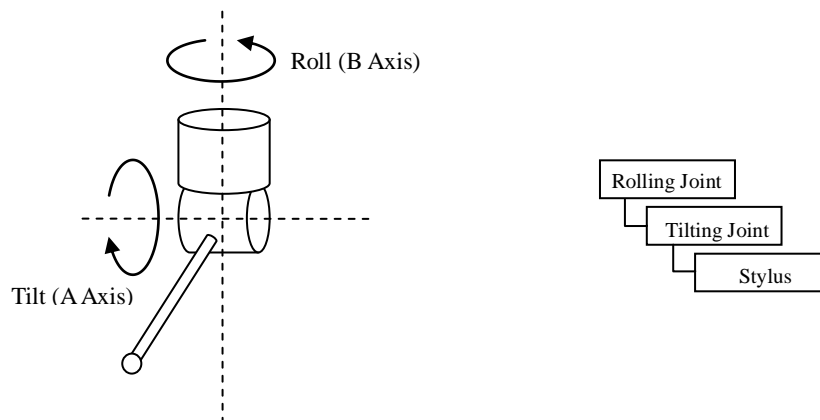


Figure 4.4 A probe model

Taking moving bridge type as an example, the complete VRML model design for a CMM with a probe attached would look like that shown in Figure 4.5.

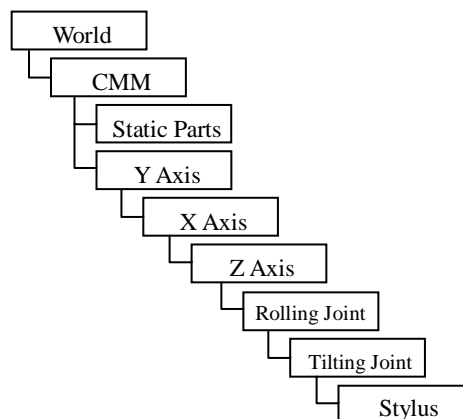


Figure 4.5 VRML model design for moving bridge CMM with 2-axis probe

### 4.3 Collision detection

Like the CMM itself, the workpiece being measured in the AVCMM is modelled in VRML, using four basic geometry nodes, box, cone, cylinder and sphere, and three advanced nodes, elevation grid, extrusion, and indexed face set. In order to detect the contact or collision between the workpiece and the probe, the collision between the nodes composing both objects should be detected. However in VRML97 standard, the only collisions being tested are those between the user avatar and objects in the virtual world, hence that a standard VRML browser cannot detect collisions between objects. To overcome this problem, three approaches of adding collision detection ability to the AVCMM are considered in this research: using established 3D collision detection algorithms, adopting an extended VRML browser with object-to-object collision detection interface, creating a new special purpose collision detection engine. The three approaches and their advantages and limitations are discussed in following sections.

### 4.3.1 3D collision detection algorithms

Collision detection for objects in 3D scenes has been studied in depth in the fields of CAD, computer games, robotics, computer graphics, computational geometry, etc. Many different methods have been proposed for a variety of applications and most of them make certain assumptions about the objects of interest and provide solutions according to the application domains (Badawy & Kelash, 2003).

When a 3D scene is rendered, either in a browser or an application, the objects in the scene are usually tessellated by polygons or triangles. Hence most existing algorithms for 3D collision detection are based on the test of overlap between pairs of polygons or triangles. Polygons within the same rigid object do not move relative to each other, so the comparison is between different objects. Suppose during certain time in a 3D scene, there are  $N$  collidable objects and of which  $M$  objects are moving and the rest  $(N - M)$  objects are static, then the number  $n$  of object pairs that need to be tracked is given by:

$$n = \frac{M(M - 1)}{2} + M(N - M) = NM - \frac{M^2}{2} - \frac{M}{2} \quad (4.3)$$

Similarly if all the  $N$  collidable objects are moving then

$$n = \frac{N(N - 1)}{2} = \frac{N^2}{2} - \frac{N}{2} \quad (4.4)$$

The algorithm time complexity for checking the relationships between the  $n$  pairs of objects at each step is  $O(N^2)$ , which is not efficient and the calculations become very time consuming as the number of objects grows. To improve the performance of the

algorithm, the number of pairs to be compared should be reduced. Most existing methods utilize some kind of hierarchical bounding volume scheme to achieve this purpose. The sweep-and-prune algorithm (Cohen et al., 1995), also known as sort-and-sweep, uses axis-aligned bounding boxes (AABBs) to surround each collidable object. As objects move, their bounding boxes are checked for intersection. As these bounding boxes are axis-aligned, they are easily projected onto the X, Y and Z axes, and only if the projections of two bounding boxes overlap in all three axes, the two bounding boxes overlap and their corresponding objects are marked for further more time consuming pairwise test. Sweep-and-prune algorithm exploits temporal coherence as it is very likely that objects only move slightly from frame to frame, therefore their spatial relationship does not change much between successive steps. For this reason insertion sort may be used to update the sorted list of bounding volume intervals.

In the pairwise test, a similar hierarchical approach (Gottschalk et al., 1996) may be used to improve efficiency. First find the approximate part where the collision happened, which contains only one or a very few primitives, and then perform exact intersection test between the triangles in the overlapping parts.

Many algorithms have been implemented and available as libraries and packages, such as DEEP (Kim et al., 2002), H-COLLIDE (Gregory et al., 1999), I-COLLIDE (Cohen et al., 1995), IMPACT (Wilson et al., 1999), PIVOT (Hoff et al., 2002), PQP (Larsen et al., 2000), RAPID (Gottschalk et al., 1996), SWIFT (Ehmann & Lin, 2000), SWIFT++ (Ehmann & Lin, 2002), V-COLLIDE (Hudson et al., 1997), etc. However, to apply these packages to the VRML environment of AVCMM system, the application would require interface to the object rendering mechanism in a VRML browser at a level low enough to access the triangles in tessellations. This would require either the source code of a VRML browser so that necessary modification can be made, or proper API to be provided by the browser so that triangle information is exposed. Even if such

requirements were fulfilled, a limitation of this approach is that a specific or modified VRML browser must always be used wherever the AVCMM system is deployed.

### 4.3.2 Object-to-object collision detection interface

VRML browsers can be standalone applications for viewing VRML scenes, and usually can also be implemented as plug-ins for standard Internet browsers, such as Internet Explorer, Netscape, Mozilla Firefox, etc., and enable Internet browsers to display 3D VRML scenes along with other contents right in the web pages. Popular packages include BitManagement's BS Contact, Octaga, Orbisnap, Vivaty (formerly Flux) Player, Cortona3D Viewer, Cosmo Player VRML Plug-in, and OpenVRML, etc. Among these packages, Cortona3D Viewer extended the standard VRML97 with a set of enhanced features (Cortona3D, n.d.), including an object-to-object collision detection interface.

The interface is implemented around two ECMAScript objects, namely 'Collidee' and 'Collision'. Object Collidee represents a shape being transformed, i.e. moved, rotated or scaled, and object Collision indicates the point where the shape contacts with another shape in the case of a collision. Table 4.1 lists the properties and methods of these two objects.

For the Collidee object, property 'body' points to the shapes being checked for collision and 'position', 'orientation' and 'scale' describes the current/initial state of the shapes. Properties 'size' and 'offset' are used for constructing an imaginary shape when 'body' points to null. The collision property references to a 'Collision' type object that stores the contact point of last collision. The properties 'scenery' and 'ignore' reference to objects that should be included or excluded for collision detection, respectively. The method 'moveTo' performs the collision detection while transforming the shape according to its input parameters, which describe the final state of the transformation. If

no collision occurs at both the initial and final position, or at any position interpolated in between, 'moveTo' returns 'true' and updates the Collidee object with the new state. Otherwise, it returns 'false' and stores the information about the contact point in a Collision object.

<b>Collidee</b>		
<b>Properties</b>		
<b>Type</b>	<b>Name</b>	<b>Permission</b>
SFNode/MFNode	body	
SFVec3f	position	
SFRotation	orientation	
SFVec3f	scale	
SFVec3f	size	
SFVec3f	offset	
Collision	collision	read-only
Collidee	scenery	
SFNode/MFNode	ignore	
<b>Methods</b>		
Boolean moveTo(SFVec3f position, SFRotation orientation, SFVec3f scale)		
<b>Collision</b>		
<b>Properties</b>		
<b>Type</b>	<b>Name</b>	<b>Permission</b>
SFVec3f	point	read-only
SFVec3f	normal	read-only
Number	faceIndex	read-only
MFNode	path	read-only

Table 4.1 Objects Collidee and Collision

For the Collision object, the 'point' and 'normal' properties store the coordinates of the contact point and the normal vector at that point respectively. The 'faceIndex' property is only for 'IndexedFaceSet' nodes and it indicates which face of the IndexedFaceSet contains the contact point. The 'path' property describes a path on the hierarchical tree of nodes, leading from the one referenced in the 'body' property of the Collidee, and to the one containing the face that collided.



The object-to-object collision detection extension described above provides a convenient interface to develop applications upon it. However in the context of AVCMM deployment, adopting this package would lead to less flexibility as the Cortona3D Viewer must be included for all deployments, since the essential function of the system relies on the package. Especially in the situation of releasing AVCMM as web application or web service, the binding to a particular VRML viewing program would require all clients to install additional software instead of using a standard browser that would be otherwise satisfactory. Furthermore, another restriction is that by adopting Cortona3D Viewer package, AVCMM has no control over the quality of collision detection in terms of precision, accuracy and speed, as it is a proprietary implementation.

### **4.3.3 Design of a collision detection engine for AVCMM**

To overcome the drawbacks of using third party libraries or a particular extended browser, a new collision detection engine is designed and developed for the AVCMM system. To achieve universal compatibility and easy deployment, the browser is released from the responsibility of collision detection and the function is implemented inside the AVCMM.

The principle for testing two polyhedra for intersection is simple. Suppose two polyhedra in a comparison pair are A and B, to check whether they overlap, first test if any of A's edges intersect with any of B's faces, then test if any of B's edges intersect with any of A's faces. If any edge of one polyhedron intersects with any face of the other polyhedron, these two polyhedra must have collided with each other. Otherwise they did not collide. This algorithm requires huge amount of calculation and normally cannot achieve real-time responsiveness when the number of objects increases. To help improve performance, certain strategies are adopted to reduce the number of objects

being tested for collision in the AVCMM system. First, since the virtual CMM is a relatively simple and predictable environment, we only mark a portion of all objects as ‘contactable’, i.e. only test those that might potentially contact or collided with other objects. For example, in a moving bridge CMM, if the movement ranges of three axes and the size of workpiece are constrained by the capacity of CMM, then the contactable objects may include the measurement table, workpiece, fixtures, probe and Z axis (the vertical spindle). This way all unnecessary parts are excluded for collision detection and the total workload of the algorithm is reduced. Second, bounding volume based hierarchical methods help to further reduce the number of objects being tested. There are many types of bounding volumes to choose from, such as the previously mentioned Axis-Aligned Bounding Box (AABB), Oriented Bounding Box (OBB), Bounding Sphere and Discrete Orientation Polytope (DOP), etc. To achieve better efficiency, the bounding volume should be as close to its surrounded object as possible and the test between volumes should be simple. In the environment of a virtual CMM, in many cases, most objects are aligned with the 3 axes of the coordinate system, hence AABB provides good approximation to object while requires very simple calculation for test. However, in some cases, for example, when the probe is tilted and rotated to an angle, as in Figure 4.6 (b), then the AABBs must be recalculated, and the reconstructed AABBs contain big empty corners around the oblique objects. For this kind of setups, OBB performs better as it can always keep tightly close to the surrounded object (see Figure 4.7), though at the cost of slightly more time for construction of the OBBs and much more time for testing intersection between two OBBs.

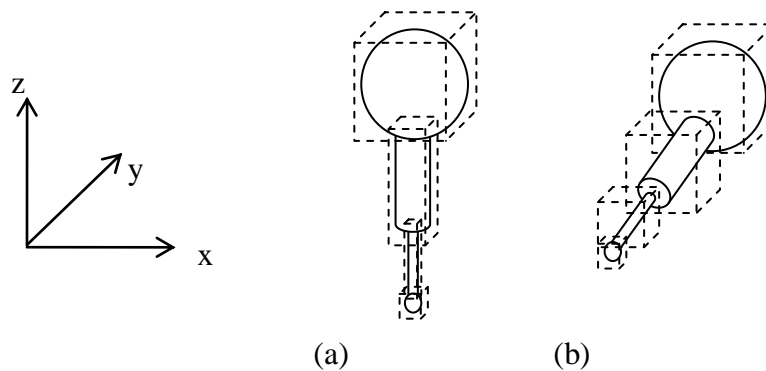


Figure 4.6 AABB for a probe at different orientations

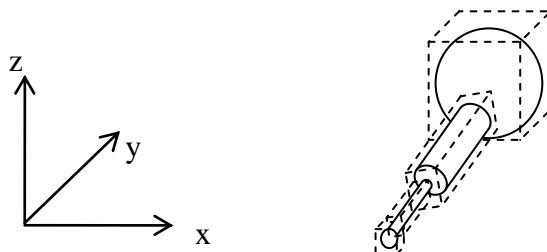


Figure 4.7 OBB for a probe at a tilted orientation

With a touch-trigger probe attached to the measuring axis, CMMs generally collect data by touching the workpiece with the probe tip. Although the probe tip is made very accurate, the radius and position of the tip relative to the machine coordinate system must be determined by qualification of the probe before measurement. For a physical machine, qualifying a probe is usually done by measuring a very accurate sphere, however in the AVCMM system this step can be omitted as the position and radius of

the tip are already known by the system from its VRML model. As shown in Figure 4.8, considering the surfaces of both the workpiece and the stylus tip as ideal, when the tip contacts the workpiece, the coordinates of the tip are mathematically corrected to the actual point of contact.

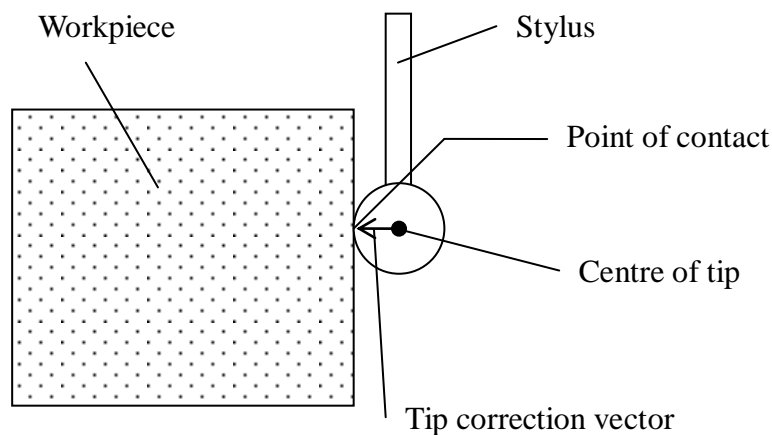


Figure 4.8 Tip correction vector

When measuring a workpiece in the AVCMM system, let  $v_m$  be the measurement speed of the probe and  $N_m$  be the number of movements (steps) the system can perform per second, then the length of each step  $\delta_m = v_m/N_m$ . Restricted by the time cost of the algorithms at each step, the current system can perform roughly tens to hundreds steps per second. If  $v_m$  is set to be close to real measurement speed, which is usually at the level of several millimetres per second, then the step length  $\delta_m$  is much longer than a normal CMM's resolution  $\delta_r$ . This way the resulted coordinates is far from accurate enough to represent a sample point. To overcome this problem and achieve same level of resolution as the modelled CMM, an additional routine is used so that each time when a contact is detected, the step  $\delta_m$  between the contacted position

and the last position before contact is divided into  $N_s$  number of sub-steps, so that the length of each sub-step  $\delta_s = \delta_r = \delta_m/N_s$ . Then, as shown in Figure 4.9, the tip is re-tested for contact with the workpiece along the path of  $N_s$  sub-steps to find the point of contact which has the same resolution as the modelled CMM. This routine is performed without updating the 3D graphical view at each sub-step so the serial of testing can be finished very rapidly.

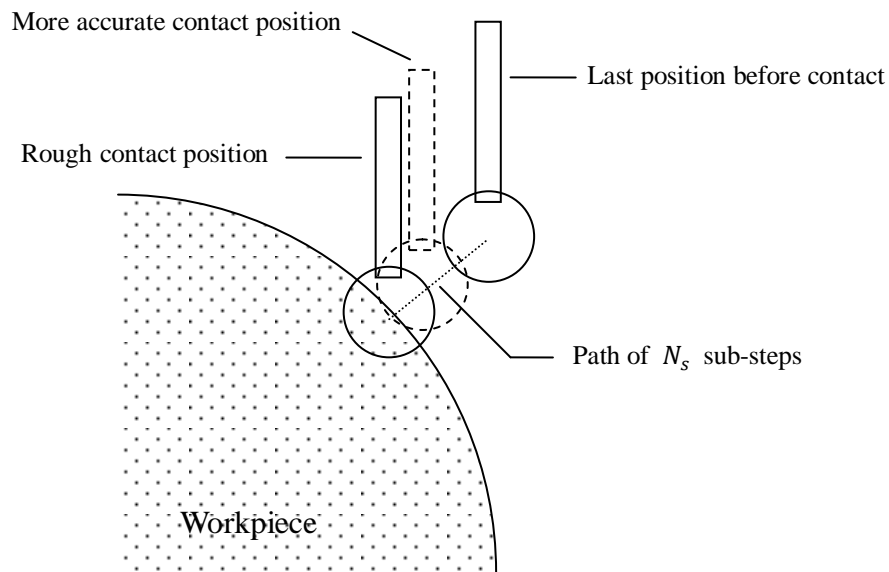


Figure 4.9 Determination of more accurate contact position

## 4.4 Error modelling

The point obtained from the collision detection is ‘accurate’ at the level of CMM resolution. Errors should be simulated and added to the sampled points so that the measurement results can be used for the purpose of uncertainty evaluation. A variety of error sources are listed in Table 3.1 in Chapter 3. At this stage, we mainly focus on the simulation of the CMM geometric errors and the probe errors as their complexity alone

requires nontrivial investigations.

#### 4.4.1 CMM geometric error modelling

The geometric error of a CMM depends on many parametric components and varies at different measurement location, therefore it is very difficult to simulate. In this research we proposed two different approaches to be used for the modelling of geometric error. The first one is to calculate the volumetric error of a CMM based on its kinematic configuration and measured parametric error components. The second one makes use of finite element method, and obtains the CMM geometric error from structural simulation.

##### 4.4.1.1 Volumetric error modelling

Most 3D CMMs are based on a frame with three orthogonal linear axes, which form a Cartesian coordinate system. For each moving axis, its associated errors can be described by specifying six degrees of freedom: three translational errors including linear position error and mutually orthogonal straightness errors, and three rotational errors in rolling, pitching and yawing directions, as shown in Figure 4.10. For three axes there are 18 geometric errors, each of which can be expressed as a function of the nominal position of the carriage along the moving axis. In addition, the axes of the CMM are not perfectly perpendicular to each other, and the squareness errors between the three pairs of axes also form part of the geometric errors (Lim & Burdekin, 2002). Hence there are 21 parametric error components that contribute to the volumetric error of a CMM. The grouping and notation of these parametric errors are listed below:

- Positional errors, or called linearity errors, are errors occur along the moving directions of the axes. Since 3D CMM has three axes, there are three positional errors,  $\delta_x(x)$ ,  $\delta_y(y)$ , and  $\delta_z(z)$ .

- Straightness errors are those that occur perpendicularly to the moving direction of one axis and in the directions of the other two axes. Hence there are six straightness errors,  $\delta_y(x)$ ,  $\delta_z(x)$ ,  $\delta_x(y)$ ,  $\delta_z(y)$ ,  $\delta_x(z)$ , and  $\delta_y(z)$ .
- Rotational errors are the rotations of one carriage about any axis. Therefore there are nine rotational errors,  $\varepsilon_x(x)$ ,  $\varepsilon_y(x)$ ,  $\varepsilon_z(x)$ ,  $\varepsilon_x(y)$ ,  $\varepsilon_y(y)$ ,  $\varepsilon_z(y)$ ,  $\varepsilon_x(z)$ ,  $\varepsilon_y(z)$ , and  $\varepsilon_z(z)$ .
- Squareness errors are the expected values of the deviations of the right angles between the three axes. Three axes form three pairs of axes, so there are three squareness errors,  $S_{xy}$ ,  $S_{xz}$ , and  $S_{yz}$ .

For each of the positional errors, straightness errors and rotational errors, the variable in the brackets indicates the nominal position of the carriage along the moving axis, and the subscript indicates the direction of the error. For each of the squareness errors, the subscript indicates the pair of axes between which the squareness error is defined.

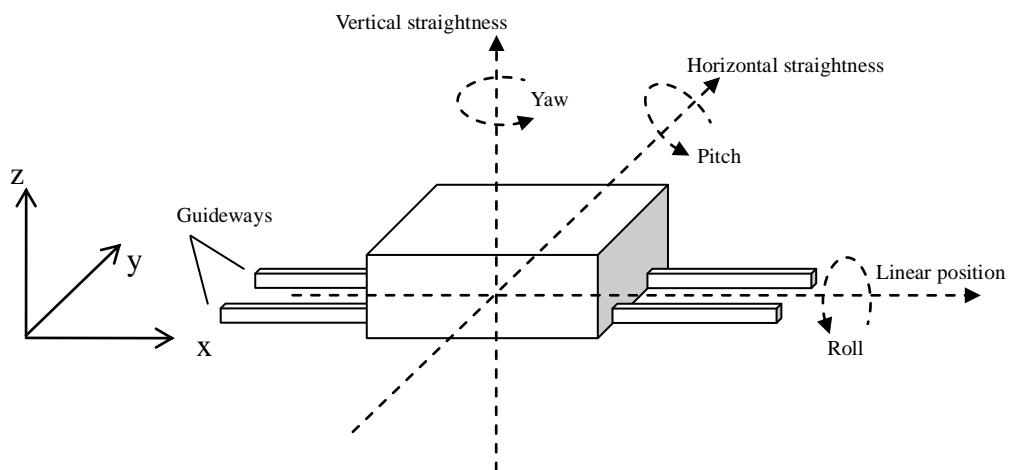


Figure 4.10 Six geometric errors along X axis

With proper equipments, each parametric error can be measured for the measuring volume of a CMM. Repeated measurements need to be made so that a distribution can be obtained for each parametric component at each measured step. The distributions of error data are approximated with normal distributions. Both the mean values and standard deviations are stored in a table for each of the parametric errors and are indexed by the positions where they are measured. When a point is sampled in the AVCMM, its coordinates are used to retrieve the mean value and standard deviation of each parametric error from the nearest interval in each corresponding table. For each parametric error, the retrieved mean value and standard deviation are then passed to a random number generator to form a normal distribution and draw a value randomly from the distribution. Once all parametric errors are sampled, the volumetric error for that point can be calculated from the 21 parametric errors. The calculation differs for different kinematic configurations of CMMs. For the five types of CMMs illustrated in Figure 4.1 and Figure 4.3, four types of kinematic models are derived, as fixed bridge (Type b) and fixed horizontal arm (Type d) CMMs actually share the same kinematic model. The method and equations for synthesizing the 21 parametric error components into the volumetric error have been described and published by other researchers (Pahk et al., 1998).

- Moving bridge, cantilever, gantry and L-shaped CMMs (Type a). If  $(X, Y, Z)$  and  $(X_a, Y_a, Z_a)$  denote the nominal and actual coordinates for a Type a CMM respectively and  $(X_p, Y_p, Z_p)$  are the probe offset coordinates, then their relationship can be described by

$$\begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = T_y \left( T_x \left( T_z \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} + L_z \right) + L_x \right) + L_y \quad (4.5)$$

where  $T_x$ ,  $T_y$ ,  $T_z$  are rotational matrices and  $L_x$ ,  $L_y$ ,  $L_z$  are translational vectors,



along each axis respectively. When the rotation angle is small in the case of CMMs, the rotational matrices can be approximated as below:

$$T_x = \begin{bmatrix} 1 & -\varepsilon_z(X) & \varepsilon_y(X) \\ \varepsilon_z(X) & 1 & -\varepsilon_x(X) \\ -\varepsilon_y(X) & \varepsilon_x(X) & 1 \end{bmatrix} \quad (4.6)$$

$$T_y = \begin{bmatrix} 1 & -\varepsilon_z(Y) & \varepsilon_y(Y) \\ \varepsilon_z(Y) & 1 & -\varepsilon_x(Y) \\ -\varepsilon_y(Y) & \varepsilon_x(Y) & 1 \end{bmatrix} \quad (4.7)$$

$$T_z = \begin{bmatrix} 1 & -\varepsilon_z(Z) & \varepsilon_y(Z) \\ \varepsilon_z(Z) & 1 & -\varepsilon_x(Z) \\ -\varepsilon_y(Z) & \varepsilon_x(Z) & 1 \end{bmatrix} \quad (4.8)$$

$$L_x = \begin{bmatrix} X + \delta_x(X) \\ \delta_y(X) - S_{xy}X \\ \delta_z(X) \end{bmatrix} \quad (4.9)$$

$$L_y = \begin{bmatrix} \delta_x(Y) \\ Y + \delta_y(Y) \\ \delta_z(Y) \end{bmatrix} \quad (4.10)$$

$$L_z = \begin{bmatrix} \delta_x(Z) - S_{xz}Z \\ \delta_y(Z) - S_{yz}Z \\ Z + \delta_z(Z) \end{bmatrix} \quad (4.11)$$

Therefore, the volumetric error  $(\Delta X, \Delta Y, \Delta Z)$  can be derived:

$$\begin{aligned} \Delta X &= X_a - (X + X_p) \\ &= \delta_x(X) + \delta_x(Y) + \delta_x(Z) + Z \left( -S_{xz} + \varepsilon_y(X) + \varepsilon_y(Y) \right) \\ &\quad - Y_p \left( \varepsilon_z(X) + \varepsilon_z(Y) + \varepsilon_z(Z) \right) \\ &\quad + Z_p \left( \varepsilon_y(X) + \varepsilon_y(Y) + \varepsilon_y(Z) \right) \end{aligned} \quad (4.12)$$

$$\begin{aligned}
\Delta Y &= Y_a - (Y + Y_p) \\
&= \delta_y(X) + \delta_y(Y) + \delta_y(Z) + X(-S_{xy} + \varepsilon_z(Y)) \\
&\quad - Z(S_{yz} + \varepsilon_x(Y) + \varepsilon_x(X)) + X_p(\varepsilon_z(X) + \varepsilon_z(Y) + \varepsilon_z(Z)) \\
&\quad - Z_p(\varepsilon_x(X) + \varepsilon_x(Y) + \varepsilon_x(Z))
\end{aligned} \tag{4.13}$$

$$\begin{aligned}
\Delta Z &= Z_a - (Z + Z_p) \\
&= \delta_z(X) + \delta_z(Y) + \delta_z(Z) - X\varepsilon_y(Y) \\
&\quad - X_p(\varepsilon_y(X) + \varepsilon_y(Y) + \varepsilon_y(Z)) \\
&\quad + Y_p(\varepsilon_x(X) + \varepsilon_x(Y) + \varepsilon_x(Z))
\end{aligned} \tag{4.14}$$

- Fixed bridge and fixed horizontal arm CMMs (Type b and d). For all four kinematic models, the rotational matrices  $T_x$ ,  $T_y$  and  $T_z$  are the same as defined in Equations (4.6) – (4.8), but the translational vectors  $L_x$ ,  $L_y$  and  $L_z$  may differ.

$$T_y \begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} + L_y = T_x \left( T_z \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} + L_z \right) + L_x \tag{4.15}$$

$$L_x = \begin{bmatrix} X + \delta_x(X) \\ \delta_y(X) - S_{xy}X \\ \delta_z(X) \end{bmatrix} \tag{4.16}$$

$$L_y = \begin{bmatrix} \delta_x(Y) \\ -Y - \delta_y(Y) \\ \delta_z(Y) \end{bmatrix} \tag{4.17}$$

$$L_z = \begin{bmatrix} \delta_x(Z) - S_{xz}Z \\ \delta_y(Z) - S_{yz}Z \\ Z + \delta_z(Z) \end{bmatrix} \tag{4.18}$$

$$\begin{aligned}
\Delta X &= \delta_x(X) - \delta_x(Y) + \delta_x(Z) + Y\varepsilon_z(Y) + Z(-S_{xz} + \varepsilon_y(X) - \varepsilon_y(Y)) \\
&\quad + Y_p(-\varepsilon_z(X) + \varepsilon_z(Y) - \varepsilon_z(Z)) \\
&\quad + Z_p(\varepsilon_y(X) - \varepsilon_y(Y) + \varepsilon_y(Z))
\end{aligned} \tag{4.19}$$

$$\begin{aligned}
\Delta Y &= \delta_y(X) + \delta_y(Y) + \delta_y(Z) + X(-S_{xy} - \varepsilon_z(Y)) \\
&\quad + Z(-S_{yz} + \varepsilon_x(Y) - \varepsilon_x(X)) \\
&\quad + X_p(\varepsilon_z(X) - \varepsilon_z(Y) + \varepsilon_z(Z)) \\
&\quad + Z_p(-\varepsilon_x(X) + \varepsilon_x(Y) - \varepsilon_x(Z))
\end{aligned} \tag{4.20}$$

$$\begin{aligned}
\Delta Z &= \delta_z(X) - \delta_z(Y) + \delta_z(Z) + X\varepsilon_y(Y) - Y\varepsilon_x(Y) \\
&\quad + X_p(-\varepsilon_y(X) + \varepsilon_y(Y) - \varepsilon_y(Z)) \\
&\quad + Y_p(\varepsilon_x(X) - \varepsilon_x(Y) + \varepsilon_x(Z))
\end{aligned} \tag{4.21}$$

- Moving horizontal arm CMM (Type c).

$$\begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = T_x \left( T_z \left( T_y \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} + L_y \right) + L_z \right) + L_x \tag{4.22}$$

$$L_x = \begin{bmatrix} X + \delta_x(X) \\ \delta_y(X) - S_{xy}X \\ \delta_z(X) \end{bmatrix} \tag{4.23}$$

$$L_y = \begin{bmatrix} \delta_x(Y) \\ Y + \delta_y(Y) \\ \delta_z(Y) \end{bmatrix} \tag{4.24}$$

$$L_z = \begin{bmatrix} \delta_x(Z) - S_{xz}Z \\ \delta_y(Z) - S_{yz}Z \\ Z + \delta_z(Z) \end{bmatrix} \tag{4.25}$$

$$\begin{aligned}
\Delta X &= \delta_x(X) + \delta_x(Y) + \delta_x(Z) - Y(\varepsilon_z(Z) + \varepsilon_z(X)) + Z(-S_{xz} + \varepsilon_y(X)) \\
&\quad - Y_p(\varepsilon_z(X) + \varepsilon_z(Y) + \varepsilon_z(Z)) \\
&\quad + Z_p(\varepsilon_y(X) + \varepsilon_y(Y) + \varepsilon_y(Z))
\end{aligned} \tag{4.26}$$

$$\begin{aligned}
\Delta Y &= \delta_y(X) + \delta_y(Y) + \delta_y(Z) - S_{xy}X - Z(S_{yz} + \varepsilon_x(X)) \\
&\quad + X_p(\varepsilon_z(X) + \varepsilon_z(Y) + \varepsilon_z(Z)) \\
&\quad - Z_p(\varepsilon_x(X) + \varepsilon_x(Y) + \varepsilon_x(Z))
\end{aligned} \tag{4.27}$$

$$\begin{aligned}
\Delta Z &= \delta_z(X) + \delta_z(Y) + \delta_z(Z) + Y(\varepsilon_x(X) + \varepsilon_x(Z)) \\
&\quad - X_p(\varepsilon_y(X) + \varepsilon_y(Y) + \varepsilon_y(Z)) \\
&\quad + Y_p(\varepsilon_x(X) + \varepsilon_x(Y) + \varepsilon_x(Z))
\end{aligned} \tag{4.28}$$

- Column CMM (Type e).

$$T_x \left( T_y \begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} + L_y \right) + L_x = T_z \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} + L_z \tag{4.29}$$

$$L_x = \begin{bmatrix} -X - \delta_x(X) \\ \delta_y(X) - S_{xy}X \\ \delta_z(X) \end{bmatrix} \tag{4.30}$$

$$L_y = \begin{bmatrix} \delta_x(Y) \\ -Y - \delta_y(Y) \\ \delta_z(Y) \end{bmatrix} \tag{4.31}$$

$$L_z = \begin{bmatrix} \delta_x(Z) - S_{xz}Z \\ \delta_y(Z) - S_{yz}Z \\ Z + \delta_z(Z) \end{bmatrix} \tag{4.32}$$

$$\begin{aligned}
\Delta X = & \delta_x(X) - \delta_x(Y) + \delta_x(Z) + Y\varepsilon_z(Y) + Z(-S_{xz} - \varepsilon_y(X) - \varepsilon_y(Y)) \\
& + Y_p(\varepsilon_z(X) + \varepsilon_z(Y) - \varepsilon_z(Z)) \\
& + Z_p(-\varepsilon_y(X) - \varepsilon_y(Y) + \varepsilon_y(Z))
\end{aligned} \tag{4.33}$$

$$\begin{aligned}
\Delta Y = & -\delta_y(X) + \delta_y(Y) + \delta_y(Z) + X(S_{xy} - \varepsilon_z(X) - \varepsilon_z(Y)) \\
& + Z(-S_{yz} + \varepsilon_x(X) + \varepsilon_x(Y)) \\
& + X_p(-\varepsilon_z(X) - \varepsilon_z(Y) + \varepsilon_z(Z)) \\
& + Z_p(\varepsilon_x(X) + \varepsilon_x(Y) - \varepsilon_x(Z))
\end{aligned} \tag{4.34}$$

$$\begin{aligned}
\Delta Z = & -\delta_z(X) - \delta_z(Y) + \delta_z(Z) + X(\varepsilon_y(X) + \varepsilon_y(Y)) - Y\varepsilon_x(Y) \\
& + X_p(\varepsilon_y(X) + \varepsilon_y(Y) - \varepsilon_y(Z)) \\
& + Y_p(-\varepsilon_x(X) - \varepsilon_x(Y) + \varepsilon_x(Z))
\end{aligned} \tag{4.35}$$

#### 4.4.1.2 Geometric error modelling using FEM

In addition to the mathematical models for the calculation of volumetric error from measured parametric error data, we also propose an approach for the acquisition of geometric error by using finite element method (FEM).

FEM, or popularly known by its application FEA (finite element analysis), is a method for solving partial differential equations and integral equations numerically. It is based on the principle of using mesh to split a continuous domain into a set of discrete sub-domains, normally called elements. FEA is widely used in engineering to solve a variety of problems through modelling and simulation. Especially for complex structure and loads, FEA is usually more cost effective than other methods. In this research, we can use FEA to obtain the geometric errors of CMM. The structure, materials and loads of a CMM need to be studied in detail in order to build an accurate model. Then a static

structural analysis can be carried out to obtain the displacement of the probe head relative to the measurement table, at a given location. This displacement should, to a significant extent, represent the combined geometric error at that location. The FEA model needs to be modified to place the probe head at different locations, and the simulation is repeated until a number of locations distributed at certain interval in the CMM measuring volume have been analysed. The resulted data hence form a geometric error map, indexed by the axes locations. When a point is sampled in AVCMM, its associated geometric error can be retrieved according to the contact location.

#### **4.4.2 Probe error modelling**

Probe is one of the major contributors of errors in CMM measurement. For a touch-trigger probe, the probe errors are mainly derived from the pre-travel variation in different probing directions (also known as ‘lobing effects’), due to the kinematic mechanism of touch-trigger probe. Pre-travel causes a small displacement of the stylus tip centre from its free position at the moment of recording, resulting from the stylus shaft deflection caused by the probing force between the stylus tip and the workpiece (Shen & Zhang, 1997). However, the pre-travel variations (lobbing effects) are not directly modelled in the collision detection engine, partly because it will significantly increase the computation and partly because they can be modelled indirectly and equivalently using probe error map. Error map is an effective way to represent and utilize measured probe errors. Probe errors are evaluated by measuring a reference sphere at certain intervals along latitude and longitude directions, as shown in Figure 4.11. The deviations of the distances between measured points and the sphere centre, from the sphere radius, are stored in a 2D matrix which can be indexed by the latitude and longitude of sampling direction. Furthermore, since the errors of a touch-trigger probe are affected by the length of stylus, a probe error map should be constructed for each setup.

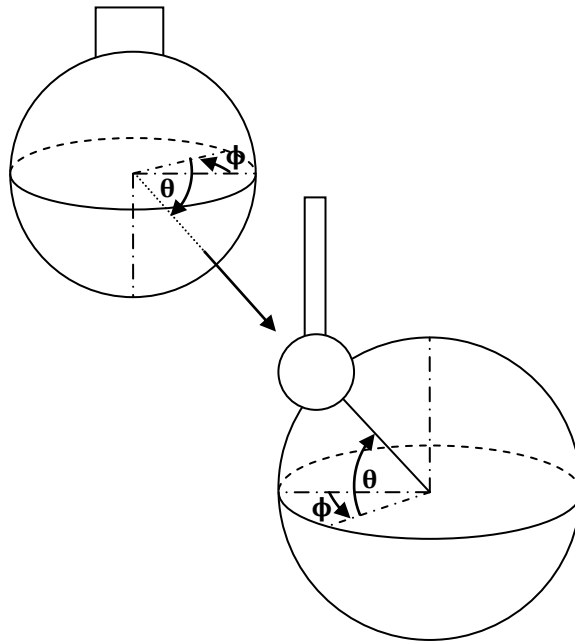


Figure 4.11 Latitude  $\theta$  and longitude  $\phi$

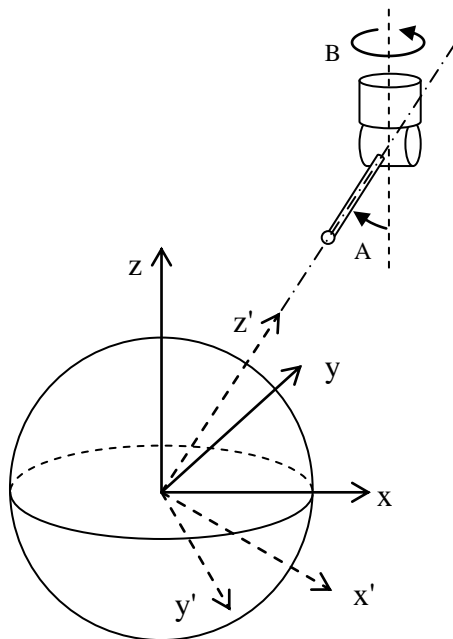


Figure 4.12 Machine and probe coordinate systems

When the probe is tilted and rotated, because the error map is defined with respect to the probe coordinate system, the probing direction in the machine coordinate system must be transformed to the probe coordinate system, as shown in Figure 4.12. This can be done by the operation of a transformation matrix  $T_p$ . Let  $A$  be the titling angle and  $B$  be the rolling angle of the probe, the relationship between the machine coordinate system  $(x, y, z)$  and the probe coordinate system  $(x', y', z')$  can be expressed as

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = T_p \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.36)$$

where

$$T_p = \begin{bmatrix} \cos A \cos B & -\sin B & -\sin A \cos B \\ \cos A \sin B & \cos B & -\sin A \sin B \\ \sin A & 0 & \cos A \end{bmatrix} \quad (4.37)$$

When a point is sampled in AVCMM, the contact position  $P_C = [X_C, Y_C, Z_C]^T$  and the last position before contact  $P_L = [X_L, Y_L, Z_L]^T$  are used to decide the probing direction. The free vector  $v$  pointing from  $P_L$  to  $P_C$  is given by

$$v = \overrightarrow{P_L P_C} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} X_C - X_L \\ Y_C - Y_L \\ Z_C - Z_L \end{bmatrix} \quad (4.38)$$

A unit vector  $\hat{v} = [M_x, M_y, M_z]^T$  representing the probing direction can then be obtained by

$$\hat{v} = \frac{v}{\|v\|} = \frac{v}{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}} \quad (4.39)$$



Since the direction  $\hat{v}$  is defined in the machine coordinate system, it must be transferred to the probe coordinate system by

$$\hat{v}' = \begin{bmatrix} M'_x \\ M'_y \\ M'_z \end{bmatrix} = T_p \hat{v} = T_p \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \quad (4.40)$$

where  $\hat{v}'$  is the unit vector representing probing direction in the probe coordinate system, and  $T_p$  is the transformation matrix defined in Equation (4.37). The latitude  $\theta$  and longitude  $\phi$  of the probing direction can be calculated from the  $\hat{v}'$

$$\theta = \sin^{-1} \left( \frac{M'_z}{\|\hat{v}'\|} \right) = \sin^{-1}(M'_z) \quad (4.41)$$

$$\phi = \tan^{-1} \left( \frac{M'_y}{M'_x} \right) \quad (4.42)$$

The obtained angles  $(\theta, \phi)$  are used to retrieve the measured probe error from the closest interval in the error map. Before it can be added to the measurement result, the retrieved probe error  $E_p(\theta, \phi)$ , which is a deviation along the probing direction in the machine coordinate system, should first be resolved into three components along the three axes of the machine coordinate system according to the probing direction  $\hat{v}$ :

$$\begin{bmatrix} \varepsilon_{p,x} \\ \varepsilon_{p,y} \\ \varepsilon_{p,z} \end{bmatrix} = E_p(\theta, \phi) \hat{v} \quad (4.43)$$

where  $(\varepsilon_{p,x}, \varepsilon_{p,y}, \varepsilon_{p,z})$  are the resolved probe error components.

### 4.4.3 Error arising from random effects

It should be noted that the error mapping for both machine geometric error and probe error mostly addresses only the errors caused by systematic effects. Although the measurement data used to compose the error map are more or less already affected by random effects, the error arising from the random effects normally cannot be quantified yet because the measurements of error map data are usually carried out only once or repeated for very limited times due to the long time consumption of the process. Instead, the error arising from random effects can be estimated by repeatedly measuring a set of points in the measuring volume and comparing the variations of the measured positions in the repeated observations. Based on large number of measurements, the probability distribution of the error arising from random effects can be obtained by statistical algorithms and can then be used to construct the random part of error in AVCMM.

## 4.5 Data processing and surface fitting

The coordinates of probed points, which include simulated errors, are passed to the Feature Calculation module to compute the desired parameters of the feature chosen by user. This is accomplished by fitting a geometric surface (or curve) to the probed point data. Commonly used surfaces in coordinate metrology include planes, spheres, cylinders, cones and others, and all of them can be described as a parametric surface  $\mathbf{f}(\mathbf{u}, \mathbf{b})$  in the real coordinate space  $R^3$ , where  $\mathbf{b} = [b_1, \dots, b_n]^T$  are the surface parameters specifying the shape and position of the surface, and  $\mathbf{u} = [u, v]^T$  are footpoint parameters that specify a corresponding point  $\mathbf{f}$  on the surface (Forbes, 2006). Given a probed point set  $\xi = \{\mathbf{x}_i\}_{i=1}^m$ , the task is to find the values of  $\mathbf{b}$  that make the surface  $\mathbf{u} \mapsto \mathbf{f}(\mathbf{u}, \mathbf{b})$  lie ‘closest’ to the data points, although the connotation of ‘closeness’ is regarded in different senses in a variety of fitting methods. There have long been debates over the advantages and disadvantages of each method and there are

no general guidelines on deciding the best choice of fitting strategy for a particular case. In AVCMM, we adopt the popular least squares and orthogonal distance regression (ODR), since there are errors in all three coordinates. In ODR, the best-fit surface is determined by minimizing the sum of squares of the orthogonal distances from the data points to the surface (Forbes, 2006). Let  $d_i(\mathbf{b}) = d(\mathbf{x}_i, \mathbf{b})$  be the distance from point  $\mathbf{x}_i$  to the surface  $\mathbf{f}(\mathbf{u}, \mathbf{b})$ , the fitting is to find parameters  $\mathbf{b} = [b_1, \dots, b_n]^T$  that solve

$$\min_{\mathbf{b}} \sum_{i=1}^m d_i^2(\mathbf{b}) \quad (4.44)$$

where  $m \geq n$ . Usually  $d_i(\mathbf{b})$  has nonlinear relationship to  $\mathbf{b}$  and we use Gauss-Newton algorithm to solve the minimization. Let  $J$  be the associated  $m \times n$  Jacobian matrix defined at an estimate  $\mathbf{b}$  by

$$J_{ij} = \frac{\partial d_i}{\partial b_j} \quad (4.45)$$

Then the estimate is updated by  $\mathbf{b} + \mathbf{p}$  where  $\mathbf{p}$  is the Gauss-Newton step, which solves the matrix equation  $J\mathbf{p} = -\mathbf{d}$  in the least square sense. The system can be solved by finding an orthogonal factorization  $J = QT$ , where  $Q$  is an  $m \times n$  orthogonal matrix and  $T$  is  $n \times n$  upper triangular. The solution  $\mathbf{p}$  can then be found by solving the upper triangular system  $T\mathbf{p} = -Q^T\mathbf{d}$ .

## 4.6 Monte Carlo simulation based uncertainty evaluation

The Monte Carlo simulation in AVCMM is performed by repeating the recorded inspection path and generating sufficient sample of measurement results. It is designed to be able to work in two modes: a) a priori mode; b) adaptive mode. In a priori mode,

the number  $M$  of Monte Carlo trials is specified by user. In the adaptive mode, the simulation performs an increasing number of trials until the results have stabilized in a statistical sense, that is, until twice the standard deviations associated with the numerical results are less than the numerical tolerance associated with the standard uncertainty  $u(y)$  of the measurement results.

Express a numerical value  $z$  in the form  $c \times 10^l$ , where  $c$  is an  $n_{dig}$  decimal digit integer representing the significant decimal digits of the value, and  $l$  is an integer. Then

$$\delta = \frac{10^l}{2} \quad (4.46)$$

is the numerical tolerance associated with  $z$ . According to GUM, usually one or two significant decimal digits are adequate when reporting uncertainty.

The work flow of the adaptive Monte Carlo simulation is presented in the Figure 4.13.

First, set  $n_{dig} = 1$  or  $2$  and the counter for the sets of Monte Carlo trials  $h = 0$ , and determine the number  $M$  of trials in each set by

$$M = \max(J, 10^4) \quad (4.47)$$

and

$$J = \left\lceil \frac{100}{1-p} \right\rceil \quad (4.48)$$

where  $p$  is the level of confidence of the coverage interval that is expected to be

obtained from the Monte Carlo simulation.

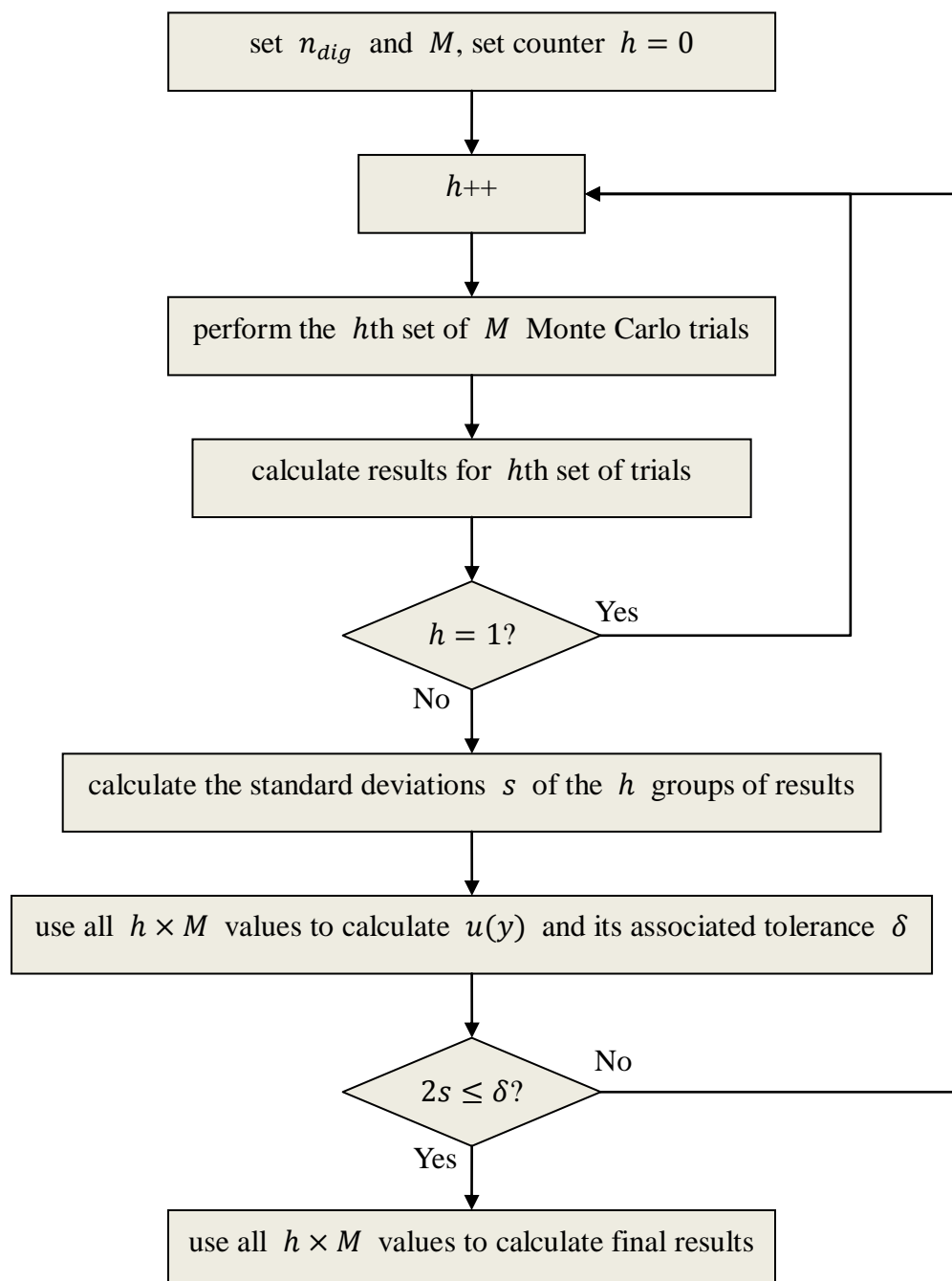


Figure 4.13 Flow chart of the adaptive Monte Carlo procedure

Before each set of Monte Carlo trials, increase  $h$  by one, where  $h$  denotes the serial number of the set in the sequence. Then perform  $M$  Monte Carlo trials by repeating the recorded inspection path  $M$  times and obtain  $M$  values, based on which the estimate  $y^{(h)}$  of  $Y$  and its associated standard uncertainty  $u(y^{(h)})$ , and the endpoints  $y_{low}^{(h)}$  and  $y_{high}^{(h)}$  of the 100p% coverage interval are calculated, for the  $h$ th set.

Then if  $h = 1$ , increase it by one and repeat the trials to make sure at least two sets of trials are performed. Thus the standard deviations  $s$  can be calculated for the results of interest obtained from the  $h$  sets of trials so far:

$$s_a^2 = \frac{1}{h(h-1)} \sum_{r=1}^h (a^{(r)} - a)^2 \quad (4.49)$$

and

$$a = \frac{1}{h} \sum_{r=1}^h a^{(r)} \quad (4.50)$$

where  $a$  represents any of the results of interest, i.e.  $y$ ,  $u(y)$ ,  $y_{low}$  and  $y_{high}$ .

Calculate the  $u(y)$  from all  $h \times M$  values available so far and determine the numerical tolerance  $\delta$ . If all of  $2s_y$ ,  $2s_{u(y)}$ ,  $2s_{y_{low}}$  and  $2s_{y_{high}}$  are less than  $\delta$ , the overall simulation is considered having stabilized, and final results including  $y$ ,  $u(y)$  and the 100p% coverage interval are computed based on all  $h \times M$  values. Otherwise, the simulation is repeated until it has stabilized.

## **4.7 Summary**

This chapter discussed the design considerations of the AVCMM system in detail. As the basis of the system, the kinematical design of the CMM model and corresponding error simulation are emphasized. The justification and design of a newly developed collision detection engine suitable for the virtual CMM environments is presented. Other related theories and design including for the data fitting and Monte Carlo simulation are also covered.

## **Chapter 5 Implementation of a Prototype of AVCMM**

### **5.1 Introduction**

In this chapter, the development and implementation of a prototype of the proposed AVCMM is introduced.

### **5.2 Development environment and runtime platforms**

The AVCMM prototype is developed mainly on a workstation with 2.16GHz duo core processor, 4GB RAM (3.25GB accessible), GeForce 7900 GS graphics card, and Windows XP SP3 operating system. Other systems used for development include: a PC with 3.4GHz processor, 2GB RAM, GeForce 8600 GT graphics card and Ubuntu 9.10 operating system; a Macintosh with 2GHz duo core processor, 2GB RAM, GMA 950 graphics unit and Mac OS X 10.5 operating system. The general development environment is composed of the MATLAB R2008a and later R2009b packages, Microsoft Visual Studio 2008, and some other supplementary tools that will be mentioned later in this chapter.

The AVCMM system is built upon MATLAB runtime libraries and .Net Framework class library, therefore the final package requires both MATLAB Compiler Runtime (MCR) and Common Language Runtime (CLR) to execute. MathWorks provides MCR for Windows, UNIX and Linux systems. Although the CLR, as Microsoft's implementation of the Common Language Infrastructure (CLI) standard, only runs on Windows operating system, there are other implementations of CLI that can run on a variety of systems, such as Mono.



## 5.3 VRML modelling

As discussed in Chapter 4, the VRML models in the AVCMM system serve as the geometry models and visual representation for both CMMs and workpieces. All models in the prototype are created using the VRML authoring tools V-Realm Builder version 2.0 (for Windows XP SP3 environment), and White\_dune 0.29 (for Mac OS X 10.5 and Ubuntu 9.10 environments).

### 5.3.1 CMM modelling

The physical CMM system modelled in the AVCMM prototype includes a MITUTOYO FN503 CMM with  $500 \times 300 \times 300(mm)$  measurement volume, a Renishaw PH9 probe head with 720 orientations in steps of  $7.5^\circ$  (15 discrete angles from  $0^\circ$  to  $105^\circ$  in the tilting axis and 48 angles from  $-180^\circ$  to  $+180^\circ$  in the rolling axis), and a Renishaw TP2 probe with interchangeable styli, as shown in Figure 5.1.



(a)



(b)

Figure 5.1 (a) FN503 CMM; (b) PH9 probe head and TP2 probe

### 5.3.1.1 Model structure

The MITUTOYO FN503 CMM is a typical moving bridge CMM, and Renishaw PH9 is a two-axis probe head, according to discussions in section 4.2, the overall modelling structure complies with the hierarchy in Figure 5.2.

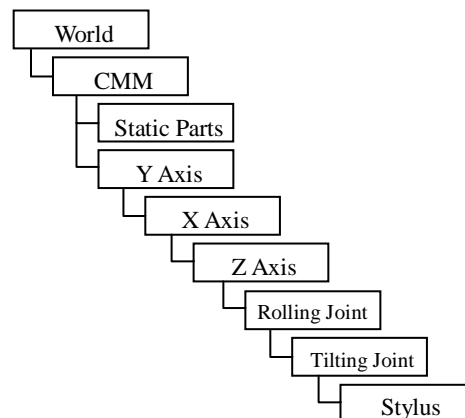


Figure 5.2 VRML model design for AVCMM prototype

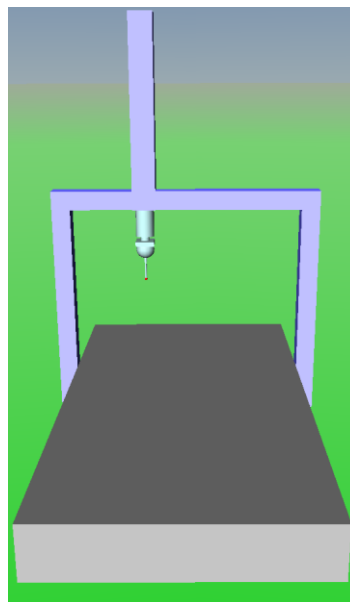


Figure 5.3 A minimal VRML model for CMM

Based on the above design, a minimal VRML model may be created as shown in Figure 5.3. However, to make the representation vivid and improve the sense of reality for user, we chose to extend the minimal model with more details and made it look like its physical counterpart. The final VRML model is shown in Figure 5.4 with a close-up of

the rolled and tilted probe.

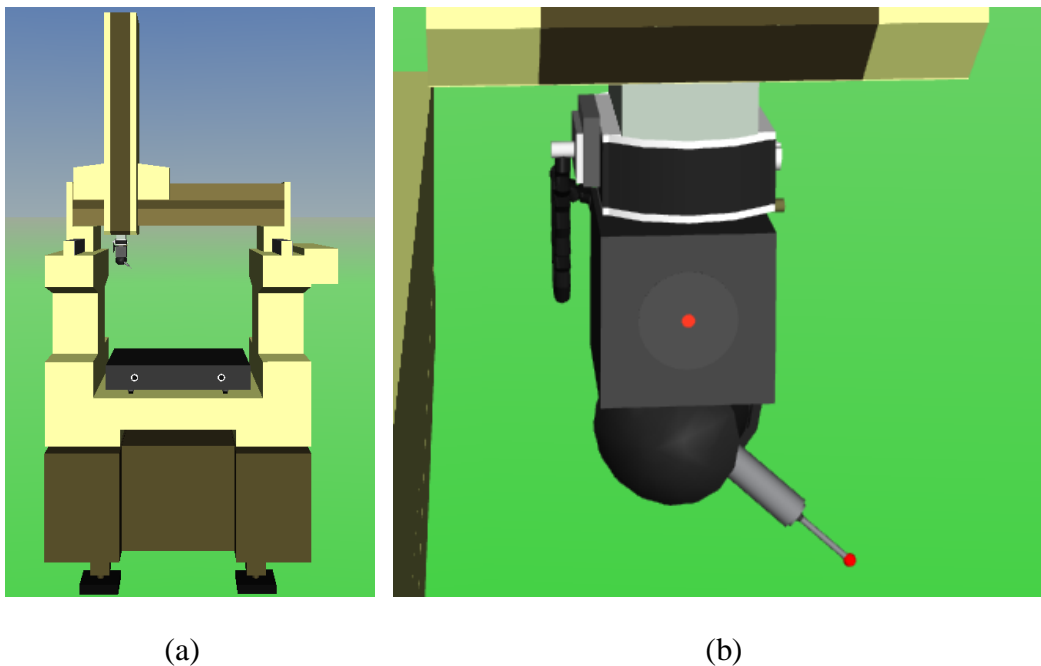


Figure 5.4 An improved VRML model for CMM

### 5.3.1.2 Interface abstraction

In order to interact with the VRML model, an interface has been defined, through which the state of the CMM model can be set and retrieved. The state of the CMM model can be described by the three axes of the CMM plus the two axes of the probe head. In the prototype, the interface consists of following five node fields of five key nodes, in the order of their hierarchy: `Y_Axis.translation`, `X_Axis.translation`, `Z_Axis.translation`, `Roller.rotation`, and `Tilter.rotation`. The former three vectors decide the position of the machine, and only one component of each vector is exposed in the interface, i.e. Y direction component for `Y_Axis.translation`, X direction component for `X_Axis.translation` and Z direction component for `Z_Axis.translation`. The two rotary

nodes decide the orientation of the probe, and each rotary node only rotates about one of its axes, i.e. Roller rotates about its Z axis and Tilter rotates about its X axis.

### 5.3.2 Workpiece modelling

Different workpieces should be able to be loaded into the virtual environment at runtime and can be changed dynamically whenever needed. To meet this requirement, workpieces are not modelled into the virtual CMM world, but in separate individual files. Each workpiece is implemented in one file as a PROTO, which defines a new node type in terms of built-in or other prototyped node types. A PROTO itself is essentially a mini VRML world that can include other nodes and hierarchical structures. Once defined, PROTOs can be instantiated in the virtual world just like the built-in node types. Furthermore, a PROTO may declare its interface so that certain fields may be exposed. This way, each instance of the same PROTO can be customized in certain ways. For example, following code defines a PROTO named 'reference\_ball', which represents a reference ball with 25.4mm diameter:

```
PROTO reference_ball [
  field SFVec3f position 0 0 0
]
{
  DEF bar Transform {
    translation IS position
    children [
      Shape {
        appearance Appearance {
          material Material {
            diffuseColor 0.24 0.24 0.24
          }
        }
      }
    ]
  }

  geometry Cylinder {
    height 0.05
    radius 0.01
  }
}
```

```
    }

}

DEF cone1 Transform {
  translation 0 0.034 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.24 0.24 0.24
        }
      }
    }

    geometry Cone {
      bottomRadius 0.01
      height 0.02
      side TRUE
      bottom TRUE
    }
  ]
}

DEF cone2 Transform {
  translation 0 0.01 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.24 0.24 0.24
        }
      }
    }

    geometry Cone {
      bottomRadius 0.008
      height 0.06
      bottom TRUE
    }
  ]
}
```

```
DEF ball Transform {
  translation 0 0.024 0
  children Shape {
    appearance Appearance {
      material Material {
      }
    }
  }
  geometry Sphere {
    radius 0.0127
  }
}
]
}
]
}
```

In this ‘reference\_ball’ PROTO, the translation field of the uppermost node ‘bar’ is exposed as ‘position’ field of the PROTO, so the position of each instantiated reference ball can be specified individually. For example following code creates an instance of ‘reference\_ball’ PROTO called ‘RefBall1’ and put it at location  $[0.015, 0.01, 0.02]^T$ , as shown in Figure 5.5:

```
DEF RefBall1 reference_ball {
  position 0.015 0.01 0.02
}
```

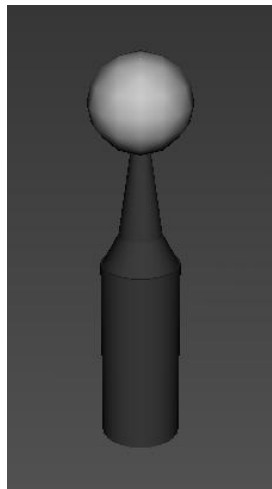


Figure 5.5 An instantiated reference ball

As all workpiece PROTOs are implemented in individual standalone files, they are referenced as EXTERNPROTOs (external prototypes). In the MATLAB code, an EXTERNPROTO can be loaded into a virtual world by using the function `addexternproto`:

```
addexternproto(cmm_world,workpiece_file_1,'workpiece1');
```

## 5.4 VRML parser

The geometrical information contained in the VRML worlds must be extracted and transformed into a more general form in order to be processed by other modules. A VRML parser has been developed to parse all geometries in a VRML world into their mathematical expressions, i.e. systems of equations or inequalities. The parser is developed partially using C# language and .NET framework for their lower I/O accessibility and higher string processing efficiency and partially in MATLAB for its



fast matrix processing and symbolic calculation.

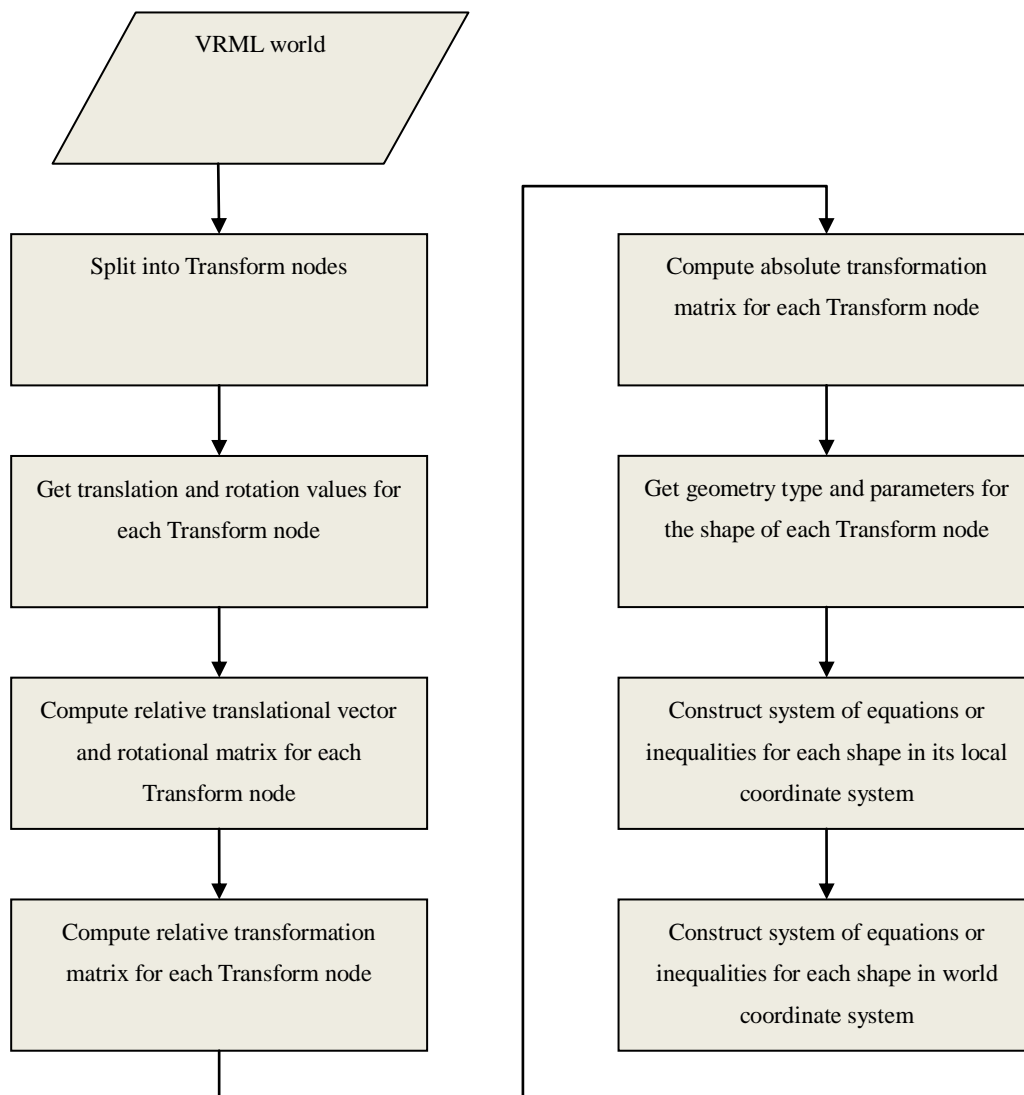


Figure 5.6 Workflow of VRML parser

Figure 5.6 shows the workflow of the VRML parser. The parser takes the whole scene as input, and first splits the world into individual Transform nodes, by defining proper delimiters and using string parsing techniques. Since the translation and rotation in VRML are relative to the Transform node’s parent Transform node, the absolute transformation of each Transform node with respect to the VRML world coordinate system must be computed. If the Transform node is a root Transform node, then its

absolute transformation is simply the combination of its own translation and rotation; otherwise, the absolute transformation should be derived from the translation and rotation of the Transform node and all its ancestor Transform nodes.

Let  $i = 1, \dots, k$  be the depth level of each Transform node in the Transform node hierarchies, for all Transform nodes the parser first get the relative translational vectors  $L_i = [lx_i, ly_i, lz_i]^T$  by simply parsing their translation fields. Then the parser computes the rotational matrices for all Transform nodes from their rotation fields. In VRML, the rotation field of a Transform node consists of four values in the form of an axis and an angle, so that the first three values form a unit vector  $\hat{u}_i = [ux_i, uy_i, uz_i]^T$  describing the axis that the rotation is about, and the fourth value is the angle  $a_i$  by how much the Transform node rotated about the axis. Based on the well established algorithm for rotation about an arbitrary axis in 3 dimensions (Murray, 2005), the relative rotational matrix for the Transform node of  $i$ th level in hierarchy depth is then given by

$$\begin{aligned}
 R_i &= \begin{bmatrix} ux_i^2 + (1 - ux_i^2)c & ux_i uy_i(1 - c) - uz_i s & ux_i uz_i(1 - c) + uy_i s \\ ux_i uy_i(1 - c) + uz_i s & uy_i^2 + (1 - uy_i^2)c & uy_i uz_i(1 - c) - ux_i s \\ ux_i uz_i(1 - c) - uy_i s & uy_i uz_i(1 - c) + ux_i s & uz_i^2 + (1 - uz_i^2)c \end{bmatrix} \\
 &= \begin{bmatrix} r11_i & r12_i & r13_i \\ r21_i & r22_i & r23_i \\ r31_i & r32_i & r33_i \end{bmatrix}
 \end{aligned} \tag{5.1}$$

where  $c = \cos a_i$  and  $s = \sin a_i$ .

A combined relative transformation matrix  $T_i$  can then be obtained using homogeneous coordinates:

$$T_i = \begin{bmatrix} r11_i & r12_i & r13_i & lx_i \\ r21_i & r22_i & r23_i & ly_i \\ r31_i & r32_i & r33_i & lz_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

The absolute transformation matrix  $T$  is computed by left multiplying the relative transformation matrix  $T_i$  from the root level ancestor Transform node to the current Transform node:

$$T = T_k \times T_{k-1} \times \dots \times T_1 = \prod_{n=0}^{k-1} T_{k-n} \quad (5.3)$$

The type (sphere, cylinder, cone, etc.) of the direct Shape node child of a Transform node is determined by parsing its ‘geometry’ field, then the corresponding parameters of the shape such as radius, height, Boolean value indicating the existence of certain face, etc., can be retrieved from the geometry node contained in the Shape node’s geometry field. According to the type of the shape, its system of equations or inequalities can be constructed, with respect to its own local coordinate system  $(x', y', z')$ . For example, a solid sphere is described by

$$X'^2 + Y'^2 + Z'^2 - radius^2 \leq 0 \quad (5.4)$$

where  $[X', Y', Z']^T$  denotes a point in the shape’s local coordinate system  $(x', y', z')$ . Let  $T$  be the absolute transformation matrix of the Transform node containing the shape and  $[X, Y, Z]^T$  be the corresponding coordinates in the VRML world coordinate system, then

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = T^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5.5)$$

So the variables  $(X', Y', Z')$  in Inequality (5.4) can be expressed using absolute coordinates  $(X, Y, Z)$  through symbolic calculation in MATLAB. Therefore, the inequality with respect to the world coordinate system can finally be obtained.

## 5.5 Virtual CMM module

The core of the AVCMM system, the Virtual CMM module is implemented using MATLAB. This module consists of several sub-modules that realize different aspects of the virtual CMM functionalities. This section introduces some of the important sub-modules.

### 5.5.1 Workpiece loader

Workpiece Loader is responsible for loading, removing and changing the workpiece models to be measured in AVCMM. When the instruction of loading certain workpiece model is received from the user interface, Workpiece Loader searches the workpiece library for the corresponding VRML file, loads the workpiece PROTO contained in the file using `addexternproto` function, then instantiates the workpiece node at specified position and orientation, and assigns a handle to the workpiece node to make it referencable. After the workpiece node is created, the loader passes the new node to VRML parser so it can be formulized. Finally the display of virtual world is updated with the new workpiece.

Workpiece Loader performs removal of a workpiece model by deleting the instantiated workpiece node and deleting its corresponding formulas. Changing the workpiece is just the combination of removing and loading.

### 5.5.2 Movement control

The movement of the CMM model is driven by either user control or recorded inspection program. There are two parameters affecting the movement at any particular instant: direction and speed. When in manual mode, both parameters can be obtained from a virtual controller, as shown in Figure 5.7.

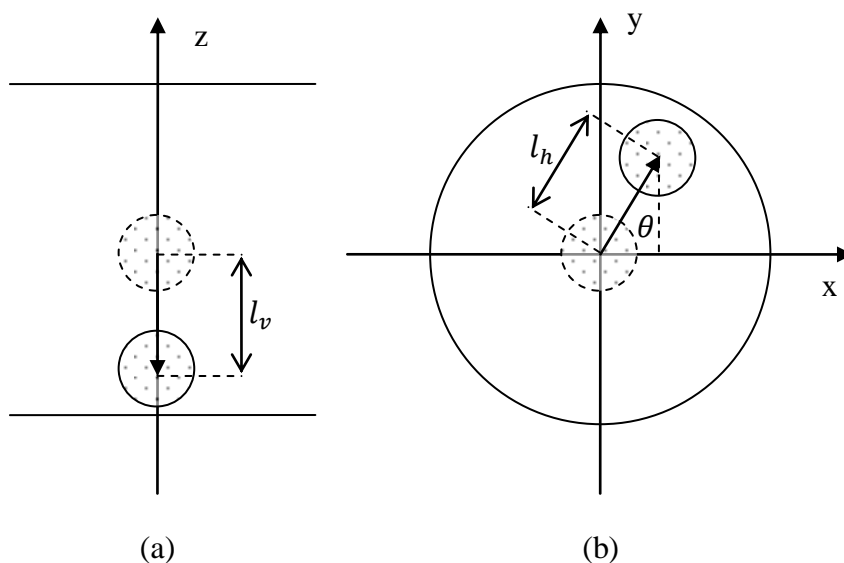


Figure 5.7 Virtual controller axes

Like a typical CMM controller, in the virtual controller the horizontal axes X and Y, and the vertical axis Z, are separated into two groups. User controls the movement of virtual machine by moving (dragging when using mouse as input device) the shaded

controlling circles away from their initial positions. When released, the controlling circles will bounce back to the initial central position. Z axis moves only in two directions, positive and negative, corresponding to lifting or lowering the Z axis carriage. The distance  $l_v$  from initial position to current position decides the vertical movement speed  $v_v$ , i.e. speed of Z axis  $v_z$  by linear relationship  $v_z = v_v = kl_v$  where  $k$  is a coefficient specified according to the desired range of movement speed.

The controlling circle in the XY horizontal plane moves within a bigger circle. The direction  $\theta$  is decided by

$$\theta = \begin{cases} 0, & \text{if } x = 0 \text{ and } y = 0 \\ \frac{\pi}{2}, & \text{if } x = 0 \text{ and } y > 0 \\ \frac{3\pi}{2}, & \text{if } x = 0 \text{ and } y < 0 \\ \tan^{-1}\frac{y}{x}, & \text{if } x > 0 \text{ and } y \geq 0 \\ \tan^{-1}\frac{y}{x} + 2\pi, & \text{if } x > 0 \text{ and } y < 0 \\ \tan^{-1}\frac{y}{x} + \pi, & \text{if } x < 0 \end{cases} \quad (5.6)$$

Similarly the horizontal speed  $v_h$  is given by  $v_h = kl_h$ , where  $l_h = \sqrt{x^2 + y^2}$  is the distance between current position and initial position. The speed components  $v_x$  and  $v_y$  in X and Y axes respectively are given by

$$v_x = v_h \cos \theta \quad (5.7)$$

and

$$v_y = v_h \sin \theta \quad (5.8)$$

The speed of CMM normally has two modes: higher movement speed and lower measurement speed. According to the FN503 configuration, in the AVCMM prototype, the range of movement speed is set as  $0 \leq v_{move} \leq 70$  (mm/s) and measurement speed  $0 \leq v_{measure} \leq 8$  (mm/s) in manual or 'learn mode'. In automatic or 'repeat mode',  $v_{move} = 60$  (mm/s) and  $v_{measure} = 3$  (mm/s).

The obtained speed components in three axes are used to determine the step length in each direction, as mentioned in Chapter 4. The machine position is updated through the three translational parameters in the VRML model interface, i.e. Y direction component of Y\_Axis.translation, X direction component of X\_Axis.translation and Z direction component of Z\_Axis.translation, by adding respective step length (signed) to the old position of each axis.

### 5.5.3 Collision and contact detection

During the inspection, the AVCMM detects any contact or collision between the moving parts (including machine axes and probe), and the stationary parts (including workpiece, table and any optional fixtures), using the collision detection engine introduced in Chapter 4. As the machine moves in two modes, movement mode and measurement mode, the collision detection engine handles the collisions between VRML objects in two different ways. When in measurement mode, any collisions between the probe stylus tip and other objects are classified as 'contacts', therefore the contact position, direction and associated errors are calculated and recorded for further process. In a contact, the machine beeps briefly to notify the user and bounces a short distance away from the probed surface so that the probe is ready for next movement. Collisions between the stationary objects and any moving objects other than the probe stylus tip in measurement mode are classified as 'collisions', as well as any collisions in the movement mode. When a collision occurs, the AVCMM stops the axes from moving

further into the object and sounds a continuous alarm until the collision state is handled. In manual mode, the collision state can be resolved by moving back the axes. In automatic mode, a collision would result a stop of the inspection program.

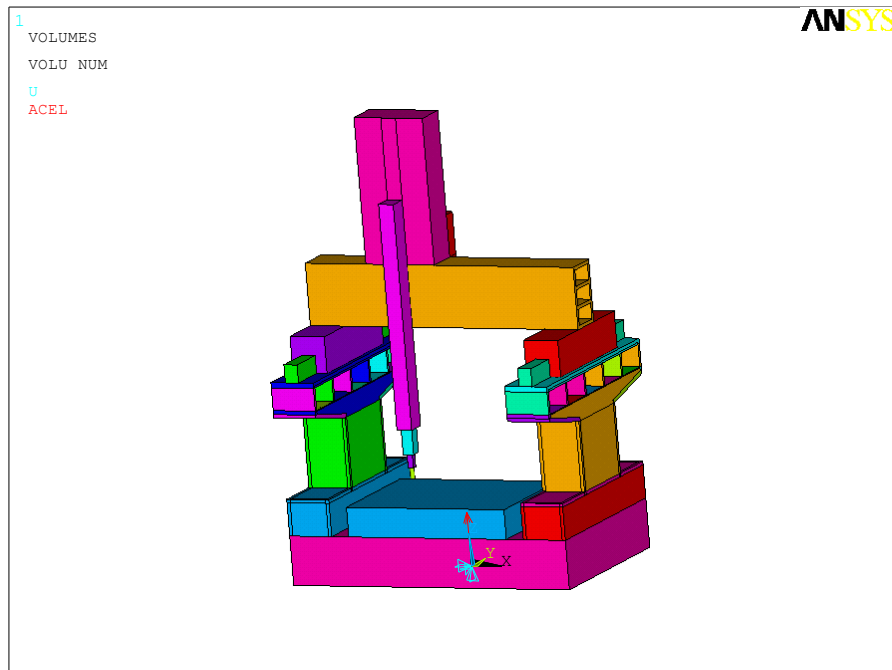
## **5.5.4 Error simulation**

### **5.5.4.1 CMM geometric error simulation**

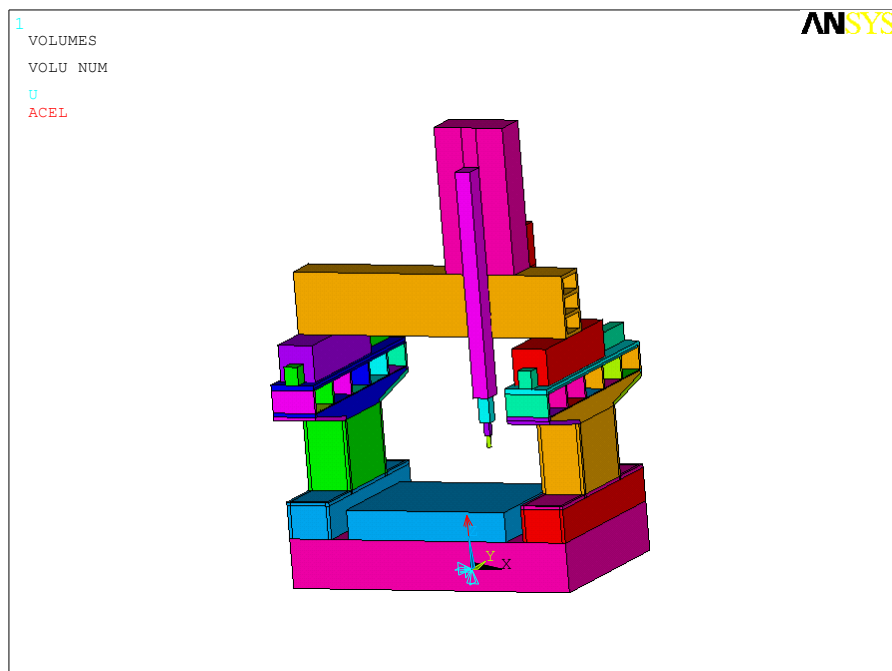
As discussed in Chapter 4, two approaches were proposed to model CMM geometric error. With the first approach, the parametric components of CMM geometric error can be determined, according to the contact location in the measurement volume, from a set of tables containing estimations and standard deviations of parametric components measured at different locations. A set of components are then drawn from their distributions at that location (approximated by normal distributions), and combined into the volumetric error components in three axis directions using the equations derived in Section 4.4.1.1.

In practical applications, in order to obtain accurate simulation results and useful indication of better/worse measurement location, all parametric errors should be measured at different locations in the measurement volume at reasonably small intervals, for each simulated CMM. However, this experimental method is very time consuming and involves the use of several special equipments and gauges. In our experimental AVCMM prototype, we only want to simulate geometric error for demonstration purpose and only validate the simulation results at the level of correct order of magnitude. Therefore we used the alternate FEA approach introduced in 4.4.1.2 to obtain the geometric error map.



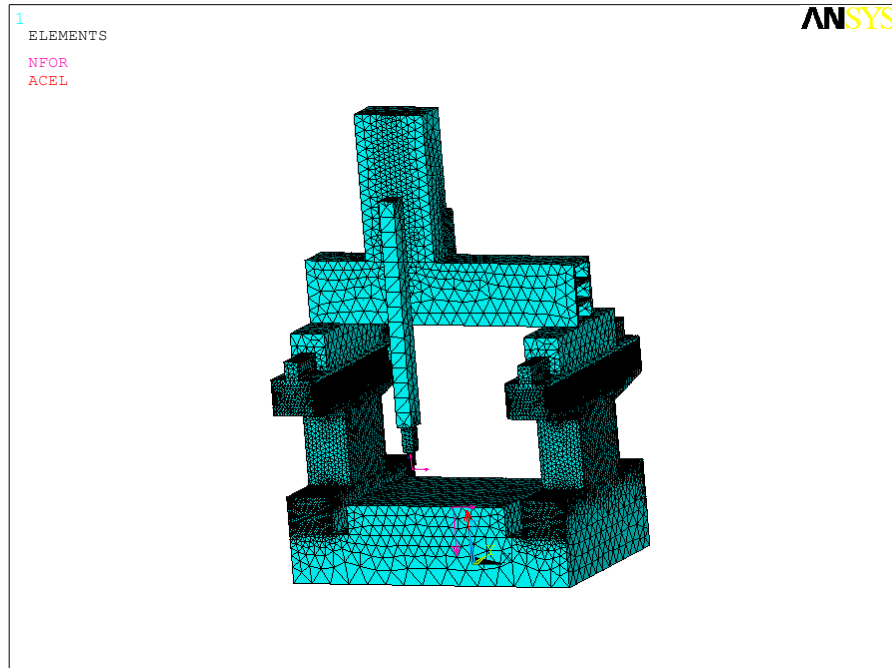


Location (a):  $X = 101, Y = 151, Z = 51$

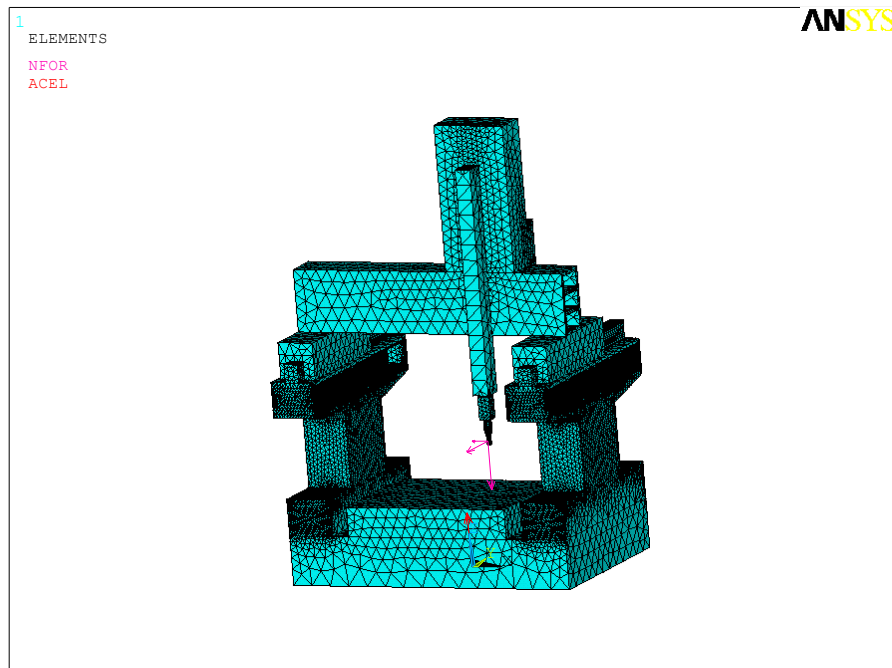


Location (b):  $X = 451, Y = 51, Z = 201$

Figure 5.8 FEA models for two locations

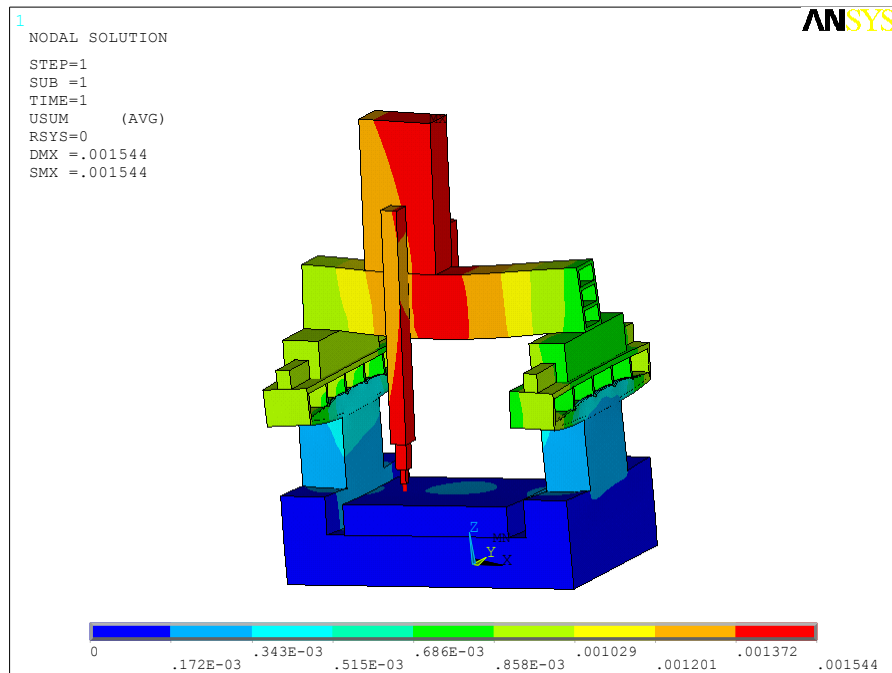


Location (a):  $X = 101, Y = 151, Z = 51$

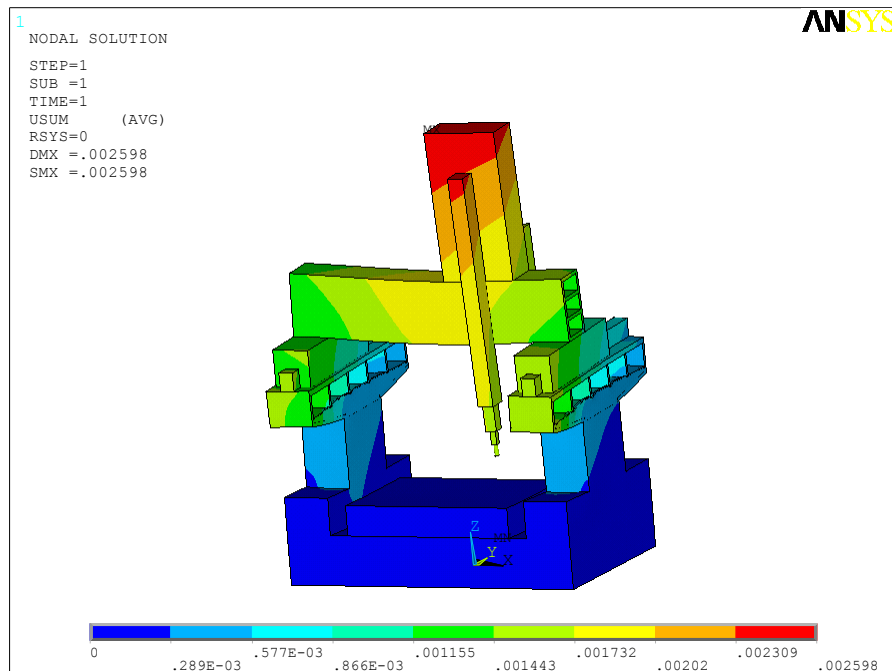


Location (b):  $X = 451, Y = 51, Z = 201$

Figure 5.9 Meshing for FEA at two locations



Location (a):  $X = 101, Y = 151, Z = 51$



Location (b):  $X = 451, Y = 51, Z = 201$

Figure 5.10 FEA results for two locations

The FEA of CMM static structure is performed using ANSYS v12 package. The structure of the CMM is modelled in a way that it reflects the relationships between three axes and all three axes can move along their defined direction. This way the simulation can be performed at multiple locations in the measuring volume at certain interval. As we only require the displacement of the probe head relative to the measuring table at individual positions, global gravity is the only load applied to the model. Table 5.1 shows the properties of the materials used in the FEA and Figure 5.8 – Figure 5.10 demonstrate two sets of FEA models, meshing and results at two different locations as examples.

Material	Elastic Modulus (GPa)	Poisson's Ratio	Density ( $g/cm^3$ )
Aluminum Alloy	79	0.33	2.7
Steel	210	0.3	7.8
Granite	40	0.3	2.6

Table 5.1 Materials used in FEA

In the FEA, we placed the probe head in the measuring volume at interval of 50mm for each axis, resulting  $10 \times 6 \times 6 = 360$  positions. The simulation took roughly 54 hours to complete on a 3.4GHz machine. Figure 5.11 shows the geometric error map formed by the errors at the 360 positions, where blue circles are the nominal positions of the probe head and red stars are the displaced positions with errors enlarged by 5000 times.

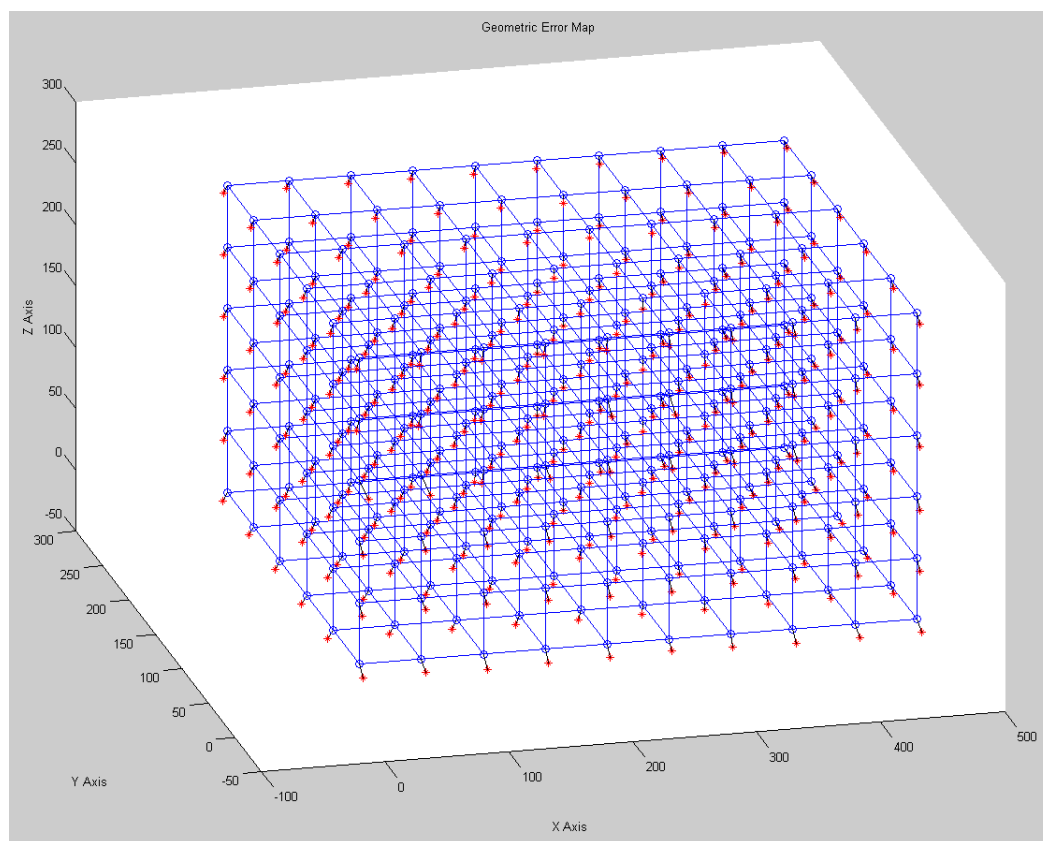


Figure 5.11 Geometric error map

The geometric errors generated from FEA simulations range from  $1.3 \mu m$  to  $3.1 \mu m$ , with mean value of  $1.6 \mu m$  and standard deviation of  $0.27 \mu m$ . The results fall well into the range of max machine error indicated in the manufacturer's specification, which is  $(3 + 3L/1000) \mu m$ , where  $L$  is the measured length in  $mm$ . This compliance suggests that the geometric errors obtained from FEA are plausible at the correct order of magnitude.

#### 5.5.4.2 Probe error simulation

The measurement of probe error was performed by measuring a reference sphere,

following the general method described in Section 4.4.2 (Yang et al., 1996). Error maps for a Renishaw TP2 5-way touch trigger probe have been established based on the measurement results. In this experiment, four factors affecting the probe error were considered: the stylus length  $L$ , the probe tilting angle  $A$ , the probing direction latitude  $\theta$  and longitude  $\phi$ . An error map was created for each combination of  $L$  and  $A$ . For each setup, the datum coordinate system for the reference sphere was aligned parallel to the probe coordinate system so that the  $z'$  axis is parallel to the direction of probe. This way the probing direction angles  $(\theta, \phi)$  equal to the latitude  $\theta$  and longitude  $\phi$  of the probed point on the reference sphere. As shown in Figure 5.12, six circles were measured at interval of  $15^\circ$  along the latitude from  $0^\circ$  to  $75^\circ$ ; plus one point at  $\theta = 90^\circ$ . For each circle, a number of points were measured at certain interval along the longitude. The number of points measured on each circle reduces as  $\theta$  increases, so that for  $\theta = 0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ$  and  $75^\circ$ , respectively 60, 60, 30, 30, 15 and 15 points were measured for each circle, thus a total number of 211 points were measured for each error map.

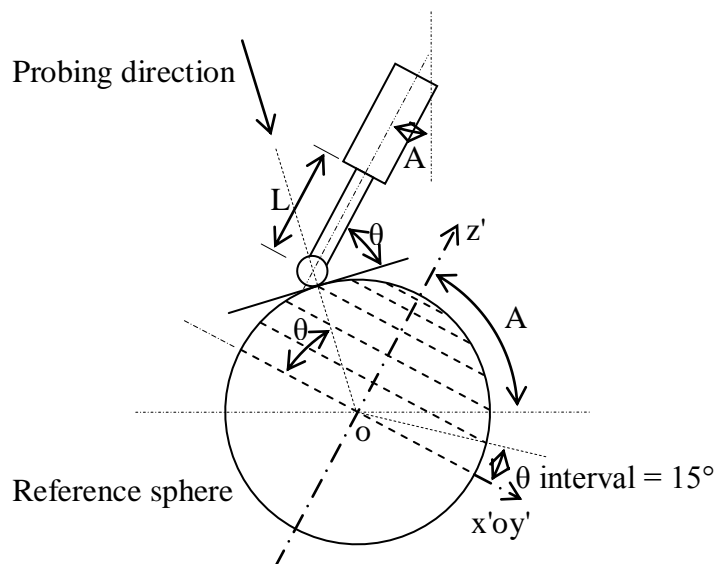


Figure 5.12 Probe error measurement

The measurement was performed in a temperature controlled environment and the ambient condition was kept constant through the experiments. The temperature was  $20 \pm 2^\circ\text{C}$  at 40% relative humidity.

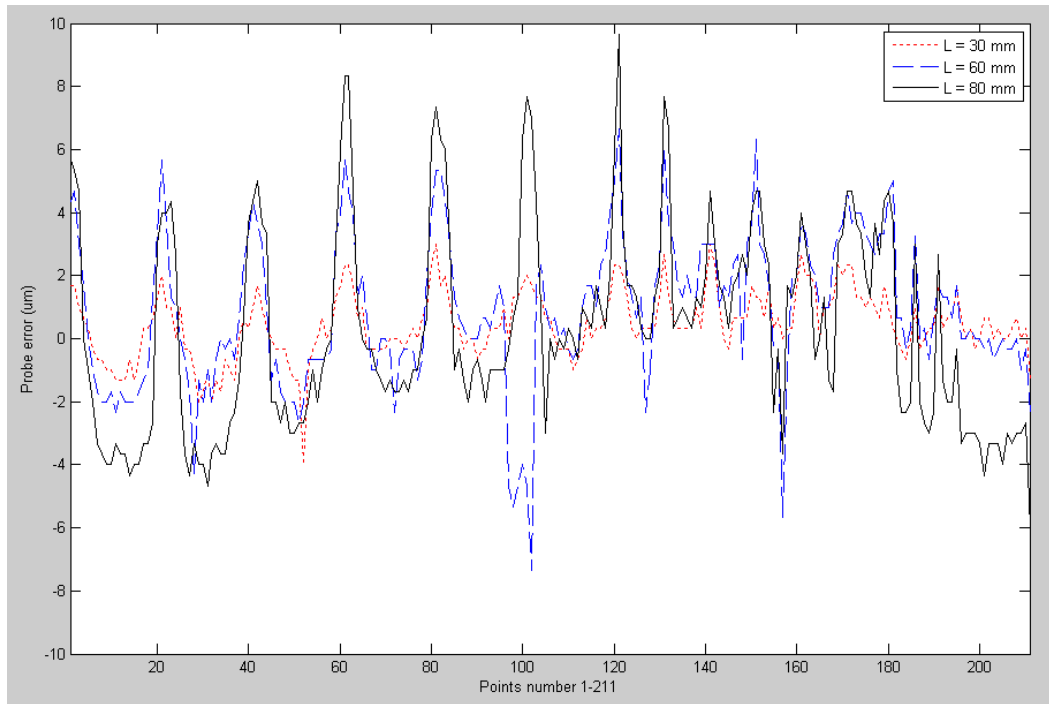


Figure 5.13 Probe error at  $A = 0^\circ$

For all setups, standard stainless steel styli with 3mm diameter were used. The ruby tip has a diameter of 4mm. The measurement speed was at 3 mm/s. The trigger force was set to 7-8 mg. The reference sphere measured in the experiment has a diameter of  $19.9960 \pm 0.0005 \text{ mm}$ .

Seven error maps have been created from the measured data, for styli of three lengths  $L = 30, 60, 80 \text{ mm}$ , at two tilting angle  $A = 0^\circ, 45^\circ$  respectively, plus stylus of  $L = 80 \text{ mm}$  at  $A = 90^\circ$ . Figure 5.13 shows the measured errors at 211 locations for all three styli at  $A = 0^\circ$ . Figure 5.14 shows the measured errors for stylus length

$L = 80 \text{ mm}$  at three tilting angles. It can be clearly observed that the probe error increases with both stylus length and the tilting angle of the probe.

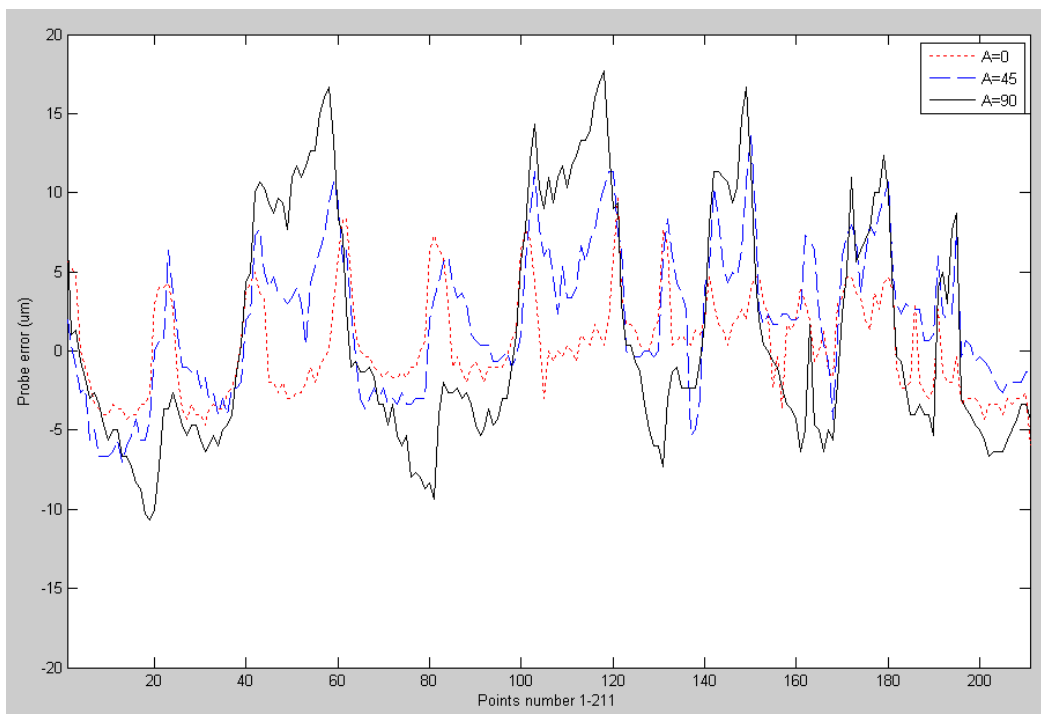


Figure 5.14 Probe error for stylus length  $L=80 \text{ mm}$

Error maps for each setup are plotted into polar coordinate system, as shown in Figure 5.15 – Figure 5.17, for styli length  $L = 30, 60, 90 \text{ mm}$ , respectively. The 3D probe lobing is clearly presented in these plots. It can be observed that the lobing decreases and inflates when latitude  $\theta$  increases. It is also obvious that the probe lobing patterns shift to a direction when the probe is tilted. This is mainly due to the weight of the styli.

From the construction of error maps, it can be verified that for minimum probe error, shorter stylus and smaller tilting angle should be given priority to when condition allows, and the probing direction should be kept parallel to the direction of the probe ( $\theta = 90^\circ$ ) wherever possible.



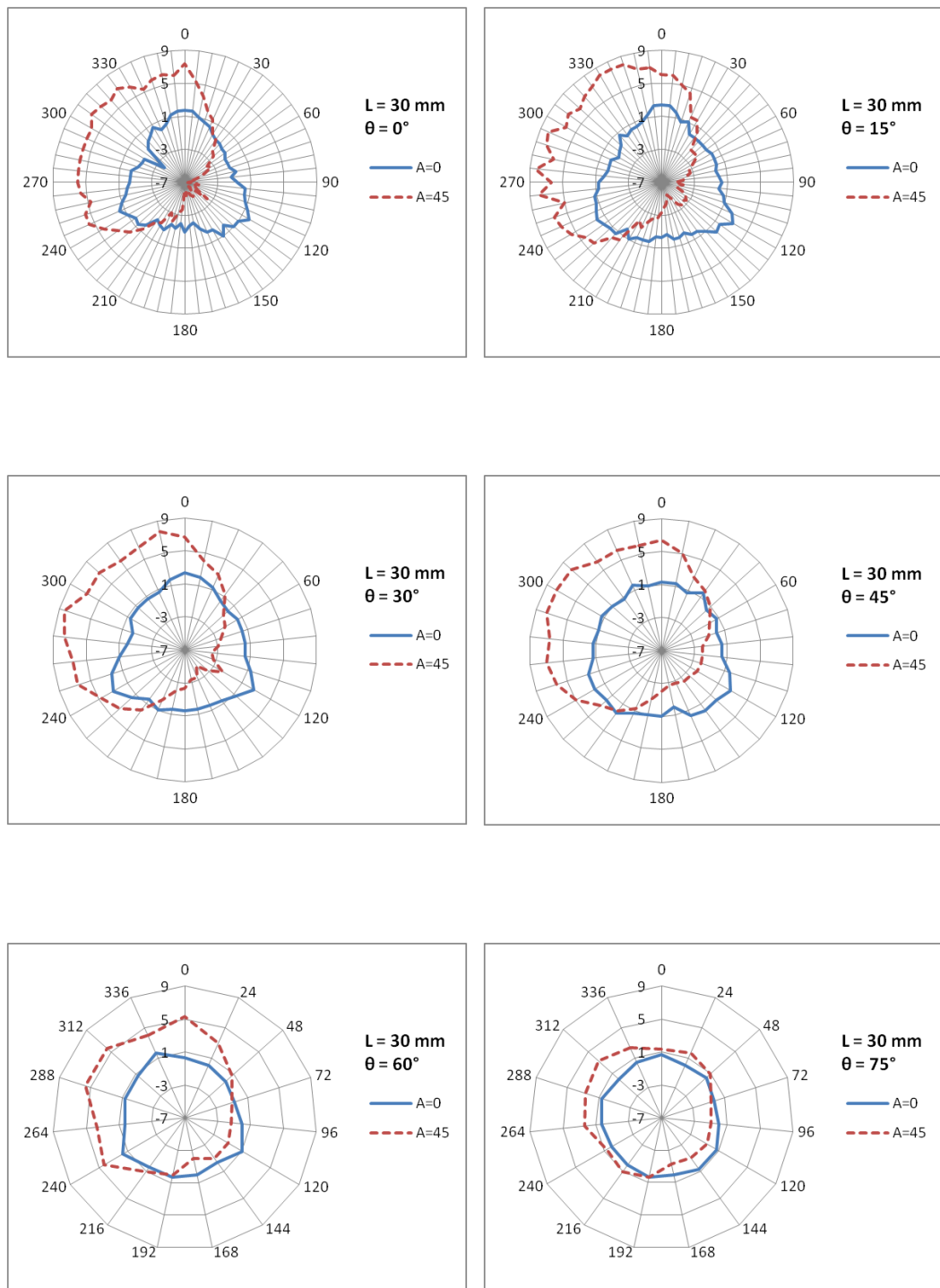


Figure 5.15 Measured probe errors ( $\mu\text{m}$ ) for stylus length  $L = 30\text{ mm}$

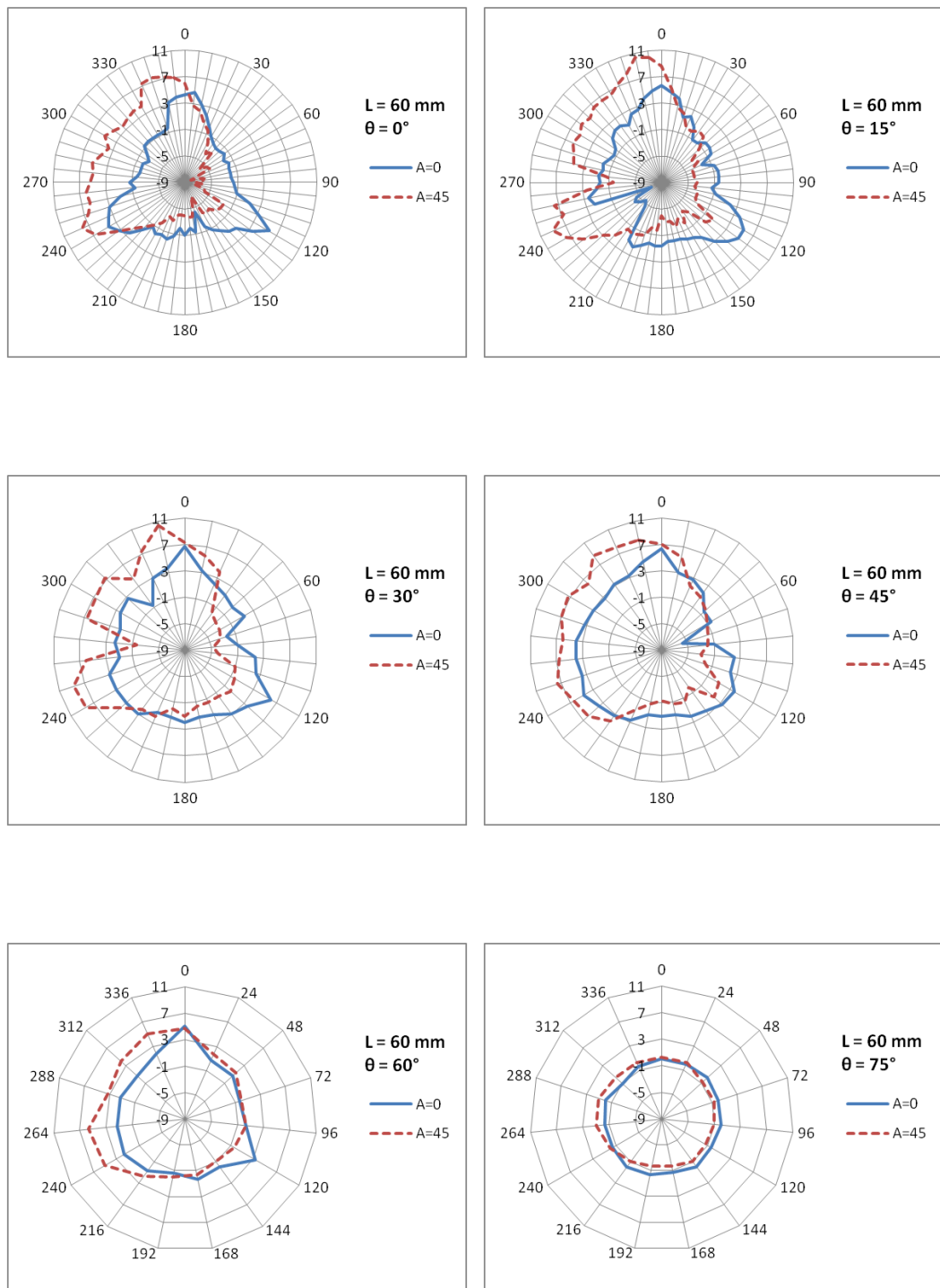


Figure 5.16 Measured probe errors ( $\mu\text{m}$ ) for stylus length  $L = 60 \text{ mm}$

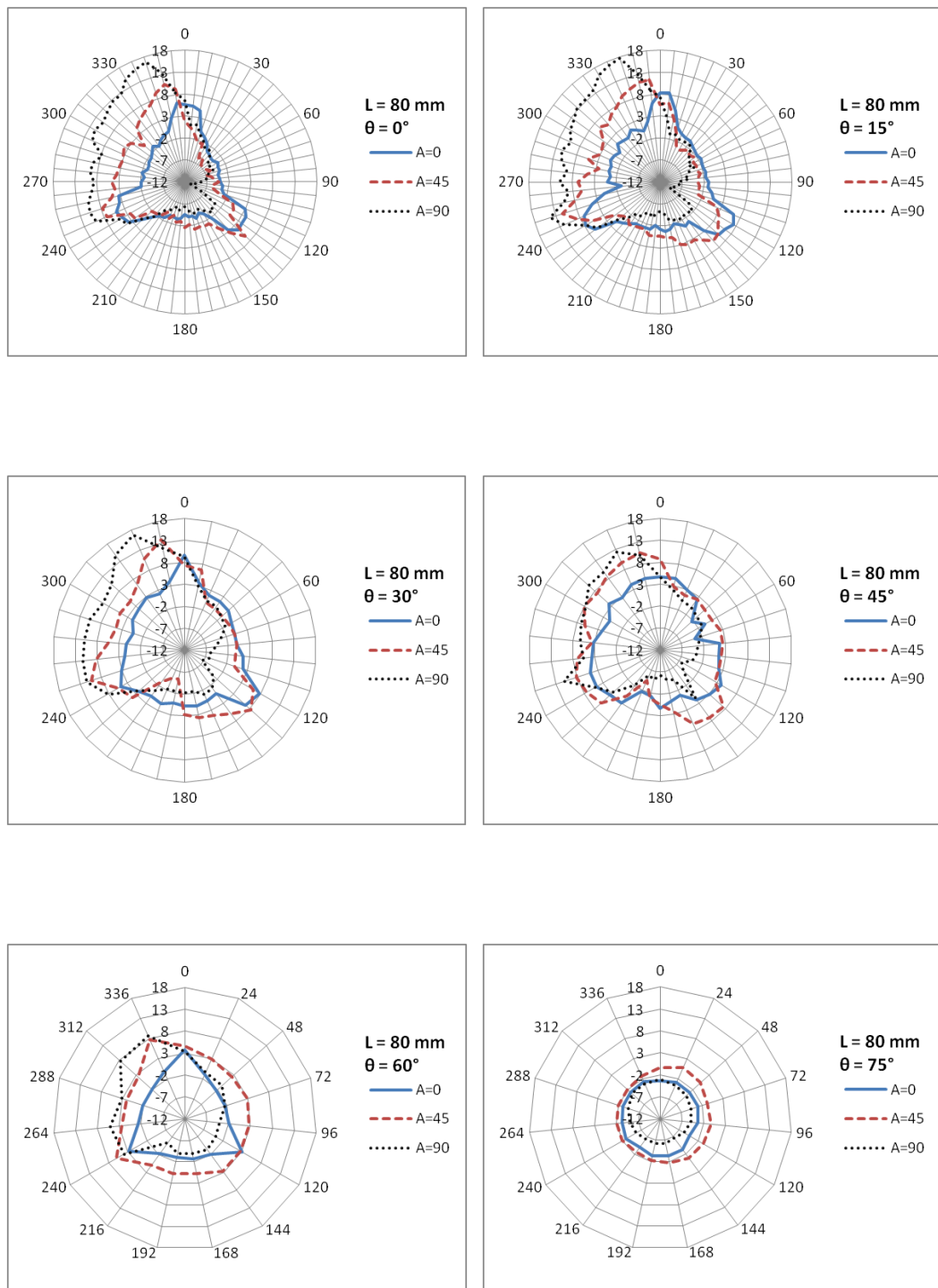


Figure 5.17 Measured probe errors ( $\mu\text{m}$ ) for stylus length  $L = 80$  mm

### 5.5.4.3 Simulation of error arising from random effects

In addition to the error maps which mostly represent the systematic errors of the CMM machine and probe, the error arising from the random effects is also estimated and simulated in AVCMM. 8 points on a ring gauge have been measured in 4 setups at different locations and orientations in the measuring volume, and 5 points on a reference sphere have been measured at 1 location, amounting to 37 points on the two artefacts. For each point, the measurement has been repeated 30 times, therefore a total number of 1110 positions have been observed for all 37 points. For each observation, the variation of the observed position from the average position of the point has been calculated. As shown in Figure 5.18 and Figure 5.19, the variations of measured positions for each individual point are approximately of normal distribution, as well as the variations of measured positions for all 37 points.

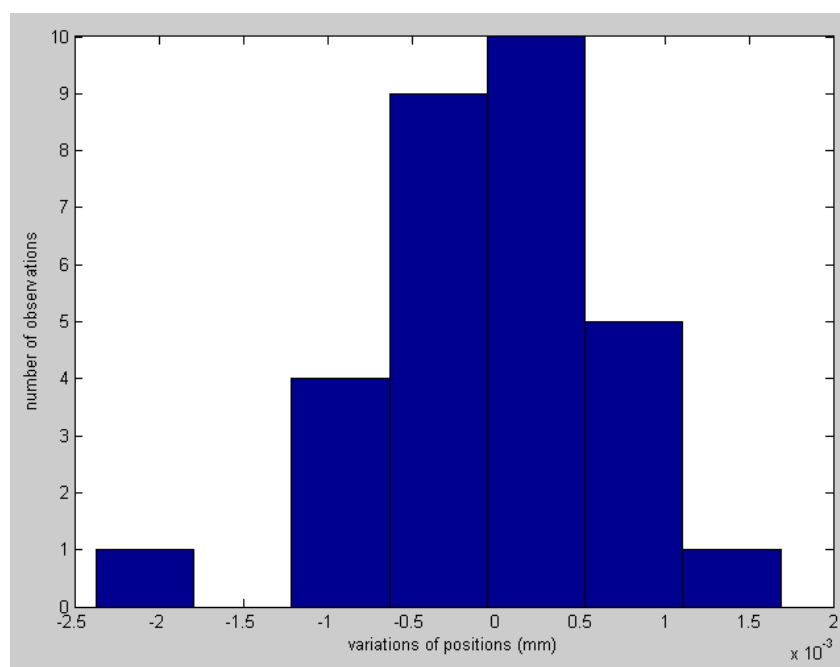


Figure 5.18 Distribution of the variations of positions for one point

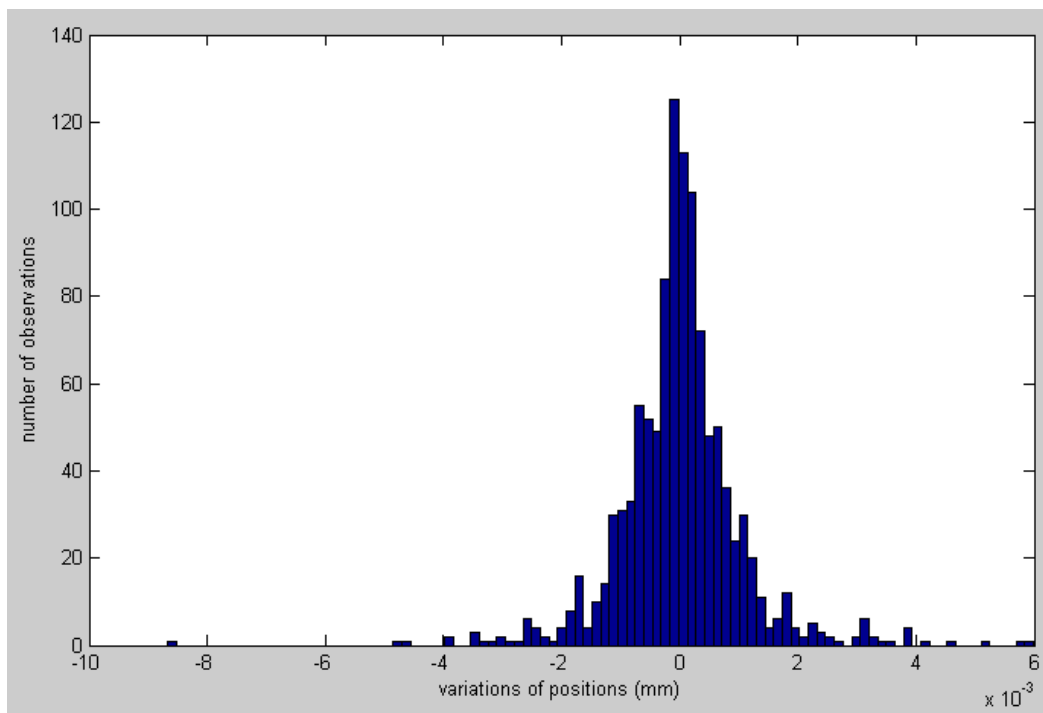


Figure 5.19 Distribution of the variations of positions for all 37 points

The standard deviation of the measured positions of each point has been calculated, and the mean value of all 37 standard deviations has been obtained, which is approximately  $0.9\mu m$ . Meanwhile the standard deviation of the positional variation in all 1110 observations has been calculated and the result is  $1.1\mu m$ . The difference between the two values is merely  $0.2\mu m$ , which may suggest that the random effects are likely to have similar characteristics at different positions in the measuring volume. Therefore for simplicity, we use a uniform value  $0.9\mu m$  as standard deviation to construct random error parts for the whole measuring volume. This uniform standard deviation is passed to the normal distribution generator in AVCMM to randomly draw a random part of error for each sampling contact, where all values drawn as the random error parts would resemble a normal distribution with mean value of  $0\mu m$  and standard deviation of  $0.9\mu m$ . The random error part is then added on top of the error values retrieved from error maps.

## 5.6 Uncertainty evaluation

The uncertainty evaluation module has been implemented to work in two ways. The first one is to automatically repeat the measurement program for specified times to obtain the sample results. The inspection program can be recorded in manual or learning mode, and when repeated, the virtual machine would go through every step to reproduce the whole progress of measurement in each repetition. User may choose to switch on/off the visualization option, which determines whether the uncertainty evaluation module updates the display of VRML world at real time. Though this ‘repetition’ way of simulation is somehow time consuming, it retains all the details in the inspection process, and in the future, other error contributors may be added into the process. As of now, in the prototype only geometric errors and probe errors, together with the random error components are considered, and in a given setup only contact positions and probing directions are required to simulate these errors. Therefore we also implemented a faster simulation method. In the manual mode, the true value of each contact point – that is, the coordinates before errors are added – is recorded, together with the probing direction at each point. Later in the simulation module, these coordinates and directions are repeatedly used to feed the error simulator to generate sets of points with simulated errors. These generated points are then sent to the Feature Calculation module to obtain sample results. Since the inspection progress is not repeated, this method allows the simulation to run much faster. Table 5.2 shows a comparison of the time cost of repetition method with visualization on/off and the faster non-repetition method, on a machine with 2GHz dual core processor, though only one core was used for the simulation. The times recorded in the table are all for inspection of a sphere with a diameter of 40 mm, during which 5 points are probed with a standard ‘1-on-top-4-on-equator’ inspection path. From the comparison we can see that the non-repetition method is more than 100 times faster than the repetition method with visualization on and more than 50 times faster than the repetition method with

visualization off, yet yields same results.

	Repetition (visualization on)	Repetition (visualization off)	Non-repetition
Time	about 20-30 min per 100 trials	about 8-10 min per 100 trials	16-17 min per $10^4$ trials

Table 5.2 Running time comparison for simulation methods

It should be noted that the running time of simulation is dependent on the number of points probed for the feature, as well as the inspection path. Figure 5.20 shows the increase of running time with the number of probed points for the same sphere. The inspection plans are all similar, so that one point is probed at the top, and a few points are probed at several circles at different latitude. From the chart it can be clearly observed that the time cost for all three methods generally increase with the number of probed points. For repetition methods, the time cost is also highly related to the overall length of inspection path, therefore stationary points may be observed on their curves as the total length of inspection path does not necessarily increase with the number of probed points. The times recorded in the figure is only for demonstration purpose and can only serve as a very rough reference of the performance, because in the AVCMM prototype the inspection planning is performed manually, so the overall length of the inspection path is highly related to the operation style of individual operator and therefore the running times are not repeatable. In contrast the running time of non-repetition method is almost proportional to the number of points as it is not affected by the length of inspection path.

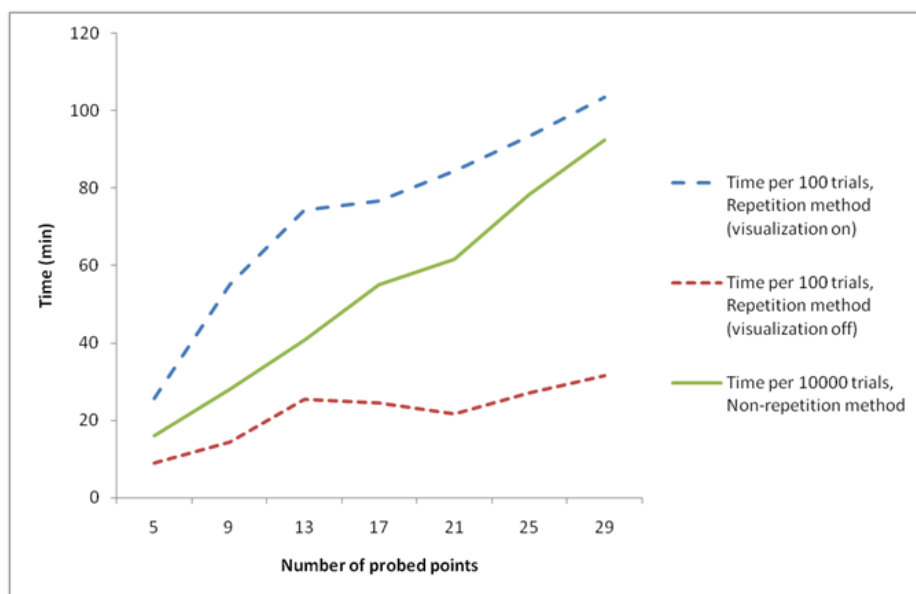


Figure 5.20 Running time for different number of points

As discussed in Section 4.6, the Monte Carlo simulation in AVCMM can work in two modes, a priori mode and adaptive mode. In a priori mode, user can specify the trial number for simulation. In adaptive mode, user can specify the required number of significant digits for standard uncertainty, and the level of confidence for the desired coverage interval. In adaptive mode the simulation continues until all results of interest have stabilized, or the trial number has exceeded the set maximum limit. Normally the estimate  $y$  of the measurand, and its associated standard uncertainty  $u(y)$ , converge much faster than the endpoints  $y_{low}$  and  $y_{high}$  of the coverage interval, with respect to the number of trials. And due to the stochastic nature,  $y_{low}$  and  $y_{high}$  cannot always be decided by the simulation (International Organization for Standardization, 2008). Generally, the number of trials required for results to stabilize is affected by both the required number of significant digits  $n_{dig}$  and the coverage probability  $p$ . For example, if  $p = 0.95$  and  $n_{dig} = 1$ , the results can normally stabilize within  $2 \sim 4 \times 10^4$  trials, but if  $p = 0.99$  and  $n_{dig} = 2$  the simulation may have to run more than  $10^5$  trials.



## 5.7 User interface

### 5.7.1 User interface design

The user interface (UI) of AVCMM was developed with user in mind. The design focuses on the simplicity and efficiency of user operation. As shown in Figure 5.21, the main user interface, which is created in MATLAB, is a graphic-based, event-driven window consisting of a virtual console and an uncertainty evaluation UI. The virtual console is composed of a message window and virtual controllers. The 3D visualization of the virtual CMM world is in a separate window, as shown in Figure 5.22.

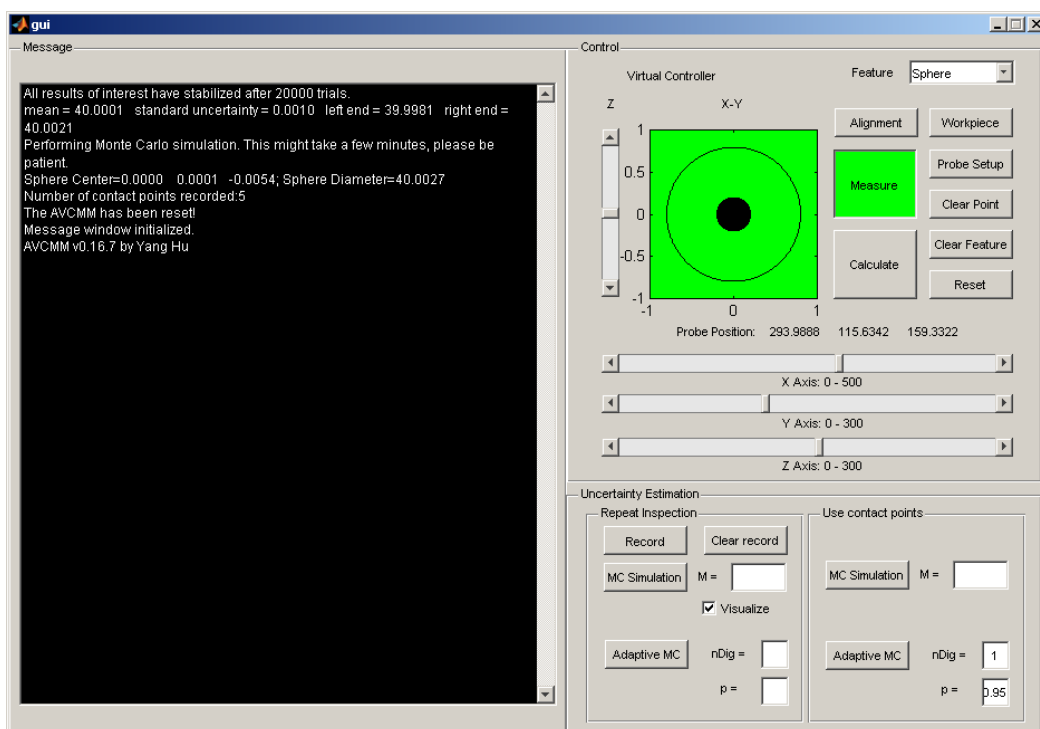


Figure 5.21 Main UI of AVCMM

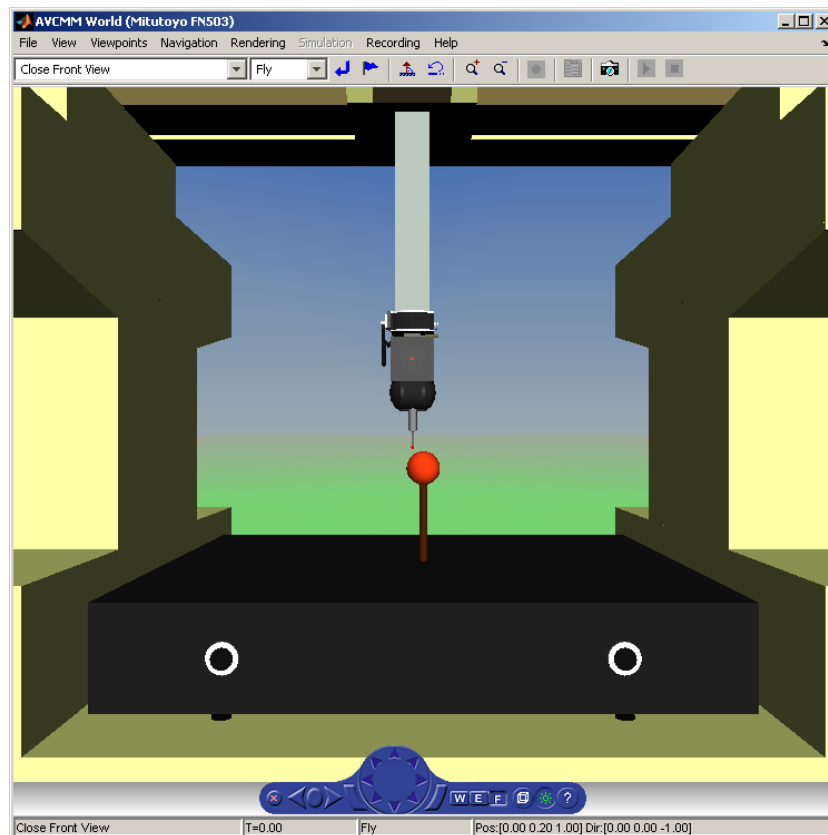


Figure 5.22 3D representation of virtual CMM world

### 5.7.2 3D representation

The visualization of VRML scenes in AVCMM is powered by Orbisnap VRML97 viewer, though it can be replaced by any VRML browser. User can rotate and zoom the scene at any time, leaving no blind point. For convenience, several viewpoints have been predefined at the positions and angles that are suitable for viewing the inspection, as shown in Figure 5.23.

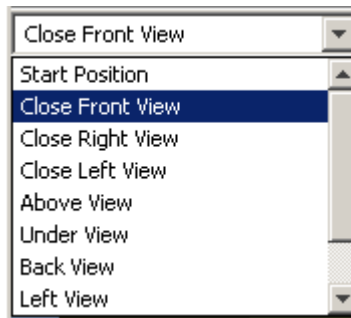


Figure 5.23 Predefined viewpoints

The virtual CMM is animated by updating the VRML scene whenever changes are made to the virtual world, such as moving the axes, loading a workpiece, etc.

### 5.7.3 Virtual Console

#### 5.7.3.1 Message window

The message window reports the status of the AVCMM, the results of measurement and uncertainty evaluation, etc. It is implemented by creating a function that vertically concatenates strings contained in a customized 'Edit Text' box, whenever a new message is arrived.

#### 5.7.3.2 Virtual controllers

The virtual controllers are intended to mimic the behaviour of typical CMM controllers. Being a virtual machine, keyboards and mice are certainly the most commonly available input devices. Although the operation can be simulated by key strokes, for example, using the arrow keys to control the X-Y axes and another pair of keys, '<' and '>' for the Z axis, however this can only provide at most 8 directions in the X-Y plain, and no indication of speed. Therefore, as discussed in Section 5.5.2, virtual controller has been

introduced to overcome this problem. User can control the movement of the three axes by dragging the virtual controllers towards desired direction, and the further the controllers are dragged away from the initial positions, the faster the axes move. In addition, three sliders are provided to both represent the current position of the machine, and to rapidly jump to a desired location, by clicking or dragging the sliders.

A few buttons and popup menus are arranged beside the controllers, providing access to workpiece library, feature selection, and other CMM operations such as probe setup, measurement mode toggling, machine reset, etc.

#### **5.7.4 Uncertainty evaluation UI**

The uncertainty evaluation UI resides below the virtual controllers. It basically provides options regarding the methods and modes by which the user wants the Monte Carlo simulation to be carried out. Therefore it is divided into two groups of controls, one for repetition method and the other for point reuse method. Each group contains options for either Monte Carlo simulation with manually specified trial number or adaptive Monte Carlo simulation.

### **5.8 Web based user interface**

To enhance the usability and ease of deployment for AVCMM, an extended Web based user interface is under development. The goal is to enable user to view and operate the virtual CMM remotely using a Web browser, without the need of downloading and installing the whole AVCMM package on each client machine. The host machine where AVCMM is running on should have the MATLAB software running a Simulink 3D Animation server session. The Web UI can then connect to the animation server and update the display of virtual world, as shown in Figure 5.24. To view the VRML scene,

the user must have at least one compatible VRML plug-in for Web browser installed on the client machine.

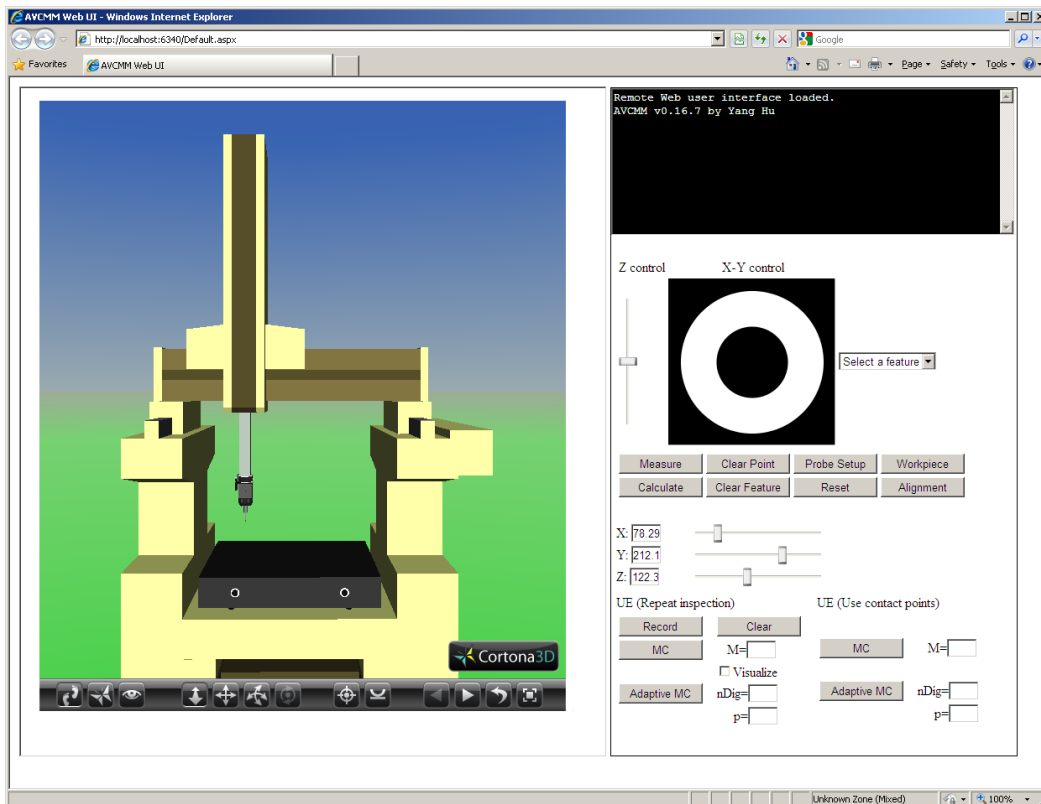


Figure 5.24 Web user interface

The Web UI is basically a reassembly of the native client UI using ASP.NET and AJAX. The ASP.NET Web application accepts user instructions and options from the Web forms and controls, and sends them to the server hosting the AVCMM and Simulink 3D Animation server sessions. The instructions are executed on the server, and any results or changes to the VRML display are collected by the Web application to update the Web UI.

## 5.9 Extension interface

In order to allow AVCMM package to be invoked or further extended by other applications, an application programming interface (API) has been designed, which can be distributed as a library. The AVCMM library contains several important classes each exposes a few methods. The AVCMM API is summarised in Table 5.3 and Table 5.4.

Class	Method	Description
estimator	repeat	Repeat the inspection program for specified times
	repeat_ada	Adaptively repeat the inspection program
	mcs	Monte Carlo simulation using recorded true contact points, with specified trial number
	mcs_ada	Adaptive Monte Carlo simulation using recorded true contact points
errorSimulator	set_probe_error_map	Point to a file containing organized probe error map
	set_parametric_error_profile	Point to files containing organized geometric error components
	get_probe_error	Get the probe error
	get_volumetric_error	Get the volumetric error
	get_combined_error	Get the combined error
VRML-parser	read	Read in a VRML file
	parse	Parse a VRML world into formulas

Table 5.3 AVCMM API (part 1)

Class	Method	Description
vcmm	get_cmm_world	Get the handle to the VRML world of CMM currently loaded and instantiated in the AVCMM
	get_probe_offset	Get the probe offset in a given setup
	get_position	Get the current position of the CMM
	get_co_record	Get the record of probed points in current inspection
	get_feature_flag	Get the type of feature being measured
	set_feature_flag	Set the type of feature being measured
	get_measure_flag	Get the state of measurement mode
	set_measure_flag	Set measurement mode on/off
	get_co_flag	Get a flag indicating whether a collision/contact just occurred
	get_visualize_flag	Get the state of the VRML display
	set_visualize_flag	Indicate whether to update the VRML display
	load_cmm	Create a virtual world from a CMM VRML model
	load_workpiece	Load a workpiece model to the virtual world
	moveTo	Move the machine to a specified position and perform collision detection
	calc	Calculate the parameters of specified feature from the probed points
	update_display	Apply any pending changes from the queue to the virtual world and refresh the display
reset	Reset the AVCMM to initial state	

Table 5.4 AVCMM API (part 2)

## 5.10 Applications of AVCMM

The developed AVCMM package is useful in following applications.

- Inspection path planning. The 3D representation of CMM and inspection process provides an intuitive way of viewing the measurement procedure. Collisions and contacts are detected, and the virtual CMM responds in the same way as real machine. Users can operate the virtual machine to carry out manual path planning in a familiar way as they do on physical machine. The planned path can be quickly evaluated and modified, which makes the planning of an optimal inspection path much easier.
- Uncertainty evaluation and prediction. Either to estimate the uncertainty of a measurement made in real machine, or to predict the uncertainty while planning the inspection path, the AVCMM provides a fast, convenient and reliable evaluation.
- Virtual configuration of CMM. The AVCMM can load models for different types of CMMs and allows different combinations of CMMs and probes. For each setup, AVCMM can provide estimations of inspection uncertainty and therefore helps the users to choose best configuration for their measurement tasks.
- Operator training. The training of CMM operation on physical machine is risky as misoperations may sometimes damage the expensive equipment. Also it is not efficient because there can only be one learner practicing on one machine at a time. AVCMM package features risk free and low cost alternative. The vivid 3D representation and simulated controls let user operate the virtual machine in a similar way to the physical machine. Also the AVCMM has the ability to load different models and configurations for different types of CMM. All these makes the



AVCMM an ideal training simulator of CMM.

- Demonstration of CMM and inspection process. The AVCMM can be used as a tool for composing dynamic and interactive graphic demonstration of CMM machines, or CMM measurements. VRML models of various types of CMM can be loaded into the package, and the desired movement of the machine can be easily programmed and replayed. Furthermore, during the demonstration a viewer can freely zoom and rotation display to observe any detail from any angle.

## **5.11 Summary**

This chapter detailed the implementation of a prototype of the AVCMM system. The modelling method, workflow and algorithms of several important components were presented, as well as the acquisition and organization of error data. The computation time of the Monte Carlo simulation was also analysed and discussed. Finally the UI design and extension interfaces were introduced. We listed some examples of the possible applications of AVCMM.

## Chapter 6 Validation, Results and Discussions

### 6.1 Introduction

Following the implementation of an AVCMM prototype, its validity and performance have been carefully evaluated through a set of experiments, which are presented in this chapter. The general approach involves carrying out a number of measurements of a calibrated artefact on a physical CMM and statistically comparing the observed results with the results predicted by AVCMM. The principles for utilising AVCMM in practice have been experimentally established and demonstrated. The detailed experiment objectives, setup and procedure are introduced and the results are reported and discussed.

### 6.2 Experiment objectives

A serial of experiments have been designed to achieve following objectives:

- To experimentally establish and demonstrate the principles for the utilization of AVCMM in practice;
- To verify the agreement between the results predicted by AVCMM and the results observed in experiments;
- To evaluate the performance of AVCMM system;
- To discuss the limitations and future improvements.

### 6.3 Experiment configurations and setups

The experiments can be divided into two sections: physical inspections and virtual simulations. For the physical inspections, experimental system consists of:

- A MITUTOYO FN503 CMM.
- A Renishaw PH9 probe head.
- A Renishaw TP2 probe.
- A set of stylus components including two M-5000-3648s, one M-5000-3647, one A-5000-7807 and one A-5000-4155.
- A set of fixtures.
- A thermocouple and a platinum resistance thermometer, both have accuracy of 0.1 °C when at around 20 °C.
- A calibrated ring gauge (hardened steel plain setting ring). This ring gauge has been calibrated by an independent procedure in NPL (National Physical Laboratory). The calibration results are reported in Table 6.1. It should be noted that all values have been corrected to 20.0 °C and each expanded uncertainty reported in the table is based on a standard uncertainty multiplied by a coverage factor  $k = 2$ , providing a level of confidence of approximately 95%.
- A reference sphere with nominal diameter of 25.4 mm.

The CMM inspections were carried out in a temperature controlled laboratory. The environmental conditions were kept constant throughout all measurements, with temperature of  $20 \pm 0.3$  °C at 40% relative humidity. The thermocouple was attached to the measurement table and the platinum resistance thermometer was placed near the measurement table. The temperature was monitored by reading from both thermometers.

Ring	Mean diameter at mid-bore at 20.0 °C (mm)	Mean uniformity of diameter (mm)	Mean departure from roundness (1-500 upr) (mm)	
			mid-bore	maximum
124H	38.137 35	0.000 20	0.000 15	0.000 30
	Expanded uncertainty (mm)			
	diameter at mid-bore	uniformity of diameter	departure from roundness	
	$\pm 0.000 09$	$\pm 0.000 13$	$\pm 0.000 08$	

Table 6.1 Calibration results of the ring gauge

The reference sphere was placed at the right hand side of the measurement table when needed, using the middle fixture screw hole. When the same fixture screw hole was required to fix the ring gauge, the reference sphere was removed from the table.

During the experiments, the ring gauge was placed at five locations in the measuring volume, including the centre and four corners, as shown in Figure 6.1. At each location, the ring gauge was placed at two orientations, parallel to the table surface (0 degree) and at an angle of 45 degrees to the table surface, providing 10 combinations of locations and orientations. When the ring gauge was parallel to the surface, it was directly placed on the measurement table; when the ring gauge was at 45 degrees angle to the surface, it

was supported by an inclined surface of a workpiece, therefore the centre of the ring gauge was lifted to height of about 8 cm. For each combination of locations and orientations, two styli were used to measure the bore of the ring gauge, forming total number of 20 setups. One stylus consists of one M-5000-3648 and one A-5000-7807, and has length of 30 mm and ruby tip diameter of 2 mm. The other stylus consists of two M-5000-3648s, one M-5000-3647 and one A-5000-4155, and has length of 60 mm and ruby tip diameter of 5 mm. In each setup, the tilting angle of the probe head (A angle) was set to either 0 or 45 degrees and the rolling angle (B angle) was set to either 0 or 180 degrees, depending on how the ring gauge was orientated. Figure 6.2 demonstrates two examples of the setups.

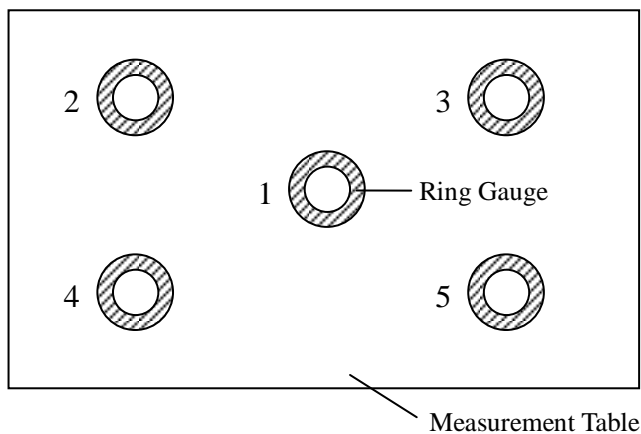


Figure 6.1 Measurement locations (bird's-eye view)

For all measurements, the movement speed of the probe head was set to 60 mm/s default and 70 mm/s maximum, while the measurement speed was set to 3 mm/s default and 8 mm/s maximum.



Figure 6.2 Two example setups

For the other section of the experiments, the virtual simulations, the experimental system consists of three computers simultaneously running three copies of AVCMM. All three computers have Windows XP Pro SP3 32bit, .NET Framework 3.5 and MATLAB r2009b installed. The reason for using three computers was to evaluate the performance of AVCMM on different hardware. The hardware configurations of the three computers are listed in Table 6.2.

Computer	CPU	RAM	Graphics
A	P4 3.4 GHz	2 G	GeForce 8600 GT
B	Core 2 Duo 2 GHz	4 G (2.96 G accessible)	GMA 950
C	Core 2 Duo 2.16 GHz	2 G	GeForce 7900 GS

Table 6.2 Hardware configurations of three computer systems

In the virtual inspections and simulations, a 1:1 VRML model of the ring gauge was created and placed in the measuring volume at locations and orientations similar to the physical inspections. The VRML model of the ring gauge has height of 29 mm, outer

diameter of 70 mm and bore diameter of 38.13735 mm, which is the mean diameter of the physical ring gauge at mid-bore obtained from calibration. No fixture is included in virtual inspections for simplicity. Similarly, for each combination of locations and orientations, stylus has been configured to two different lengths, 30 mm and 60 mm, allowing total 20 setups. The virtual probe head was also tilted and rolled to access the bore of the ring in each setup, with A set to 0 or 45 degrees and B set to 0 or 180 degrees. Figure 6.3 shows two examples of virtual inspection setups.

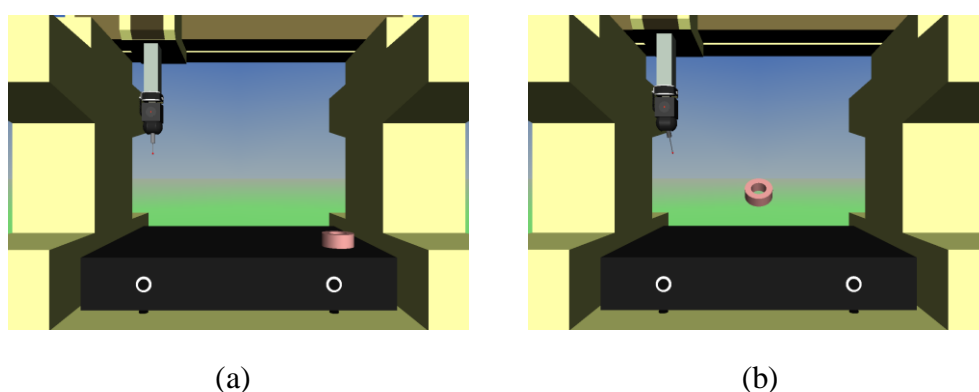


Figure 6.3 Two example setups in virtual inspections

## 6.4 Experiment procedure

The experiments were carried out following the procedure reported below.

- 1) The ring gauge was placed in the same laboratory as the CMM for 1 day before starting the measurements, for it to ‘cool down’ to the same temperature.
- 2) The thermocouple was attached to the surface of the measurement table and the platinum resistance thermometer was placed near the measurement table. The temperature was deemed stabilized if both thermometers had the same reading, which suggests the temperature difference between table and air was less than

0.1 °C. Measurements were only made when the temperature was stable.

Order of setup	Stylus length (mm)	Ring gauge orientation (degree)	Ring gauge location	Probe angle A (degree)	Probe angle B (degree)
1	30	0	1	0	0
2	30	0	2	0	0
3	30	0	3	0	0
4	30	0	4	0	0
5	30	0	5	0	0
6	30	45	1	45	0
7	30	45	2	45	180
8	30	45	3	45	180
9	30	45	4	45	0
10	30	45	5	45	0
11	60	0	1	0	0
12	60	0	2	0	0
13	60	0	3	0	0
14	60	0	4	0	0
15	60	0	5	0	0
16	60	45	1	45	180
17	60	45	2	45	180
18	60	45	3	45	180
19	60	45	4	45	0
20	60	45	5	45	0

Table 6.3 Order of the measurements at 20 setups

- 3) The serial measurements of the ring gauge bore diameter at the 20 setups were made following the order listed in Table 6.3. Temperature was recorded before and after



the measurements at each setup.

The probe was calibrated and recalibrated using the reference sphere whenever the stylus was changed or the angles of the probe were changed. It should be noted that the B angle of probe was not set to 0 or 180 degrees arbitrarily, but due to the fact that the actual measuring volume of the CMM ‘shrinks’ when the probe is tilted. As illustrated in Figure 6.4, when placed at the edge of the measuring volume, the ring gauge may become inaccessible by probe pointing towards the ‘inside’ of the measurement volume. Therefore when  $A = 45^\circ$  we set  $B = 180^\circ$  for location 2 and 3, and  $B = 0^\circ$  for location 4 and 5. As for location 1, which was at the centre of the table, B may be chosen freely, and we used  $0^\circ$  for setup no. 6 and  $180^\circ$  for setup no. 16.

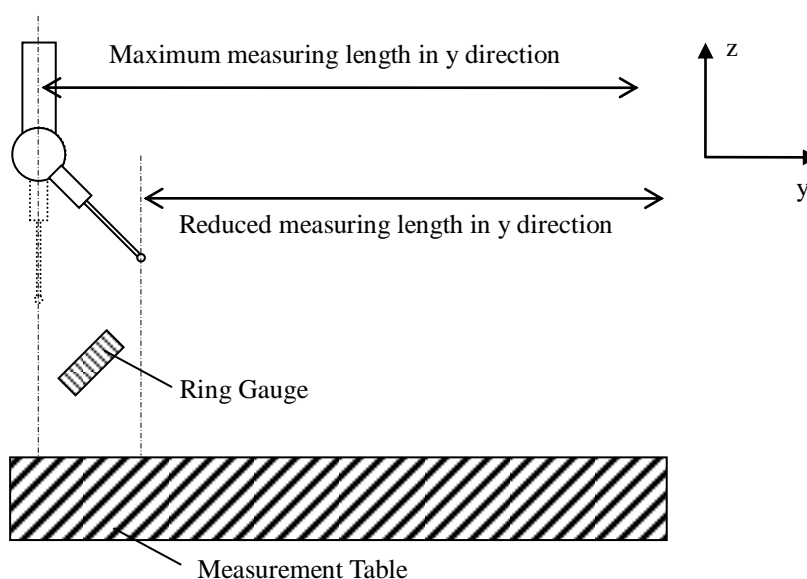


Figure 6.4 Illustration of reduced measuring length with tilted probe

Before the measurements at each setup, alignment was performed to establish part coordinate system  $(x', y', z')$  for the ring gauge, as shown in Figure 6.5. The  $x'y'$  plane was defined by the top surface of the ring gauge, and origin was the centre of

a circle that was the projection of the bore on  $x'y'$  plane. The direction of  $x'$  axis was chosen arbitrarily, but mostly parallel to the  $x$  axis of the machine coordinate system for simplicity.

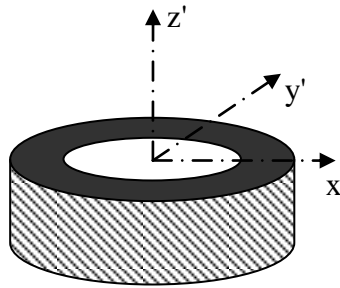


Figure 6.5 Part coordinate system for ring gauge

A same inspection program was used to measure the bore diameter in each setup. The program probes 4 points on the bore surface to fit a circle and calculate the diameter. As shown in Figure 6.6, the 4 points were sampled in counter-clockwise direction near the mid-bore of the ring gauge, at  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  and  $270^\circ$  relative to the  $x'$  direction, respectively. For each setup, the inspection program repeated the same measurement for 30 times, so that 30 measured values of the bore diameter were obtained in each setup, amounting to total 600 values from all 20 setups. All results were recorded by the inspection program in output files for further analysis and comparison.

- 4) Virtual measurements of the VRML ring gauge model were carried out in AVCMM, at 20 setups similar to the CMM inspections. For each setup, stylus length and probe orientations were set to the same configurations as in CMM inspections, using the probe setup window in AVCMM, as shown in Figure 6.7. The alignment process was simplified and we simply rotated the local coordinate system of the ring gauge

node parsed from the VRML file by  $90^\circ$  about the  $x'$  axis and used it as the part coordinate system. This part coordinate system is slightly different from the one we established in physical inspections, because the  $x'y'$  plane is at the mid bore instead of top surface, as shown in Figure 6.8. However since these two planes are parallel with each other, the result of interest – the diameter of the bore – should not be affected.

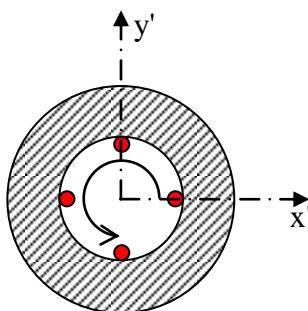


Figure 6.6 Sampling path (bird's-eye view)

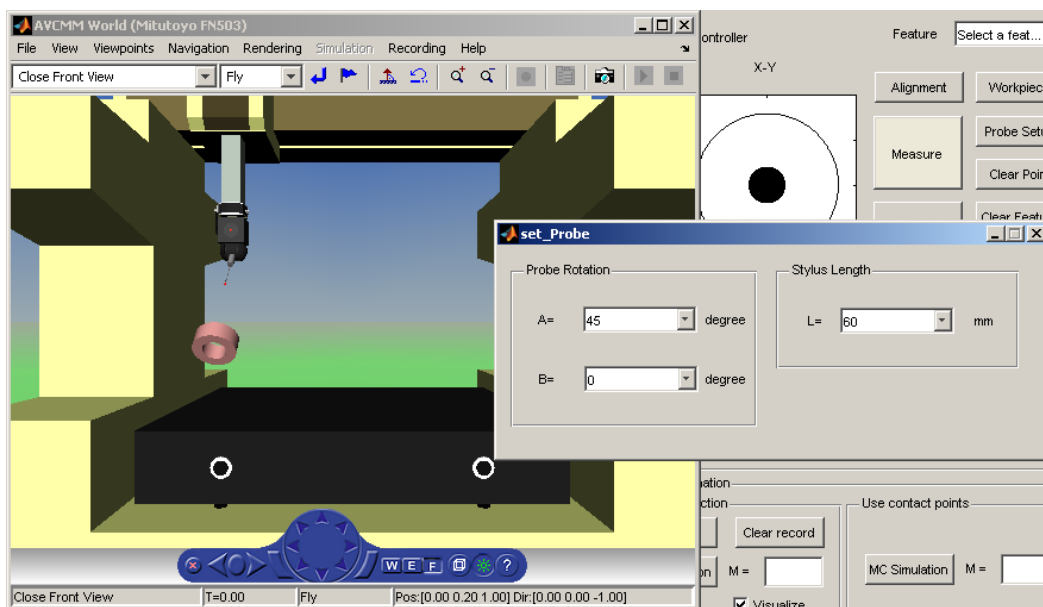


Figure 6.7 Probe setup in AVCCMM

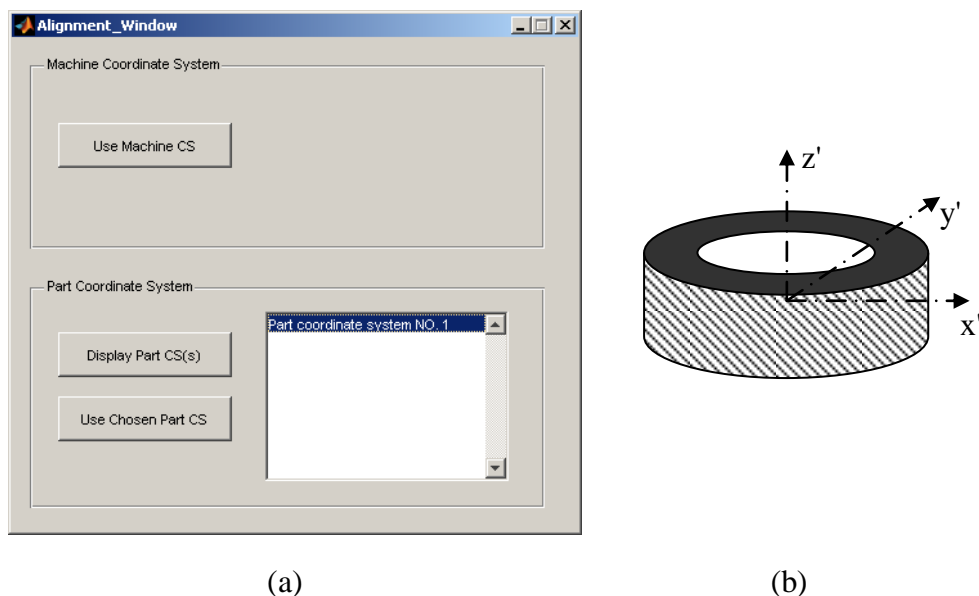


Figure 6.8 (a) Simplified alignment; (b) part coordinate system for ring gauge in AVCMM

At each setup in AVCMM, a virtual measurement was carried out by sampling 4 points at similar positions to physical measurements. Since the  $x'$  direction in AVCMM was not necessarily always the same as in physical measurement, the virtual sampling was handled manually to make sure the probing direction was as close to the probing direction in physical measurements as possible, but the difference between the two cannot be eliminated entirely. After each measurement, adaptive Monte Carlo simulation was carried out. The number of significant digits was set to 1 and the level of confidence was set to 95%. After the simulation was stabilized, the estimate of the ring bore diameter, its associated standard uncertainty and a coverage interval with 95% coverage probability were outputted and recorded. Total 20 sets of simulation results were obtained for all 20 setups.

## 6.5 Results and discussions

In the physical inspections, temperature was recorded in the beginning and end of the measurements at each setup. In case there was difference between the two values, an average temperature was calculated, as shown in Table 6.4. All results measured from physical inspections were corrected to 20.0 °C using a value for the coefficient of linear thermal expansion for steel of 11.7 ppm/°C.

Order of setup	Initial temperature (°C)	Final temperature (°C)	Average temperature (°C)
1	19.7	19.8	19.75
2	19.8	19.8	19.80
3	19.9	19.9	19.90
4	19.9	19.9	19.90
5	19.9	19.9	19.90
6	20.2	20.2	20.20
7	20.3	20.3	20.30
8	20.3	20.3	20.30
9	20.3	20.3	20.30
10	20.3	20.3	20.30
11	19.8	19.8	19.80
12	19.8	19.8	19.80
13	19.7	19.8	19.75
14	19.8	19.8	19.80
15	19.8	19.7	19.75
16	19.7	19.7	19.70
17	19.7	19.8	19.75
18	19.8	19.9	19.85
19	19.9	20.0	19.95
20	20.0	20.0	20.00

Table 6.4 Recorded temperatures for CMM experiments

For each setup, a mean value of the bore diameter was calculated from the 30 measurements. And for each setup, AVCMM predicted an estimated result for that setup from simulation. Figure 6.9 compares these two sets of results. From the figure, it can be clearly observed that the results predicted by AVCMM well resemble the mean values of measured results of each setup, in terms of value distribution, curve shape and general trend. The variation of the predicted results is somehow slightly ‘flatter’ than the mean values of measured results, where the standard deviation of the former is  $1.1 \mu\text{m}$  and the latter is  $1.6 \mu\text{m}$ . This difference may be mainly attributed to the fact that the actual measurements were affected by many other uncertainty contributors that are not considered in this AVCMM prototype.

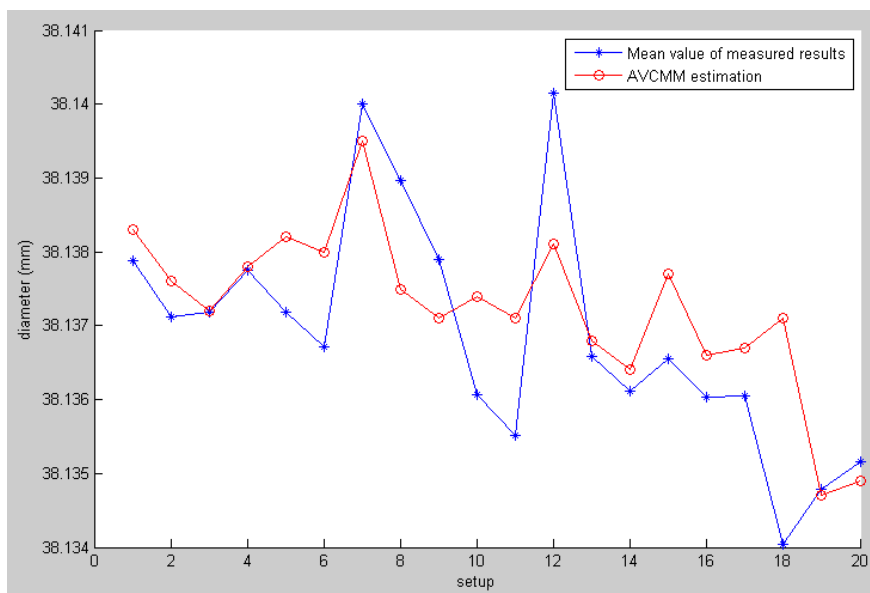
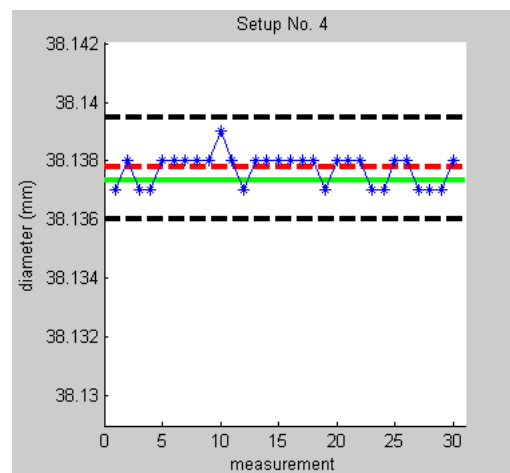
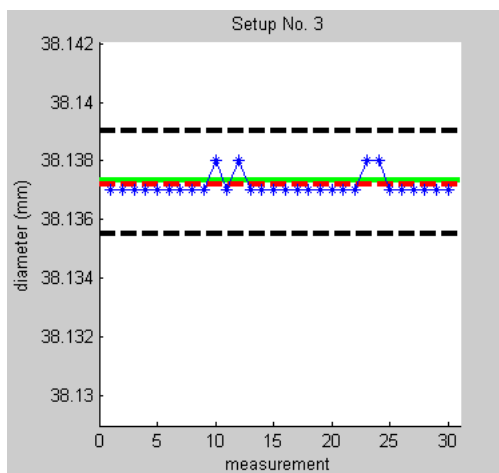
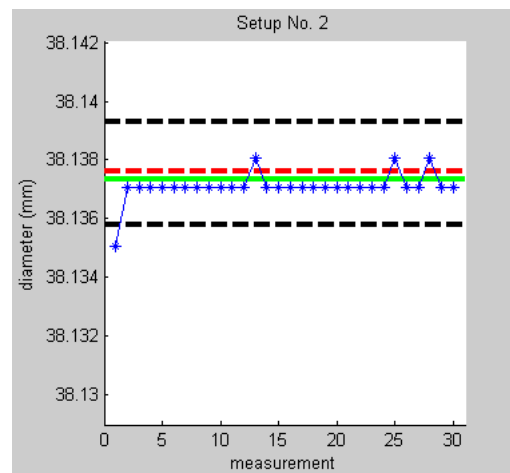
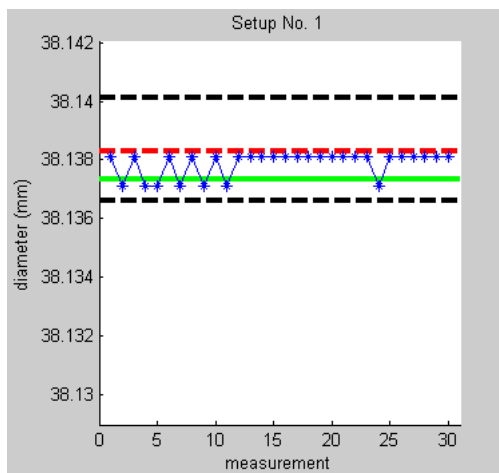
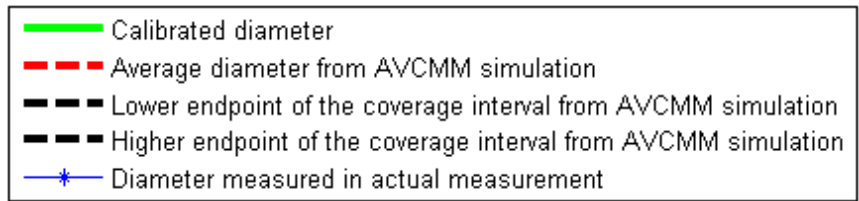
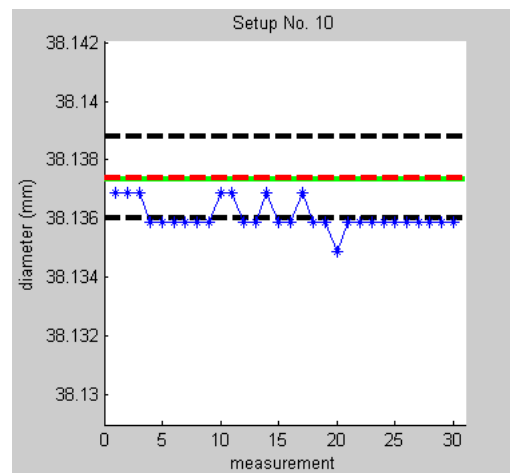
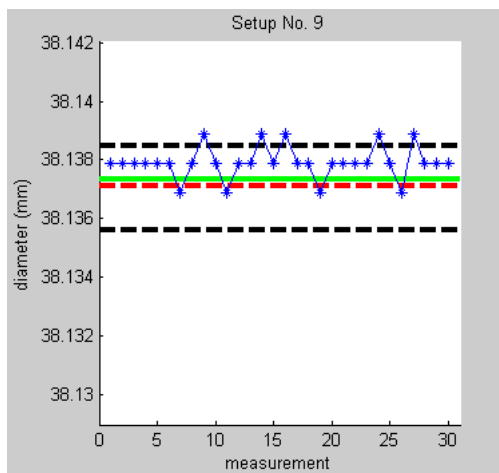
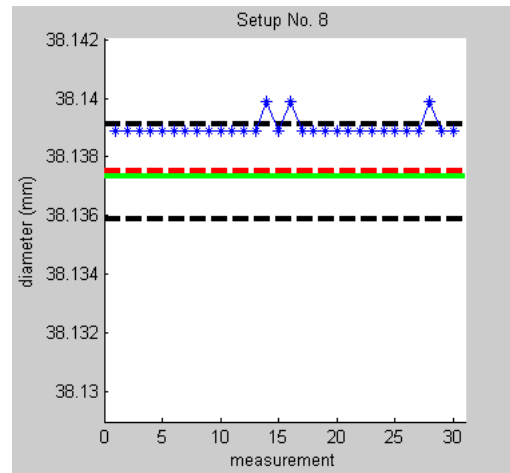
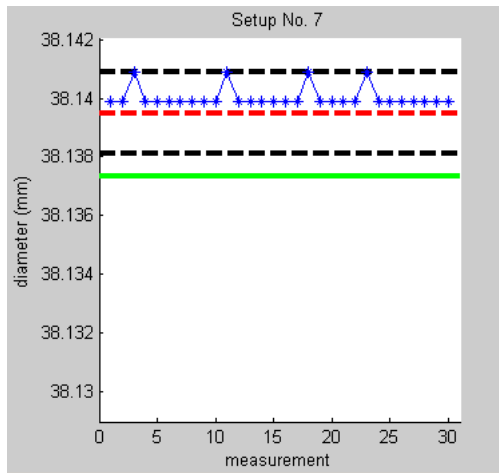
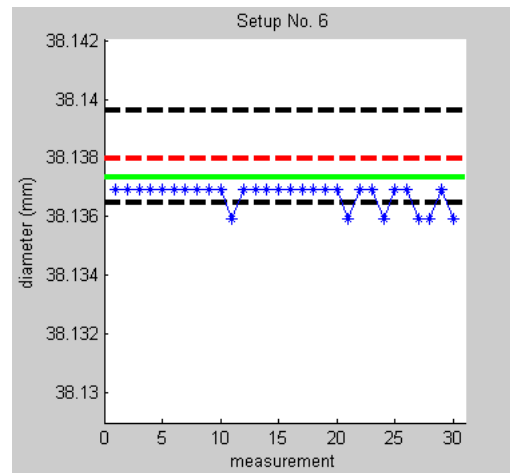
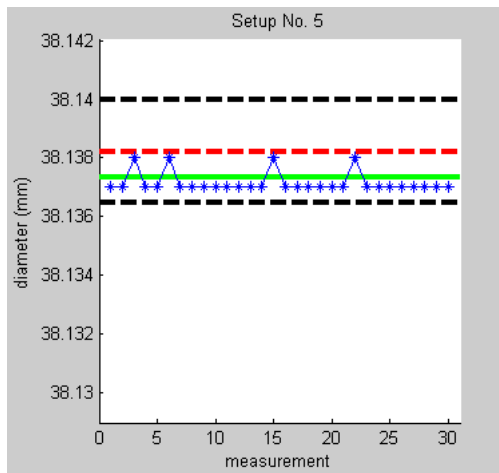


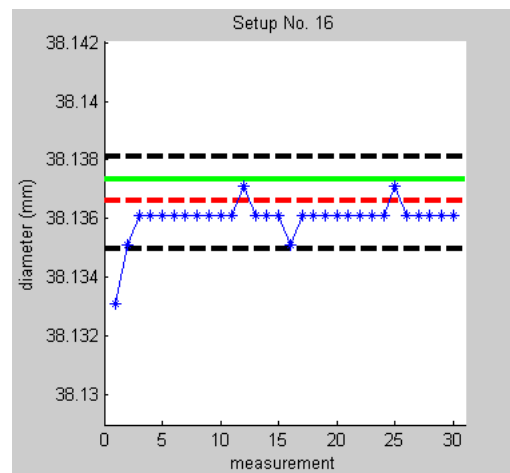
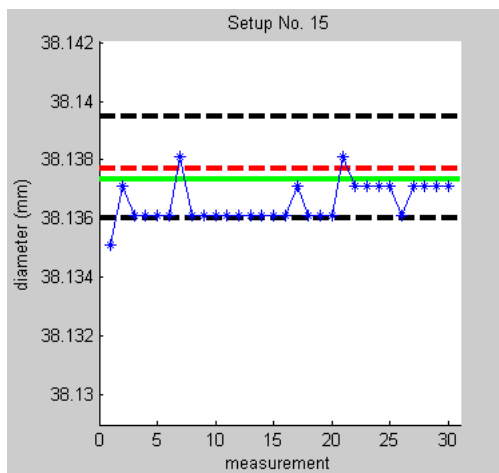
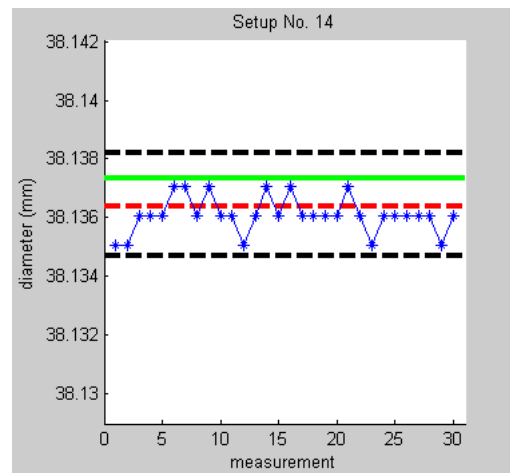
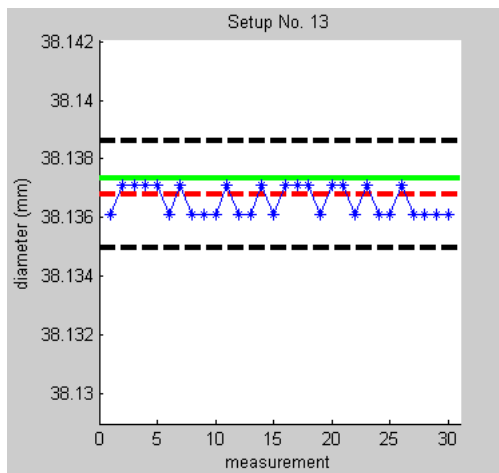
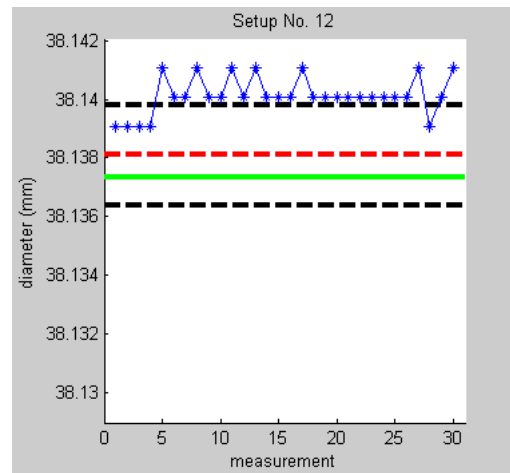
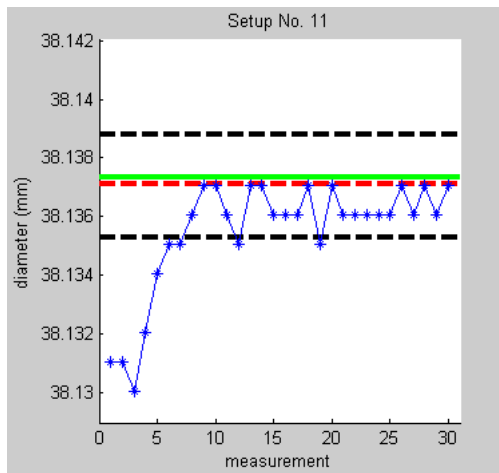
Figure 6.9 Mean values of measurement results versus AVCMM estimation

Legend:









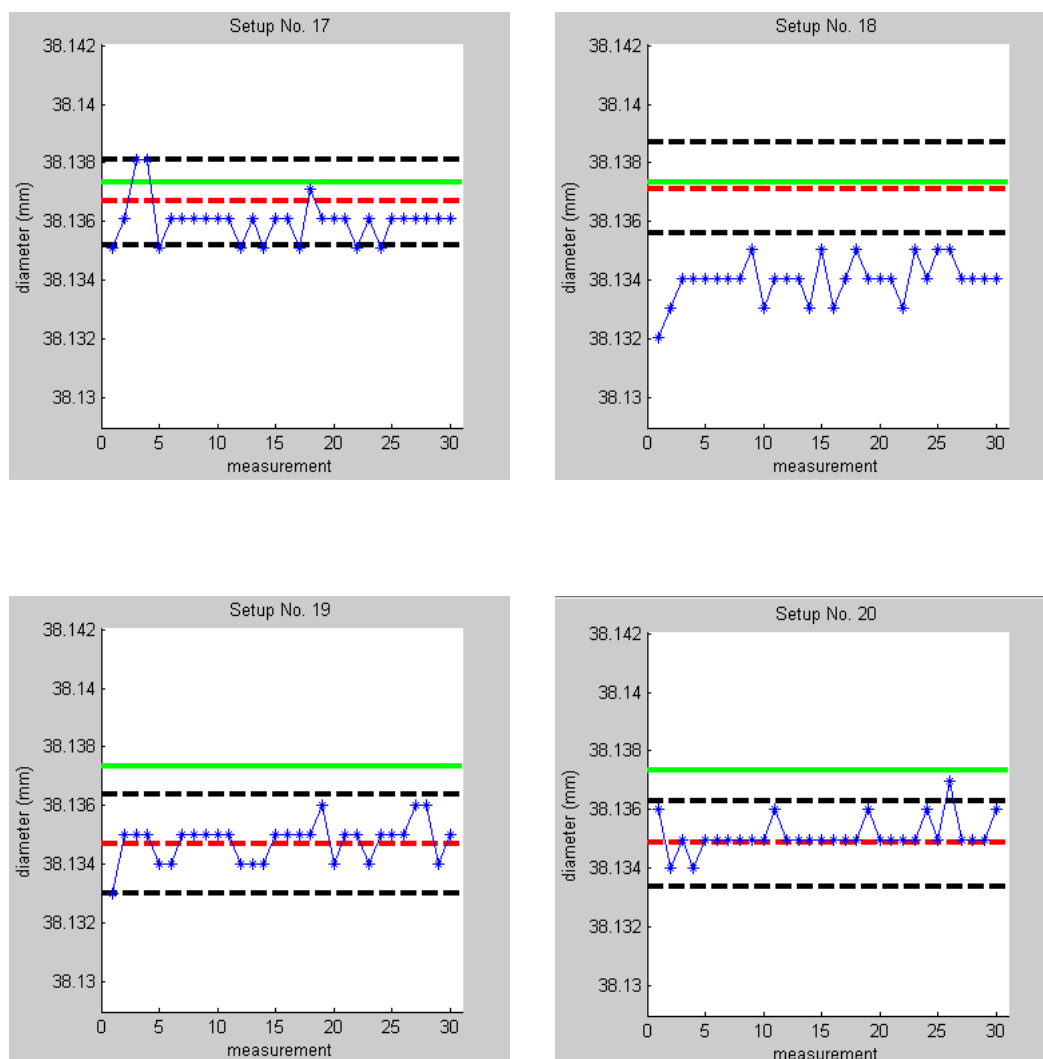


Figure 6.10 Measurement results versus coverage interval predicted by AVCMM

For detailed analysis, for each setup, the 30 measured values of bore diameter were compared to the coverage interval obtained from the corresponding simulation in AVCMM. Figure 6.10 plots both measured values and predicted coverage interval on the same graph for each of the 20 setups individually, so that their relationships can be easily compared. The calibrated value of the bore diameter is also marked for reference. It can be seen that for most setups, the predicted coverage interval encompasses all or majority of the measured values. And for almost all setups, at least some of the

measured values are covered by the predicted coverage interval, with the only exception of setup 18. At setup 18, the measured values clearly deviate from the predicted result and coverage interval, and this deviation may be due to a number of possible reasons including: a) form deviation of the ring gauge; b) difference between actual and virtual probing directions; c) the effects of other unconsidered uncertainty contributors; d) other unnoticed errors introduced during measurements.

According to GUM, in normal circumstances, the result of a measurement should be corrected for all recognized significant systematic effects, instead of taking such effects into account by enlarging the ‘uncertainty’ assigned to the result. In AVCMM, when a point is probed and its error simulated, the error values retrieved from error maps may be considered as estimates of the error components arising from systematic effects of the probe and CMM machine, and the random error part drawn from normal distribution contributes to the uncertainty. Therefore, using AVCMM there are two statistically equivalent methods to obtain the correction to compensate for the systematic effects, regarding a specific measurement task: a) performing a virtual measurement with the random error simulation disabled, so that the difference between the value modelled in VRML file and the result of the virtual measurement approximately indicates the systematic bias; b) in case a calibrated value of the measurand is available and if sufficient trials have been performed in AVCMM Monte Carlo simulation, the difference between the calibrated value and the average result predicted by AVCMM for that measurement task can be treated as an estimate of the systematic bias. In our experiments, we used method b to compensate for systematic effects since all requirements were met. The uncertainty arising from the compensation itself, i.e. the uncertainty of the correction applied to a measurement result, was deemed insignificant comparing to other contributors and was therefore neglected.

Setup	$\bar{y}$	$U$	Plausible?
1	38.13693	0.0018	Yes
2	38.13687	0.0018	Yes
3	38.13733	0.0018	Yes
4	38.13729	0.0018	Yes
5	38.13633	0.0018	Yes
6	38.13606	0.0016	Yes
7	38.13785	0.0014	Yes
8	38.13882	0.0016	Yes
9	38.13815	0.0014	Yes
10	38.13602	0.0014	Yes
11	38.13577	0.0018	Yes
12	38.13941	0.0018	No
13	38.13713	0.0018	Yes
14	38.13707	0.0018	Yes
15	38.13619	0.0018	Yes
16	38.13678	0.0016	Yes
17	38.13669	0.0014	Yes
18	38.13428	0.0016	No
19	38.13744	0.0018	Yes
20	38.13762	0.0014	Yes
Total	-	-	18 Yes

Table 6.5 Plausibility test result for mean values of measurements

After compensation of systematic effects, we tested measurement results against the expanded uncertainties evaluated by AVCMM, following the recommendation of ISO/TS 15530-4:2008 regarding the methods of testing uncertainty evaluating software.

First, we tested the 20 mean values of the measurement results at 20 setups against their associated task-specific expanded uncertainties determined by AVCMM, to find out the proportion of the cases where the plausibility criterion was satisfied. The criterion is that a statement of uncertainty is plausible if:

$$\frac{|\bar{y} - y_{cal}|}{\sqrt{U_{cal}^2 + U^2}} \leq 1 \quad (6.1)$$

where  $\bar{y}$  is the mean value of the results of 30 repeated measurements at each setup;  $y_{cal} = 38.13735 \text{ mm}$  is the calibrated value of the bore diameter;  $U_{cal} = 0.00009 \text{ mm}$  is the expanded uncertainty of the calibrated diameter of the bore, with coverage factor  $k = 2$ ;  $U$  is the task-specific expanded uncertainty determined by AVCMM for each setup, also with coverage factor  $k = 2$ .

After computation, we found that the plausibility criterion was satisfied for 18 out of 20 setups, or 90% of the time, which is very close to the ideal proportion 95% since  $k = 2$ . Table 6.5 lists the results.

Furthermore, we checked each individual measurement within each of the 20 setups against the coverage of the related uncertainty ranges. The plausibility criterion used is slightly different from Inequality (6.1):

$$\frac{|y - y_{cal}|}{\sqrt{U_{cal}^2 + U^2}} \leq 1 \quad (6.2)$$

where  $y$  is the result of each individual measurement, and  $y_{cal}$ ,  $U_{cal}$  and  $U$  remain unchanged.

Setup	Expanded uncertainty	Number of plausible cases	Number of measurements	Proportion of plausible cases
1	0.0018	30	30	100%
2	0.0018	29	30	97%
3	0.0018	30	30	100%
4	0.0018	30	30	100%
5	0.0018	30	30	100%
6	0.0016	24	30	80%
7	0.0014	30	30	100%
8	0.0016	27	30	90%
9	0.0014	25	30	83%
10	0.0014	7	30	23%
11	0.0018	21	30	70%
12	0.0018	5	30	17%
13	0.0018	30	30	100%
14	0.0018	30	30	100%
15	0.0018	29	30	97%
16	0.0016	29	30	97%
17	0.0014	21	30	70%
18	0.0016	0	30	0%
19	0.0018	30	30	100%
20	0.0014	29	30	97%
Total	-	486	600	81%

Table 6.6 Plausibility test result for all measurements

Setup	Number of plausible cases (after adjustment)	Number of measurements	Proportion of plausible cases (after adjustment)
1	30	30	100%
2	29	30	97%
3	30	30	100%
4	30	30	100%
5	30	30	100%
6	24	30	80%
7	30	30	100%
8	27	30	90%
9	25	30	83%
10	29	30	97%
11	25	30	83%
12	23	30	77%
13	30	30	100%
14	30	30	100%
15	29	30	97%
16	29	30	97%
17	30	30	100%
18	0	30	0%
19	30	30	100%
20	29	30	97%
Total	539	600	90%
Total (excluding setup 18)	539	570	95%

Table 6.7 Plausibility test result after applying adjustment

The test result is presented in Table 6.6. It can be seen that for most of the setups, the plausible rate stays quite close to the ideal 95%. And even with the clear deviation of setup 18 (0%) – the possible causes of which we already discussed – the overall proportion of plausible cases is still 81%, reasonably close to 95%. If we take a careful look at the graphs of result distribution in Figure 6.10, it can be clearly observed that a number of points are outside the coverage intervals but right next to the edges, especially for setup 10 and 12. These small differences are very likely to be caused by a simple reason: the AVCMM rounds all outputted values to  $0.0001 \mu m$  while the CMM software rounds all measurement results to  $0.001 \mu m$ . If we include all the ‘edge points’ as plausible cases, the overall proportion of plausible cases can be improved to 90%. If we take a further step of excluding setup 18 as outlier, then the overall proportion become the ideal 95%. Table 6.7 shows the results obtained after applying such adjustment.

While the validity of the AVCMM system was verified by the experiments and analysis, its performance (speed) relative to the hardware on which it is running, was also evaluated during the experiments. As described before, three computers have been utilized to simultaneously run 3 copies of AVCMM for the purpose of comparing the impacts caused by different hardware to the AVCMM performance.

Generally, the AVCMM package can run smoothly on all three systems. Regarding the 3D representation, no noticeable lag was experienced on any of the systems although the graphics were rendered more smoothly with dedicated graphics cards. However, during the test, an issue with the dual-display support was discovered. Regardless of which graphics card was used, whenever a secondary display was attached and enabled, the 3D animation of the virtual environment became extremely slow to an extent that the virtual machine was barely operable. This issue is more likely to be related to one of the supporting packages rather than AVCMM itself.



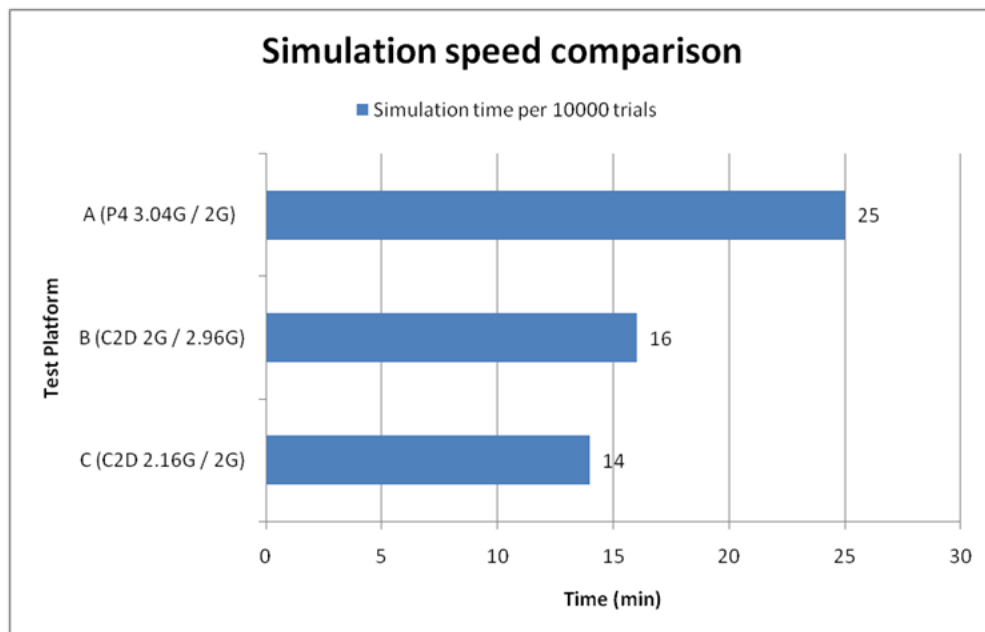


Figure 6.11 Comparison of simulation performance

While the user interface performed roughly the same on all three systems, the time required to run Monte Carlo simulation showed big difference. Figure 6.11 illustrates the average times for running 10000 trials on each of the three platforms. All times recorded were for adaptive Monte Carlo simulation using non-repetition method, of the ring gauge inspection program, in which 4 points were sampled to determine a circle. The number of significant numbers was set to 1 and the level of confidence was set to 95%.

It should be noted that the simulation was running on only one core in all tests. It can be observed that the simulation speed was very sensitive to the performance of CPUs. While RAM should also have an influence on the performance, it seemed that 2G was already satisfactory in these configurations for the tasks we have tested, and the extra 0.96G in computer B did not make any significant difference.

## **6.6 Summary**

In this chapter, the AVCMM system was tested experimentally. The general validity of the estimation made by AVCMM was verified by a set of experiments and careful analysis. The effectiveness and feasibility of utilising AVCMM system in practical use were confirmed and demonstrated during the experiments. A comparison was also performed regarding the speed of AVCMM system on different hardware, which provided a brief evaluation of the AVCMM performance and a guide to the choice of hardware for AVCMM deployment.

## Chapter 7 Conclusions and Future Work

### 7.1 Conclusions

A few important conclusions can be drawn from the discussions and experimental results presented in the previous chapters.

- Urged by the continually rising demand for measurement efficiency and accuracy in the fast developing manufacturing industry, CMM and its related technologies have become an emphasized research field. Among other approaches, virtual CMM (VCMM) is a category of CMM tools that facilitate the planning of optimal inspection program and the evaluation of uncertainty associated with measurement results. These two sets of functionalities were previously handled by two categories of VCMMs, inspection oriented and uncertainty evaluation oriented respectively. The literature review has revealed the lack of a comprehensive, integrated virtual CMM system in the existing solutions.
- The proposed Advanced Virtual Coordinate Measuring Machine (AVCMM) organically combines the two sets of functionalities into one seamless system, in which the same models of CMM machines are used for both inspection planning and uncertainty evaluation. The data acquired from the inspection planning can easily be transferred to the uncertainty evaluation modules and feedbacks are rapidly provided regarding the quality of the inspection path.
- The AVCMM system adopts a modular, multitier architecture to cope with the complexity and requirements for extendibility and scalability. The system includes libraries of CMM and workpiece models, a VRML parser, a novel virtual CMM

module, an uncertainty evaluation module and several interfaces.

- The kinematical models of different types of CMMs have been derived and corresponding VRML models have been created.
- The CMM machine geometric error and probe error have been modelled from FEA simulations and measured data. An error simulator has been developed to generate errors according to probing positions and directions during inspection.
- A new VRML collision detection engine has been designed and implemented for the AVCMM system. The engine supports the detection of collisions in the CMM inspection, and also allows the acquisition of accurate contact points.
- An uncertainty evaluation module has been developed using Monte Carlo method, complying the requirements specified in Supplement 1 to GUM standard. Multiple simulation methods and modes have been provided, allowing either visual demonstration or fast computation. The number of trials can be controlled either manually or adaptively.
- Both local and Web user interfaces have been provided, featuring vivid, interactive 3D representation and convenient operation similar to physical machine.
- The validity and usefulness of AVCMM system have been confirmed by experiments, which have shown significant similarity and closeness between the results from actual inspections and virtual inspections.

## 7.2 Contributions to knowledge

This research project has made a number of contributions to knowledge, which are summarized below.

- The proposed AVCMM is the first to integrate inspection planning with uncertainty evaluation and provide a full-featured comprehensive solution.
- A novel VRML collision detection engine has been designed and implemented, which is suitable for the virtual environment of simulated CMM inspection in terms of high efficiency and accuracy.
- The methodology of transforming CMM kinematical models into VRML models has been established and demonstrated.
- The methodology of modelling volumetric error and probe error for the use of the error simulator in AVCMM has been designed and implemented with error data obtained by other means, i.e. FEA and actual measurement.
- The Monte Carlo method adopted for uncertainty evaluation has been implemented with two methods, namely process repetition and points reuse, offering options for visualization of the simulation and fast computation.
- User interface has been made available both locally with the package and remotely from Web page. The combination of VRML, MATLAB and .NET framework has made a user friendly environment and vivid 3D representation.

### 7.3 Recommendations for future work

This research project intends to provide a complete solution covering various aspects related to CMM measurement. While the current research outcome has proved effective and valid, inevitably there are still certain areas that are not thoroughly investigated due to the wide range of studies involved. Therefore following is a list of recommendations for future improvements.

- Owing to the modular design and layered architecture, the AVCMM can be easily modified and extended. The support for importing CMM and workpiece models created in other formats can be added, avoiding duplication of effort.
- More error sources can be considered and simulated in AVCMM.
- An interface from the AVCMM application to physical CMM can be developed, to allow direct execution of inspection program planned in the virtual environment. Also this interface may make possible the direct acquisition of actual measured data for the AVCMM to provide evaluation of the uncertainty of a performed inspection.
- With cooperation from CMM manufacturers, the AVCMM package can be integrated into the CMM software. This way, AVCMM can share the 3D data processing and CMM programming modules in the CMM software, and can support fast, risk-free online/offline inspection planning in a virtual environment. Furthermore, the AVCMM can provide quick prediction or estimation of measurement uncertainty for a planned or performed inspection.
- Joysticks with force feedback capability can be utilized in the future version of AVCMM, so that user can feel the collisions and contacts in the virtual environment

and select measurement points more easily.

- In addition to the Web based user interface, the AVCMM may be deployed as a Web service, which is essentially a Web API that allows easier utilisation and possibility of implementing various kinds of web applications based upon the AVCMM service.

## References

Albuquerque, V.A., Liou, F.W. & Mitchell, O.R., 2000. Inspection point placement and path planning algorithms for automatic CMM inspection. *International Journal of Computer Integrated Manufacturing*, 13(2), pp.107-20.

Badawy, W.A. & Kelash, H.M., 2003. Collision detection in VRML: A survey. *Emirates Journal for Engineering Research*, 8(2), pp.1-8.

Baldo, C.R. & Donatelli, G.D., 2004. Improvements in the evaluation of the measurement uncertainty of CMMs using calibrated parts. *VDI Berichte*, (1860), pp.251-56.

Balsamo, A. et al., 1999. Evaluation of CMM uncertainty through Monte Carlo simulations. *CIRP Annals - Manufacturing Technology*, 48(1), pp.425-28.

Bauza, M.B., Hocken, R.J., Smith, S.T. & Woody, S.C., 2005. The development of a virtual probe tip with application to high aspect ratio microscale features. *Review of Scientific Instruments*, 76(9), p.095112.

Beaman, J. & Morse, E., 2010. Experimental evaluation of software estimates of task specific measurement uncertainty for CMMs. *Precision Engineering*, 34(1), pp.28-33.

British Standards Institution, 1987. *BS 6808-1:1987 Coordinate measuring machines — Part 1: Glossary of terms*. London: British Standards Institution.

Calonego, N., Abackerli, A.J. & Consularo, L.A., 2006. User interface design for



VCMMs: An approach to increase fidelity and usability. In *Proceedings of 2006 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*. La Coruna, 2006. IEEE.

Calonego, N., Kirner, C., Kirner, T.G. & Abackerli, A.J., 2004. Implementation of a virtual environment for interacting with a coordinate measuring machine. In *IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*. Boston, MA, 2004. IEEE.

Chang, T.C., 1990. *Expert Process Planning for Manufacturing*. Reading, Massachusetts, USA: Addison-Wesley.

Cheng, Y. & Cai, F., 1995. Automated fixturing planning for inspection on coordinate measuring machine. In *Proceedings of the SPIE - The International Society for Optical Engineering. International Conference on Intelligent Manufacturing*. Wuhan, China, 1995. SPIE - The International Society for Optical Engineering. no full text.

Chen, Y.H., Wang, Y.Z. & Yang, Z.Y., 2004. Towards a haptic virtual coordinate measuring machine. *International Journal of Machine Tools & Manufacture*, 44(10), pp.1009-17.

Chen, Y.H., Yang, Z.Y. & Lian, L.L., 2005. On the development of a haptic system for rapid product development. *Computer-Aided Design*, 37(5), pp.559-69.

Chen, Y.H., Yang, Z.Y. & Wang, Y.Z., 2005. Haptic modeling for a virtual coordinate measuring machine. *International Journal of Production Research*, 43(9), pp.1861-78.

Cohen, J.D., Lin, M.C., Manocha, D. & Ponamgi, M.K., 1995. I-COLLIDE: An

interactive and exact collision detection system for large-scale environments. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*. Monterey, CA, 1995.

Cortona3D, n.d. *Cortona3D Viewer*. [Online] Available at: <http://www.parallelgraphics.com/products/cortona3d/> [Accessed 11 February 2010].

Cortona3D, n.d. *Object-to-object collision detection interface*. [Online] Available at: <http://www.parallelgraphics.com/developer/products/cortona/extensions/collision> [Accessed 11 February 2010].

Cui, C., Fu, S. & Huang, F., 2009. Research on the uncertainties from different form error evaluation methods by CMM sampling. *International Journal of Advanced Manufacturing Technology*, 43(1-2), pp.136-45.

De Aquino Silva, J.B. et al., 2009. Approach for uncertainty analysis and error evaluation of four-axis co-ordinate measuring machines. *International Journal of Advanced Manufacturing Technology*, 41(11-12), pp.1130-39.

Ehmann, S.A. & Lin, M.C., 2000. Accelerated proximity queries between convex polyhedra by multi-level voronoi marching. In *Proceedings of International Conference on Intelligent Robots and Systems*. Takamatsu, 2000.

Ehmann, S.A. & Lin, M.C., 2002. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Computer Graphics Forum*, 20(3), pp.500-10.

Forbes, A.B., 2006. Surface fitting taking into account uncertainty structure in coordinate data. *Measurement Science and Technology*, 17(3), pp.553-58.

Forbes, A.B., 2006. Uncertainty evaluation associated with fitting geometric surfaces to coordinate data. *Metrologia*, 43, pp.S282-90.

Furutani, R. et al., 1994. Autonomous inspection planning system with coordinate measuring machine. In *Proceedings of ICARCV'94, Third International Conference on Automation, Robotics and Computer Vision*. Singapore, 1994. Nanyang Technology University. no full text.

Gerhardt, L.A. & Hyun, K., 1995. View planning applied to coordinate measuring machine (CMM) measurement. In *Proceedings IEEE Conference on Industrial Automation and Control Emerging Technology Applications*. Taipei, Taiwan, 1995. IEEE.

Gottschalk, S., Lin, M. & Manocha, D., 1996. OBB-tree: A hierarchical structure for rapid interference detection. In *Proceedings of ACM SIGGRAPH 96*. New Orleans, Louisiana, 1996.

Gregory, A.D., Lin, M.C., Gottschalk, S. & Taylor, R.M., 1999. H-Collide: A framework for fast and accurate collision detection for haptic interaction. In *Proceedings of IEEE Virtual Reality Conference 1999*. Houston, TX, 1999.

Gu, P., 1994. A knowledge-based inspection process planning system for coordinate measuring machines. *Journal of Intelligent Manufacturing*, 5(5), pp.351-63.

Haitjema, H., Van Dorp, B., Morel, M. & Schellekens, P.H.J., 2001. Uncertainty estimation by the concept of virtual instruments. In Decker, J.E. & Brown, N., eds. *Recent Developments in Traceable Dimensional Measurements, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*. Munich, Germany, 2001.

SPIE.

Hoff, K.E., Zaferakis, A., Lin, M. & Manocha, D., 2002. *Fast 3D geometric proximity queries between rigid and deformable models using graphics hardware acceleration*. Technical Report. Department of Computer Science, University of North Carolina.

Hudson, T.C. et al., 1997. V-COLLIDE: Accelerated collision detection for VRML. In *Proceedings of the 1997 2nd Symposium on the Virtual Reality Modeling Language, VRML*. Monterey, CA, 1997. ACM.

International Organization for Standardization, 1993, corrected and reprinted in 1995. *ISO/IEC Guide 98:1995 Guide to the expression of uncertainty in measurement (GUM)*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 1993. *International vocabulary of basic and general terms in metrology*. second edition ed. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 1997. *ISO/IEC 14772-1:1997 Information technology -- Computer graphics and image processing -- The Virtual Reality Modeling Language -- Part 1: Functional specification and UTF-8 encoding*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 2000. *ISO 10360-1:2000 Geometrical Product Specifications (GPS) — Acceptance and reverification tests for coordinate measuring machines (CMM) — Part 1: Vocabulary*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 2004. *ISO/TS 15530-3:2004(E) Geometrical product specifications (GPS) — Coordinate measuring machines (CMM): Technique for determining the uncertainty of measurement — Part 3: Use of calibrated workpieces or standards*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 2006. *ISO/TS 23165:2006(E) Geometrical product specifications (GPS) — Guidelines for the evaluation of coordinate measuring machine (CMM) test uncertainty*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 2008. *ISO/IEC GUIDE 98-3:2008(E) Uncertainty of measurement — Part 3: Guide to the expression of uncertainty in measurement (GUM:1995)*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 2008. *ISO/IEC Guide 98-3:2008/Suppl 1:2008 Propagation of distributions using a Monte Carlo method*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 2008. *ISO/TS 15530-4:2008(E) Geometrical product specifications (GPS) — Coordinate measuring machines (CMM): Technique for determining the uncertainty of measurement — Part 4: Evaluating task-specific measurement uncertainty using simulation*. Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization, 2009. *ISO/DIS 15530-3:2009 Geometrical product specifications (GPS) — Coordinate measuring machines (CMM):*

*Technique for determining the uncertainty of measurement — Part 3: Use of calibrated workpieces or standards.* Geneva, Switzerland: International Organization for Standardization.

Jakubiec, W. & Starczak, M., 2004. EMU - friendly software for estimation of measurement uncertainty for CMM. In *8th International Symposium on Measurement and Quality Control in Production*. Erlangen, Germany, 2004.

Jankovic, L., 2000. Games development in VRML. *Virtual Reality*, 5(4), pp.195-203.

Jiang, B.C. & Chiu, S.D., 2002. Form tolerance-based measurement points determination with CMM. *Journal of Intelligent Manufacturing*, 13(2), pp.101-08.

Kim, Y.J., Lin, M.C. & Manocha, D., 2002. DEEP: Dual-space Expansion for Estimating Penetration Depth between convex polytopes. In *IEEE International Conference on Robotics and Automation*. Washington, DC, 2002.

Kruth, J.P., Van Gestel, N., Bleys, P. & Welkenhuyzen, F., 2009. Uncertainty determination for CMMs by Monte Carlo simulation integrating feature form deviations. *CIRP Annals - Manufacturing Technology*, 58(1), pp.463-66.

Kurfess, T.R., 2006. What can CMMs do? They can measure almost anything. *Manufacturing Engineering*, 136(3), p.173.

Larsen, E., Gottschalk, S., Lin, M.C. & Manocha, D., 2000. Fast proximity queries with swept sphere volumes. In *Proceedings of IEEE International Conference on Robotics and Automation*. San Francisco, CA, 2000.

Limaiem, A. & ElMaraghy, H.A., 2000. Integrated accessibility analysis and measurement operations sequencing for CMMs. *Journal of Manufacturing Systems*, 19(2), pp.83-93.

Lim, C.K. & Burdekin, M., 2002. Rapid volumetric calibration of coordinate measuring machines using a hole bar artefact. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* , 216(8), pp.1083-93.

Lim, C.P. & Menq, C.H., 1994. CMM feature accessibility and path generation. *International Journal of Production Research*, 32(3), pp.597-618.

Lin, W.G., Che, R.S. & Chen, G., 1999. A study of virtual inspection of virtual coordinate measuring machine. In *Conference Proceedings of ICEMI'99: Fourth International Conference on Electronic Measurement & Instruments*. Harbin, China, 1999.

Lin, Y.J. & Mahabaleshwarkar, R., 1999. A generic algorithm for CAD-directed CMM dimensional inspection planning. In *Proceedings of International Conference on Robotics and Automation*. Detroit, MI, USA, 1999. IEEE.

Lin, Y.J., Mahabaleshwarkar, R. & Massina, E., 2001. CAD-based CMM dimensional inspection path planning – a generic algorithm. *Robotica*, 19(2), pp.137-48.

Lin, Y.J. & Murugappan, P., 2000. A new algorithm for CAD-directed CMM dimensional inspection. *International Journal of Advanced Manufacturing Technology*, 16(2), pp.107-12.

Liu, X.L., Ma, S.Y., Wu, P.D. & Chen, Z.L., 2002. An Internet-based measurement

system for coordinate measurement machines. In *Proceedings of the Second International Symposium on Instrumentation Science and Technology*. Jinan, China, 2002.

Lu, C.G., Morton, D., Wu, M.H. & Myler, P., 1999. Genetic algorithm modelling and solution of inspection path planning on a coordinate measuring machine (CMM). *International Journal of Advanced Manufacturing Technology*, 15(6), pp.409-16.

Lu, C.G., Myler, P. & Wu, M.H., 1995. An artificial intelligence path planning system for multiple tasks inspection on co-ordinate measuring machine. In *Proceedings of the 31st International MATADOR Conference*. Manchester, UK, 1995. Macmillan Press.

Lu, C., Wu, M.H., Mylar, P. & Healey, R., 1995. Developing an information model for LINDO software input-optimal path planning for CMM. In *Proceedings of 1st World Congress on Intelligent Manufacturing Processes and Systems*. Mayaguez/San Juan, Puerto Rico, 1995. University of Puerto Rico.

Ma, S.Q. & Gao, Z.L., 2009. A solution to integrate CAD/CAE system and VR for complex product design processes. In *2nd International Conference on Information and Computing Science*. Manchester, England, 2009. IEEE.

Majstorovic, V.D., Bojanic, P. & Milacic, V., 1995. Expert system for inspection planning on coordinate measuring machine. In *Proceedings of 1st World Congress on Intelligent Manufacturing Processes and Systems*. Mayaguez/San Juan, Puerto Rico, 1995. University of Puerto Rico.

McMurtry, D.R., 1979. *US Patent 4153998 - Probes*. United States.



Murray, G., 2005. *Rotation about an arbitrary axis in 3 dimensions*. [Online] Available at: <http://inside.mines.edu/~gmurray/ArbitraryAxisRotation/ArbitraryAxisRotation.html> [Accessed 27 April 2010].

Pahk, H.J., Burdekin, M. & Peggs, G.N., 1998. Development of virtual coordinate measuring machines incorporating probe errors. *Proceedings of the Institution of Mechanical Engineers Part B - Journal of Engineering Manufacture*, 212(7), pp.533-48.

Peggs, G., 2003. Virtual technologies for advanced manufacturing and metrology. *International Journal of Computer Integrated Manufacturing*, 16(7), pp.485-90.

Phillips, S.D. et al., 1997. The calculation of CMM measurement uncertainty via the method of simulation by constraints. In *Proceedings of the 12th Annual Meeting of the American Society for Precision Engineering*. Norfolk, VA, 1997.

Qin, Y., Wang, L., Xie, L. & Huang, Y., 2008. Vision guided automatic measuring in coordinate metrology. In *Fourth International Symposium on Precision Mechanical Measurements*. Anhui, China, 2008. SPIE - The International Society for Optical Engineering.

Raghunandan, R. & Venkateswara Rao, P., 2008. Selection of sampling points for accurate evaluation of flatness error using coordinate measuring machine. *Journal of Materials Processing Technology*, 202(1-3), pp.240-45.

Rodger, G., Flack, D. & McCarthy, M., 2007. *A review of industrial capabilities to measure free-form surfaces*. London: National Physical Laboratory National Physical Laboratory.

Salleh, M.R., Yang, Q.P. & Jones, B., 2006. Evaluation of touch trigger probe measurement uncertainty using FEA. In *23rd IEEE Instrumentation and Measurement Technology Conference*. Sorrento, Italy, 2006. IEEE.

Shen, T.S., Huang, J. & Menq, C.H., 2000. Multiple-sensor integration for rapid and high-precision coordinate metrology. *IEEE/ASME Transactions on Mechatronics*, 5(2), pp.110-21.

Shen, Y.L. & Zhang, X., 1997. Modelling of pretravel for touch trigger probes on indexable probe heads on coordinate measuring machines. *The International Journal of Advanced Manufacturing Technology*, 13(3), pp.206-13.

Sprauel, J.M., Linares, J.M., Bachmann, J. & Bourdet, P., 2003. Uncertainties in CMM measurements, control of ISO specifications. *CIRP Annals - Manufacturing Technology*, 52(1), pp.423-26.

Spyridi, A.J. & Requicha, A.A.G., 1990. Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines. In *Proceedings of 1990 IEEE International Conference on Robotics and Automation*. Los Angeles, CA, USA, 1990. IEEE.

Stouffer, K. & Horst, J., 2000. Controller driven VRML animation of the next generation inspection system (NGIS) real-time controller. In *Conference on Sensors and Controls for Intelligent Manufacturing*. Boston, MA, 2000. SPIE.

Summerhays, K.D. et al., 2002. A tool for determining task-specific measurement uncertainties in GD&T parameters obtained from coordinate measuring machines. In *Proceedings of the 17th Annual Meeting of the American Society for Precision*

*Engineering*. St. Louis, Missouri, 2002.

Takamasu, K., 2002. *International standard development of virtual CMM (coordinate measuring machine)*. NEDO.

Takamasu, K. et al., 2004. Evaluation of uncertainty by form deviations of measured workpieces in specified measuring strategies. In *8th International Symposium on Measurement and Quality Control in Production*. Erlangen, Germany, 2004.

Trenk, M., Franke, M. & Schwenke, H., 2004. The "virtual CMM" a software tool for uncertainty evaluation - Practical application in an accredited calibration lab. In *ASPE Summer Topical Meeting on Uncertainty Analysis in Measurement and Design*. State College, Pennsylvania, USA , 2004.

Van Dorp, B. et al., 2001. Virtual CMM using Monte Carlo methods based on frequency content of the error signal. In Decker, J.E. & Brown, N., eds. *Recent Developments in Traceable Dimensional Measurements, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*. Munich, Germany, 2001. SPIE.

Waldele, F. & Schwenke, H., 2002. Automated calculation of measurement uncertainties on CMMs - Towards industrial application. *Technisches Messen*, 69(12), pp.550-57.

Wang, Y., Chen, Y., Nan, Z. & Hu, Y., 2006. Accessibility analysis for CMM inspection planning using haptic device. In *IEEE International Conference on Robotics and Biomimetics - ROBIO2006*. Kunming, China, 2006. IEEE.

Wang, Y. et al., 2009. Accessibility analysis for CMM inspection planning by means of haptic device and STL representation. In *2009 IEEE International Conference on*

*Virtual Environments, Human-Computer Interfaces and Measurements Systems (VECIMS 2009)*. Hong Kong, China, 2009. IEEE.

Wilson, A., Larsen, E., Manocha, D. & Lin, M.C., 1999. Partitioning and handling massive models for interactive collision detection. *Computer Graphics Forum*, 18(3), pp.C327-29.

Yang, Q., 1992. *A high precision probe system for three dimensional coordinate measurement. PhD Thesis*. London: Brunel University.

Yang, Q., Butler, C. & Baird, P., 1996. Error compensation of touch trigger probes. *Measurement*, 18(1), pp.47-57.

Yang, Z. & Chen, Y., 2005. Inspection path generation in haptic virtual CMM. *Computer-Aided Design and Applications*, 2(1-4), pp.273-82.

Ziemian, C.W. & Medeiros, D.J., 1997. Automated feature accessibility algorithm for inspection on a coordinate measuring machine. *International Journal of Production Research*, 35(10), pp.2839-56.

Ziemian, C.W. & Medeiros, D.J., 1998. Automating probe selection and part setup planning for inspection on a coordinate measuring machine. *International Journal of Computer Integrated Manufacturing*, 11(5), pp.448-60.

## **Appendices**

**Appendix A Measurement results**

Table A-1 records the measurement results of the ring gauge bore diameter of all 20 setups. All results listed in the table are directly outputted by CMM software, before correction for temperature.

<b>Ring Gauge Bore Diameter Measurement Results</b>					
(Before correction for temperature)					
Feature	Number of Points	Center X (mm)	Center Y (mm)	Center Z (mm)	Diameter (mm)
<b>Setup 1</b>					
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.004	0	38.137
CIRCLE	4	0.001	0.003	0	38.138
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138

CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138
CIRCLE	4	0.002	0.002	0	38.138
<b>Setup 2</b>					
CIRCLE	4	-0.002	-0.001	0	38.135
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.138
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.138
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.138
CIRCLE	4	-0.004	-0.002	0	38.137
CIRCLE	4	-0.004	-0.002	0	38.137
<b>Setup 3</b>					
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137

CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.138
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.138
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.005	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.138
CIRCLE	4	0.004	0.001	0	38.138
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
CIRCLE	4	0.004	0.001	0	38.137
<b>Setup 4</b>					
CIRCLE	4	-0.003	0.001	0	38.137
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.003	0.001	0	38.137
CIRCLE	4	-0.004	0	0	38.137
CIRCLE	4	-0.003	0	0	38.138
CIRCLE	4	-0.003	0	0	38.138
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.003	0	0	38.138
CIRCLE	4	-0.004	0	0	38.139
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.003	0.001	0	38.137
CIRCLE	4	-0.003	0	0	38.138
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.003	0	0	38.138
CIRCLE	4	-0.004	0	0	38.138



CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.004	0.001	0	38.137
CIRCLE	4	-0.003	0	0	38.138
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.004	0	0	38.138
CIRCLE	4	-0.003	0.001	0	38.137
CIRCLE	4	-0.003	0.001	0	38.137
CIRCLE	4	-0.003	0	0	38.138
CIRCLE	4	-0.003	0.001	0	38.138
CIRCLE	4	-0.004	0	0	38.137
CIRCLE	4	-0.004	0	0	38.137
CIRCLE	4	-0.004	-0.001	0	38.137
CIRCLE	4	-0.003	0	0	38.138
<b>Setup 5</b>					
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0	0.002	0	38.138
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.001	0	38.137
CIRCLE	4	0	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137

CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
<b>Setup 6</b>					
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.136
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.136
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.136
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.136
CIRCLE	4	-0.291	-0.004	0	38.136
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.136
CIRCLE	4	-0.291	-0.004	0	38.136
CIRCLE	4	-0.291	-0.003	0	38.137
CIRCLE	4	-0.291	-0.004	0	38.136
<b>Setup 7</b>					
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.027	0.53	0	38.141
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14

CIRCLE	4	-0.027	0.529	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.529	0	38.141
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.141
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.141
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
CIRCLE	4	-0.028	0.531	0	38.14
CIRCLE	4	-0.028	0.529	0	38.14
CIRCLE	4	-0.028	0.53	0	38.14
<b>Setup 8</b>					
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.14
CIRCLE	4	0	0	0	38.139

CIRCLE	4	-0.001	0	0	38.14
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0	0	0	38.139
CIRCLE	4	0.001	0.001	0	38.14
CIRCLE	4	0	0.001	0	38.139
CIRCLE	4	0	0	0	38.139
<b>Setup 9</b>					
CIRCLE	4	0	-0.002	0	38.138
CIRCLE	4	0	-0.003	0	38.138
CIRCLE	4	0	-0.002	0	38.138
CIRCLE	4	0	-0.002	0	38.138
CIRCLE	4	0	-0.002	0	38.138
CIRCLE	4	0	-0.003	0	38.138
CIRCLE	4	0	-0.003	0	38.137
CIRCLE	4	0	-0.003	0	38.138
CIRCLE	4	0	-0.003	0	38.139
CIRCLE	4	0	-0.003	0	38.138
CIRCLE	4	0	-0.003	0	38.137
CIRCLE	4	0	-0.003	0	38.138
CIRCLE	4	0	-0.003	0	38.138
CIRCLE	4	0	-0.003	0	38.139
CIRCLE	4	0	-0.003	0	38.138
CIRCLE	4	0	-0.003	0	38.139
CIRCLE	4	-0.001	-0.003	0	38.138
CIRCLE	4	-0.001	-0.003	0	38.138
CIRCLE	4	0	-0.003	0	38.137
CIRCLE	4	-0.001	-0.002	0	38.138
CIRCLE	4	-0.001	-0.003	0	38.138
CIRCLE	4	-0.001	-0.003	0	38.138
CIRCLE	4	-0.001	-0.003	0	38.138
CIRCLE	4	0.001	-0.003	0	38.139
CIRCLE	4	-0.001	-0.003	0	38.138



CIRCLE	4	-0.006	0.003	0	38.134
CIRCLE	4	-0.005	0.004	0	38.135
CIRCLE	4	-0.005	0.004	0	38.135
CIRCLE	4	-0.005	0.004	0	38.136
CIRCLE	4	-0.005	0.004	0	38.137
CIRCLE	4	-0.005	0.004	0	38.137
CIRCLE	4	-0.005	0.004	0	38.136
CIRCLE	4	-0.006	0.003	0	38.135
CIRCLE	4	-0.004	0.002	0	38.137
CIRCLE	4	-0.005	0.004	0	38.137
CIRCLE	4	-0.004	0.004	0	38.136
CIRCLE	4	-0.005	0.004	0	38.136
CIRCLE	4	-0.006	0.004	0	38.136
CIRCLE	4	-0.005	0.003	0	38.137
CIRCLE	4	-0.005	0.004	0	38.135
CIRCLE	4	-0.005	0.004	0	38.137
CIRCLE	4	-0.004	0.003	0	38.136
CIRCLE	4	-0.005	0.003	0	38.136
CIRCLE	4	-0.004	0.003	0	38.136
CIRCLE	4	-0.004	0.003	0	38.136
CIRCLE	4	-0.004	0.003	0	38.136
CIRCLE	4	-0.005	0.002	0	38.137
CIRCLE	4	-0.005	0.003	0	38.136
CIRCLE	4	-0.004	0.002	0	38.137
CIRCLE	4	-0.005	0.003	0	38.136
CIRCLE	4	-0.005	0.003	0	38.137
<b>Setup 12</b>					
CIRCLE	4	-0.003	-0.002	0	38.139
CIRCLE	4	-0.004	-0.003	0	38.139
CIRCLE	4	-0.004	-0.003	0	38.139
CIRCLE	4	-0.004	-0.003	0	38.139
CIRCLE	4	-0.004	-0.003	0	38.141
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.141
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.141
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.141
CIRCLE	4	-0.004	-0.003	0	38.14

---

CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.141
CIRCLE	4	-0.003	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.003	-0.003	0	38.14
CIRCLE	4	-0.003	-0.003	0	38.14
CIRCLE	4	-0.003	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.14
CIRCLE	4	-0.005	-0.003	0	38.141
CIRCLE	4	-0.004	-0.003	0	38.139
CIRCLE	4	-0.003	-0.003	0	38.14
CIRCLE	4	-0.004	-0.003	0	38.141
<b>Setup 13</b>					
CIRCLE	4	0.002	0.002	0	38.136
CIRCLE	4	0.002	0.002	0	38.137
CIRCLE	4	0.002	0.002	0	38.137
CIRCLE	4	0.002	0.002	0	38.137
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137

CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.137
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
CIRCLE	4	0.002	0.003	0	38.136
<b>Setup 14</b>					
CIRCLE	4	-0.002	-0.001	0	38.135
CIRCLE	4	-0.001	-0.001	0	38.135
CIRCLE	4	-0.002	-0.001	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	-0.002	-0.001	0	38.137
CIRCLE	4	-0.001	-0.001	0	38.137
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.137
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	0	-0.001	0	38.135
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	0	-0.001	0	38.137
CIRCLE	4	0	-0.001	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.137
CIRCLE	4	0	-0.001	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	0	-0.001	0	38.136
CIRCLE	4	0	-0.001	0	38.136
CIRCLE	4	-0.001	0	0	38.136
CIRCLE	4	0	-0.001	0	38.137
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	0	-0.001	0	38.135
CIRCLE	4	0	0	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	0	0	0	38.136
CIRCLE	4	0	0	0	38.136
CIRCLE	4	0	0	0	38.136
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	0	0	0	38.136
<b>Setup 15</b>					
CIRCLE	4	0.001	0.001	0	38.135
CIRCLE	4	0.001	0	0	38.137
CIRCLE	4	0.001	0.001	0	38.136



Appendix A Measurement results

CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	-0.001	0.001	0	38.138
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.002	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.002	0.001	0	38.136
CIRCLE	4	0.002	0.001	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0.001	0.001	0	38.136
CIRCLE	4	0	0.002	0	38.138
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.002	0.002	0	38.136
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
CIRCLE	4	0.001	0.002	0	38.137
<b>Setup 16</b>					
CIRCLE	4	-0.004	0.002	0	38.133
CIRCLE	4	-0.004	0.001	0	38.135
CIRCLE	4	-0.004	0.002	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.002	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.137
CIRCLE	4	-0.004	0.001	0	38.136

CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.135
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.005	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.005	0.001	0	38.137
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
CIRCLE	4	-0.004	0.001	0	38.136
<b>Setup 17</b>					
CIRCLE	4	0.008	0.003	0	38.135
CIRCLE	4	0.008	0.003	0	38.136
CIRCLE	4	0.007	0.003	0	38.138
CIRCLE	4	0.007	0.003	0	38.138
CIRCLE	4	0.009	0.003	0	38.135
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.135
CIRCLE	4	0.008	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.135
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.01	0.003	0	38.135
CIRCLE	4	0.009	0.003	0	38.137
CIRCLE	4	0.01	0.004	0	38.136
CIRCLE	4	0.01	0.003	0	38.136
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.01	0.003	0	38.135
CIRCLE	4	0.01	0.003	0	38.136

CIRCLE	4	0.01	0.003	0	38.135
CIRCLE	4	0.009	0.003	0	38.136
CIRCLE	4	0.01	0.003	0	38.136
CIRCLE	4	0.011	0.004	0	38.136
CIRCLE	4	0.011	0.003	0	38.136
CIRCLE	4	0.01	0.003	0	38.136
CIRCLE	4	0.011	0.003	0	38.136
<b>Setup 18</b>					
CIRCLE	4	0.002	-0.001	0	38.132
CIRCLE	4	0.002	0	0	38.133
CIRCLE	4	0.002	0	0	38.134
CIRCLE	4	0.002	0	0	38.134
CIRCLE	4	0.002	0	0	38.134
CIRCLE	4	0.002	0	0	38.134
CIRCLE	4	0.002	0	0	38.134
CIRCLE	4	0.002	0	0	38.134
CIRCLE	4	0.002	0	0	38.134
CIRCLE	4	0.003	-0.001	0	38.135
CIRCLE	4	0.002	0	0	38.133
CIRCLE	4	0.003	0	0	38.134
CIRCLE	4	0.003	0	0	38.134
CIRCLE	4	0.003	0	0	38.134
CIRCLE	4	0.003	0	0	38.133
CIRCLE	4	0.002	0	0	38.135
CIRCLE	4	0.003	0	0	38.133
CIRCLE	4	0.003	-0.001	0	38.134
CIRCLE	4	0.003	0	0	38.135
CIRCLE	4	0.004	0	0	38.134
CIRCLE	4	0.003	0	0	38.134
CIRCLE	4	0.003	0	0	38.134
CIRCLE	4	0.003	0	0	38.133
CIRCLE	4	0.005	0	0	38.135
CIRCLE	4	0.003	-0.001	0	38.134
CIRCLE	4	0.005	0	0	38.135
CIRCLE	4	0.004	-0.001	0	38.135
CIRCLE	4	0.004	-0.001	0	38.134
CIRCLE	4	0.004	0	0	38.134
CIRCLE	4	0.004	0	0	38.134
CIRCLE	4	0.004	0	0	38.134
<b>Setup 19</b>					
CIRCLE	4	0.001	0.001	0	38.133
CIRCLE	4	-0.001	0	0	38.135

CIRCLE	4	0	0.001	0	38.135
CIRCLE	4	0	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	0	0.001	0	38.135
CIRCLE	4	0	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	0	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.136
CIRCLE	4	-0.001	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.002	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	-0.001	0.001	0	38.135
CIRCLE	4	0	0.001	0	38.136
CIRCLE	4	-0.001	0.001	0	38.136
CIRCLE	4	-0.002	0.001	0	38.134
CIRCLE	4	-0.001	0.001	0	38.135
<b>Setup 20</b>					
CIRCLE	4	0	0	0	38.136
CIRCLE	4	-0.001	-0.001	0	38.134
CIRCLE	4	0	-0.001	0	38.135
CIRCLE	4	-0.001	0	0	38.134
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	0	0	0	38.135
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	0	0	0	38.135
CIRCLE	4	-0.001	-0.001	0	38.135
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	-0.001	0	0	38.136
CIRCLE	4	-0.001	0	0	38.135

CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	0	0	0	38.135
CIRCLE	4	-0.001	-0.001	0	38.135
CIRCLE	4	0	0	0	38.135
CIRCLE	4	0	0	0	38.135
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	-0.001	-0.001	0	38.136
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	0	0	0	38.135
CIRCLE	4	0	0	0	38.135
CIRCLE	4	0	-0.001	0	38.135
CIRCLE	4	-0.001	0	0	38.136
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	-0.001	-0.001	0	38.137
CIRCLE	4	0	0	0	38.135
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	-0.001	0	0	38.135
CIRCLE	4	-0.001	-0.001	0	38.136

Table A-1 Ring gauge bore diameter measurement results

## Appendix B Segments of programs

Due to the length of the programs, Appendix B only lists two segments of the code used in the AVCMM system and the analysis of experimental results.

The function for adaptive Monte Carlo simulation (non-repetition method):

```
function [ya uya ylowa yhigha totalTrial stableFlag]=amcSim(obj,
vcmmObj,nDig,p)
% Adaptive Monte Carlo Simulation

% Inputs:
% obj: Handle to the estimator object
% vcmmObj: Handle to the virtual cmm object
% nDig: number of significant digits
% p: coverage probability

% Outputs:
% ya: estimate of output quantity (average)
% uya: standard uncertainty associated with ya
% (ylowa, yhigha): coverage interval for the ouput quantity,
% with coverage probability of 100p%
% totalTrial: number of trials performed
% stableFlag: flag indicating whether the simulation has
% stablized

% determine the number of trials per set
J=ceil(100/(1-p));
M=max(J,10000);

h=0;
AllSamples=[];

% get recorded true contact points from the virtual cmm object
trueContacts=vcmmObj.trueContact_recorder;

% set a max trial number, in case the simulation takes too long
% to stablize.
MaxTrial=300000;
```

```
stableFlag=0;

while M*h<MaxTrial
    h=h+1;
    emptyA=zeros(M,1);
    GroupSamples=cell(size(emptyA));

    for k=1:M
        vcmmObj.co_num=0;
        vcmmObj.co_rec={};

        for i=1:length(trueContacts)

            % get contact location
            probed_coord=trueContacts{i}(1:3);

            % get probing direction
            direction_vector=trueContacts{i}(4:6);

            % get probe error

            probeError=vcmmObj.error_simulator.getProbeError(vcmmObj.probe_L,
vcmmObj.probe_A, direction_vector);

            % get geometric error

            volError=vcmmObj.error_simulator.getVolumetricError(probed_coord);

            % get combined error

            combinedError=vcmmObj.error_simulator.getCombine_error(volError,probeE
rror);

            % get point with error
            errored_coord=probed_coord+combinedError;

            vcmmObj.co_recording(errored_coord);
        end

        % calculate the feature
        [msg, result]=vcmmObj.calc();
    end
end
```

```

        GroupSamples{k}=result;
    end

    samplesArr=obj.cell2matrix(GroupSamples);
    AllSamples=[AllSamples; samplesArr];

    % calculate results for hth set
    y(h)=mean(samplesArr(:,1));
    uy(h)=std(samplesArr(:,1));
    [ylow(h) yhigh(h)]=obj.getCI(samplesArr(:,1),p);

    % perform at least 2 sets
    if h==1
        continue;
    end

    % calculate the standard deviations associated with the h
    % sets of results
    sy=std(y);
    suy=std(uy);
    sylow=std(ylow);
    syhigh=std(yhigh);

    % use all valued obtained so far to calculate standard
    % uncertainty
    U=std(AllSamples(:,1));

    % calculate the numerical tolerance
    [coeff,exponent]=strread(strrep(sprintf('%E',U),'E','#'),'%f#%f');
    tolerance=0.5*10^(exponent-(nDig-1));

    % determine whether the simulation has stablized
    if sy*2>tolerance || suy*2>tolerance || sylow*2>tolerance ||
syhigh*2>tolerance
        continue;
    else
        stableFlag=1;
        break;
    end
end

totalTrial=M*h;

```



```
ya=mean(AllSamples(:,1));
uya=std(AllSamples(:,1));
[ylowa yhigha]=obj.getCI(AllSamples(:,1),p);
end
```

Function for the comparison of the measured results and the simulation results:

```
function [ std_mean_measure, std_mean_sim, plaus_counter,
ave_plaus_counter, mean_comp_v ] = cmp( sr, vr, temp )
% CMP compare the measurement results with the simulation results and plot
% figures
%
% Inputs:
% sr: the measured results of 30 measurements * 20 setups = 600 values
% vr: the results from simulation, 20 intervals
% temp: the recorded temperature for each setup

% Outputs:
% std_mean_measure: standard deviation of the mean values of measured
% results after correction for temperature
% std_mean_sim: standard deviation of the mean values of the simulation
% results
% plaus_counter: counter for plausible cases in each setup
% ave_plaus_counter: counter for plausible cases, testing mean value of
% results of each setup
% mean_comp_v: mean value of measured results after compensation for
% systematic effects

% thermal expansion coefficient for steel
k=11.7e-6;

% calibrated diameter
cd=38.13735;

% expanded uncertainty associated with the calibrated diameter
Ucal=0.00009;

% mean values of measured results after correction for temperature
```

```
mean_measure=zeros(1,20);

% mean values of the simulation results
mean_sim=zeros(1,20);

% counter for plausible cases in each setup
plaus_counter=zeros(1,20);

% counter for plausible cases, testing mean value of results of each setup
ave_plaus_counter=zeros(1,20);

% mean value of measured results after compensation for systematic effects
mean_comp_v=zeros(1,20);

for i=1:20
    figure(i);
    set(i, 'NumberTitle', 'off');
    set(i, 'Name', ['Setup No. ', num2str(i)]);
    set(i, 'Position', [300, 200, 375, 350]);

    % get temperature
    tp=temp(i,9);

    % get measured values
    mv=sr((i-1)*30+1):(i-1)*30+30,5);

    % corret measured values to 20 degrees celsius
    cv=mv/(1+k*(tp-20));

    mean_measure(i)=mean(cv);

    % get coverage interval
    meanValue=vr(i,3);
    LEnd=vr(i,5);
    REnd=vr(i,6);

    mean_sim(i)=meanValue;

    correction=cd-meanValue;

    % compensation for systematic effects
    comp_v=cv+correction;
```

```

% get expanded uncertainty
U=vr(i,4)*2;

% test for plausibility criterion
plaus_matrix=abs(comp_v-cd)/sqrt(Ucal^2+U^2)<=1;

% count plausible times
plaus_counter(i)=nnz(plaus_matrix);

mean_comp_v(i)=mean(comp_v);
if abs(mean_comp_v(i)-cd)/sqrt(Ucal^2+U^2)<=1
    ave_plaus_counter(i)=1;
end

hold on;

title(['Setup No. ', num2str(i)]);
xlabel('measurement');
ylabel('diameter (mm)');
xlim([0,31]);
ylim([38.129, 38.142]);

% plot the calibrated diameter 38.13735
plot(0:31, cd*ones(1,32), 'g','LineWidth',3)

% plot coverage interval
plot(0:31, meanValue*ones(1,32), '--r','LineWidth',3);
plot(0:31, LEnd*ones(1,32), '--k', 'LineWidth',3);
plot(0:31, REnd*ones(1,32), '--k', 'LineWidth',3);

% plot corrected measurement results
plot(1:30, cv', '-b*');

hold off;
end

figure(21);
set(21, 'NumberTitle', 'off');
set(21, 'Name', 'mean_measure VS mean_sim');
set(21, 'Position', [300, 200, 600, 400]);

```

```
hold on;

% plot the mean values of measured results vs the mean values of simulation
% results
plot(mean_measure, '-b*');
plot(mean_sim, '-ro');

hold off;

std_mean_measure=std(mean_measure);
std_mean_sim=std(mean_sim);

end
```

### Appendix C List of publications

Hu, Y., Yang, Q., Sun, X. & Wei, P., 2010. Applying semantic web services to enterprise web. *International Journal of Manufacturing Research, Special issue on Systems for Extended Enterprises*. Accepted.

Hu, Y., Yang, Q. & Wei, P., 2009. Development of a novel virtual coordinate measuring machine. In *IEEE Instrumentation and Measurement Technology Conference (I2MTC09)*. Singapore, 5-7 May 2009.

Wei, P., Yang, Q., Hu, Y. & Li, Y., 2009. Web-based evaluation of measurement uncertainty using symbolic computation. In *IEEE Instrumentation and Measurement Technology Conference (I2MTC09)*. Singapore, 5-7 May 2009.

Hu, Y., Yang, Q., Sun, X. & Wei, P., 2008. Applying semantic web services to enterprise web. In *the 6th International Conference on Manufacturing Research (ICMR08)*. London, UK, 9-11 September 2008.