

國立臺灣大學電機資訊學院電機工程學系



碩士論文

Department of Electrical Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

運用 7 軸 3D 測量機台對複雜工件之精準量測及驗證

3D Coordinate Measurement Machine with 7-DoF Configuration

for Precise On-Machine Inspection and Validation of Complex

Curvature Workpieces

李晟

Cheng Li

指導教授：羅仁權 博士

Advisor: Ren.C. Luo, Ph.D

中華民國 106 年 7 月

July 2017

誌謝



碩士班生涯是我嘗試研究生生活的一段美好回憶。感謝父母親辛苦的栽培與教誨，家人們給予我支持與鼓勵，讓我能夠全心全意接受挑戰，專心於我喜愛的領域中。非常感謝我最敬愛的指導老師羅仁權教授，提供我們豐富的資源，讓我可以親手打造理想中的機器人。從理念發想、設計、實驗、直到最後的成果展現，無論在哪個環節都是一生難得的體驗，我也從中學到非常多寶貴的知識與經驗。這一切都要感謝老師對我的指導及信任。兩年的師生互動中，老師充分展現出國際級學者的視野與風範，不僅帶領我們探索學術知識的廣度與深度，更是給予我們充分的機會去體驗本科系以外的經驗，希望我們成為均衡發展的人才。在待人處事方面，老師更是諄諄教誨，提醒我們許多社會上應對進退的準則。這一切都使我獲益良多，也確立我對於機器人領域的研究興趣，並決心朝此方向繼續努力。此外，我想特別感謝研究所時期教導過我的黃漢邦老師與陳明新老師，是他們為我打下機器人與控制理論的研究基礎，我從他們的課程中得到很多啟發。研究生生活中，感謝鏡文、陞佑、瑋隆、金成、獻章、繼棠、昕映、玲盈、志遠、東榕、正倫等博班學長姊，還有智賢、士紘、柏宏、邦甫、冠志、建偉、文謙、金博、耕銘、銘駿、煒森、建安等碩班學長姊，以及同屆的同學師兄俊豪、莉彤、長鈞、晴岡、靖霖、昱佑、達方、凱鈞、仲凱、柏凱，最後還有積極認真又可愛的學弟妹們：培淳、石巖、武昱、嵩詠、智堅、威辰、錦賢、育榕、育澤、名彥、展嘉、何鑫、曾旻、王昊，謝謝大家給我的幫助，在研究上面一起討論，同甘共苦的參加比賽，一起幫忙大大小小的事務，一起打球運動，這段時間真的很開心能夠跟大家一起經歷這麼多事情，相信這都是令人印象深刻的生活點滴。也很感謝助理群們，有你們幫忙處理繁瑣的事務，我們才能快速了解實驗室的情況，更專注於研究上。另外也很感謝一直保持連絡的好朋友們，有你們在的地方就有歡笑，再大的陰霾與你們聚聚之後都會煙消雲散，希望大家繼續保持聯絡。由衷的感謝大家，謝謝大家的幫助與支持，祝福大家平安快樂、一切順遂。

李晟 謹誌

一百零六年七月

中文摘要



近年來，CNC robot 不斷推陳出新，其功用之廣泛且穩定已使其成為在眾多機器人中成為相當具有潛力的一種，然而在一系列加工流程中，量測為一核心且極具影響力的項目，在許多在精度方面要求甚高的工業中，量測為期提供了品質保證、誤差判定的功效，如在航太業上，飛機機翼是一精度要求極高的大型加工件，且其獨特的設計考量了氣流、風力、風阻等多項因素，一點點加工上的誤差便會改變飛機在空中的穩定性，直接影響了上百人的安全，在小型加工件方面，如齒輪，些微的誤差都會容易造成齒輪旋轉過程中卡住，破壞機械結構使整台機器故障甚至報廢，由已上便可知道量測在加工中是如此重要。此篇碩論提出一就由裝置高精度探針的多軸量測機來完成對複雜曲面之量測。

多自由度之測量機台搭配冗餘度的特性後可在空間中有著相當大的工作範圍及自由性，其末端點可以在位置不動的情況下改變旋轉以此來確定末端點之探針探測工件表面的點做量測時可以保證其以法向量觸碰，藉此得到準確的量測數據，大多數的測量機台由於沒有多自由度之特性，常常在以非法向量的方式量測後再額外進行計算出真正的觸碰點位置，此方法的缺點為由於無法真正準確地知道觸碰點與探針末端點的關係，導致難以準確補償物差造成精確度下降。而在量測技術方面，對於一有複雜曲面之工件的量測往往會由於缺乏參考點而無法找出量測點，此篇碩論中一提出一針對複雜曲面之量測流程，其包括之內容有物件定位、探針半徑補償、曲面量測點取樣、曲面重建、誤差分析及曲面精度分析等。

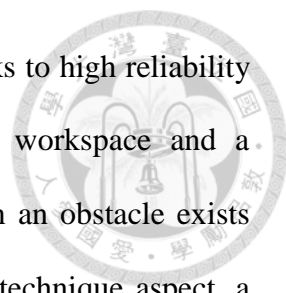
關鍵字： CNC 加工機、量測系統、曲面量測、探針、量測機台

ABSTRACT



For these years, CNC(Computer Numerical Control) machine has been developed at an incredibly high speed. Because of its high stability and wide applicability, this kind of machine is considered as the most powerful potential in the future development. In the series of machining procedure, measurement is a core and certainly influential item. For many industries which require extremely high precision of a workpiece, measurement provides the insurance of quality and error decision for each workpiece. Take aerospace industry for example, the airfoil is a large machined part which requires extremely high precision, which is specially designed that considers a lot of factors such as air flow, air mechanics, windage, and so on. Any small error may incredibly influence the characteristics above and decrease the flying stability. Another great example is gear. Stuck is happened while the precision cannot be promised, which may furthermore destroy the whole equipment. This paper proposed a method for measuring a complex surface by a precise probing system mounted on the end-effector of measurement machine with high DoF.

For some measuring strategy, they use different sides of the end-effector to measure the points because of the limitation of lower degree of freedom. The merit of the strategy is effectively reflected on lower cost of hardware. However, the difficulty is increased because extra computation is needed for points with inaccessible orientation and position. What's more, since the error of the end-effector is inevitably affects the real position and command position, the accuracy of measuring value must be much lower. This paper presents an algorithm to explain how to get the normal vector of each point located on the surface of a workpiece and how to avoid the dangerous points which may lead to collision during measuring is presented as well. Finally, the



algorithm is implemented on a 7-DoF measurement machine. Thanks to high reliability and high degree of freedom, the machine has extremely wide workspace and a redundant angle to make the end-effector can reach the target even an obstacle exists that may disturb the path of the manipulator. In the measurement technique aspect, a series of measurement procedure will be introduced and explained, including workpiece localization, probe radius compensation, complex surface sampling, surface reconstruction, error analysis and precision analysis of real workpiece.

Keywords: CNC machine, measurement system, surface validation, probe, Coordinate measurement machine

CONTENTS

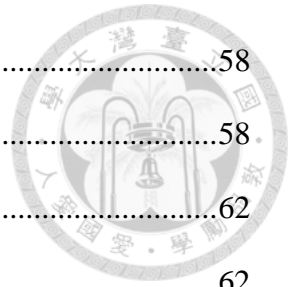


誌謝	i
中文摘要	ii
ABSTRACT	iii
CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xi
Chapter 1 Introduction.....	1
1.1 Era of Coordinate Measurement Machine	1
1.2 Freeform Surface Inspection and Validation	5
1.3 CAD Model Representation	6
1.3.1 IGES File Specification.....	6
1.3.2 IGES File Specification.....	7
1.4 Proposed Measurement and Validation Procedure	8
Chapter 2 Scenario.....	9
2.1 Scenario Introduction.....	9
2.2 Coordinate Measurement Machine in iCeIRA	10
2.2.1 Mechanism Design of Multi-axis CNC Robot.....	10
2.2.2 Anti-collision sensor and Home sensor.....	15
2.2.3 High accuracy probe	16
2.3 Measured Workpiece	17
2.4 Geometrical Data Extraction	18
2.4.1 NURBS Specification	18



2.4.2	Data Extraction from CAD model	19
2.4.3	Geometrical Characteristics Computation	21
2.5	Simulation Workspace	24
Chapter 3	CMM Kinematics	26
3.1	Spatial Descriptions and Transformation	26
3.1.1	Three-Angle Representation	27
3.1.2	Angle–Axis Representation	28
3.1.3	Unit Quaternions	29
3.2	Differential Motion	32
3.3	CMM Kinematics	34
3.3.1	Forward Kinematics of CMM	34
3.3.2	Velocity Relationship: The Manipulator Jacobian	36
3.4	Forward Kinematics Model of iCeIRA CNC robot	38
3.5	Inverse Kinematics Model of iCeIRA CNC robot	40
3.5.1	Numerical Solution - The Jacobian Transpose Method	40
3.5.2	Analytic Solution – Closed-Form Solution	41
3.6	Linear Blending Motion	44
Chapter 4	Workpiece Registration	45
4.1	Description	45
4.2	Reference-based Registration	47
4.3	Algorithm-based Registration	51
Chapter 5	Freeform Surface Measurement and Validation	55
5.1	Concept of Proposed Measurement Process	55
5.2	Measured point generation	56
5.3	Working Consideration	57

5.3.1	Joint Limit Consideration.....	58
5.3.2	Collision Detection.....	58
5.4	Measurement Trajectory Generation	62
5.4.1	Probing Posture Determination	62
5.4.2	Initial Trajectory generation	64
5.4.3	Joint Limit Response.....	67
5.4.4	Collision Response.....	69
5.5	Error Consideration	72
5.5.1	Pretraveling problem.....	73
5.5.2	Radius Compensation.....	74
5.6	Surface Validation.....	75
5.6.1	Localization Uncertainties	77
5.6.2	Reconstruction Uncertainty.....	79
5.6.3	Surface Validation	80
Chapter 6	Experiment	82
6.1	Localization Experiment.....	82
6.2	Freeform Surface Measurement Experiment.....	84
6.3	Data Analysis and Surface Validation.....	87
Chapter 7	Conclusion	90
	REFERENCE	91
	VITA.....	95



LIST OF FIGURES



Fig. 1-1 Common types of CMM	3
Fig. 1-2 Inner structure of a standard probe	4
Fig. 1-3 sensors or peripherals for CMM	5
Fig. 1-4 IGES file example.....	7
Fig. 1-5 STL object example	8
Fig. 1-6 proposed freeform surface measurement and validation process	8
Fig. 2-1 Scenario of measurement task	9
Fig. 2-2 Gantry-type machine and manipulator.....	10
Fig. 2-3 The CAD diagram of the multi-axis machine tool with gantry-type and the its axis definition.	11
Fig. 2-4 Engineering drawing of AC servo motor	12
Fig. 2-5 The Engineering drawing of the reducer which is manufactured by APEX Inc.	14
Fig. 2-6 slot-type photomicrosensor	15
Fig. 2-7 Integration of a high accuracy probe into CNC robot to reach the functionality of CMM	16
Fig. 2-8 measured workpiece.....	17
Fig. 2-9 NURBS surface onto the workpiece	20
Fig. 2-10 NURBS surface in mesh-type	21
Fig. 2-11 Normal vector of the designed surface	22
Fig. 2-12 Principal direction of a point.....	23
Fig. 2-13 Gaussian curvature for the surface.....	24
Fig. 2-14 Simulation workspace of real environment	25

Fig. 3-1	the algorithm pseudo code of computing a quaternion convert from a rotation matrix	31
Fig. 3-2	definition of standard D-H parameters	35
Fig. 3-3	the assigned DH link frames and	38
Fig. 3-4	The procedure of numerical method for solving IK	41
Fig. 3-5	The IK result working on an irregular path	43
Fig. 3-6	Linear blending motion.....	44
Fig. 4-1	Workpiece Registration.....	46
Fig. 4-2	Effect of inaccurate localization	47
Fig. 4-3	Obtain normal vector of the surface.....	48
Fig. 4-4	normal vectors of the referenced surface	49
Fig. 4-5	localize the translation of WCS by probing strategy	51
Fig. 4-6	schematic diagram of ICP algorithm	52
Fig. 4-7	Result of geometrically stable sampling for ICP algorithm.....	54
Fig. 5-1	Schematic diagram of the measured point generator	57
Fig. 5-2	Skeleton of the iCeIRA CNC robot 3D model.....	59
Fig. 5-3	Skeleton of the iCeIRA CNC robot.....	60
Fig. 5-4	example of collision detection where the collision happen on Link 2.....	61
Fig. 5-5	Skeleton of the iCeIRA CNC robot.....	62
Fig. 5-6	Path with optimized posture determination	63
Fig. 5-7	Trajectory planning with control points.....	64
Fig. 5-8	The four Hermite basis functions.....	65
Fig. 5-9	Initial measurement path.....	66
Fig. 5-10	CMM motion on initial path	66
Fig. 5-11	Interpolation problem of joints for orientation part	67

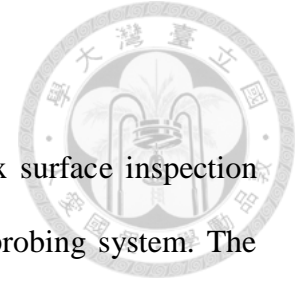
Fig. 5-12 Path modified by joint limit	68
Fig. 5-13 CMM motion on path with joint limit response	68
Fig. 5-14 Example of workspace to C-space for a 2D translating robot	70
Fig. 5-15 Simplified measured surface which is used for collision avoidance	70
Fig. 5-16 C-space example for a simple 2DoFs robot manipulator.....	71
Fig. 5-17 Path modified with consideration of joint limit and collision	71
Fig. 5-18 CMM motion on path with joint limit and collision response	72
Fig. 5-19 Error in the measurement process.....	73
Fig. 5-20 schematic diagram of pretraveling problem	74
Fig. 5-21 schematic diagram of radius compensation	75
Fig. 5-22 schematic diagram Error caused by different reasons	76
Fig. 5-23 Freeform surface inspection considering the uncertainties in measurement, localization and surface reconstruction	76
Fig. 5-24 Reconstruction error and localization error	79
Fig. 6-1 Implementation of reference-based localization.....	83
Fig. 6-2 Implementation of freeform surface measurement	84
Fig. 6-3 The first reconstructed surface compared with CAD model	85
Fig. 6-4 Surface reconstruction result with different number of measured points	85
Fig. 6-5 The final reconstructed surface compared with CAD model.....	86
Fig. 6-6 Error descent of reconstruction error	86
Fig. 6-7 The resultant error distribution	87
Fig. 6-8 3D printer used to manufacture the workpiece	89

LIST OF TABLES



Table 2-1 Specifications of raw, pitch and yaw	13
Table 3-1 link parameter of iCeIRA CNC robot	39
Table 5-1 Joint Limit of iCeIRA CNC robot	58
Table 6-1 Localization Result.....	83

Chapter 1 Introduction



This thesis main topic is a measurement process for complex surface inspection which works on a 7-DoF coordinate measurement machine with probing system. The principal and method will be figured out in the subsections below. Actually, the ideal of the main topic is gradually constructed by the experience of developing the robot control system and useful functions during my master career. The overview of this thesis and the simplified control scheme of the coordinate measurement system will be shown in this chapter.

First, the era of CMM is in section 1.1 and then section 1.2 introduces the task we would like to reach and some detail about the freeform surface measurement. Besides, the reason why we choose freeform surface as measured object is explained. Second, datatype and detail about the 3D CAD model are listed in section 1.3, including the most common one in industry, IGES, and the one for 3D printer, STL. Then the measurement procedure we purposed in this thesis is shown in section 1.4.

1.1 Era of Coordinate Measurement Machine

As the industry grows up rapidly in these years, accuracy of a workpiece is an important issue that more and more researches pay attention on. For a real workpiece fabricated from the computer numerical control (CNC) machines, there must exists error compared with the computer aided design (CAD) model. However, for some special industry such as aerospace industry, the accuracy of a workpiece is necessary to be promised. Therefore, coordinate measurement machines (CMMs) are utilized to validate the extent of difference between a real workpiece and its corresponding CAD model.

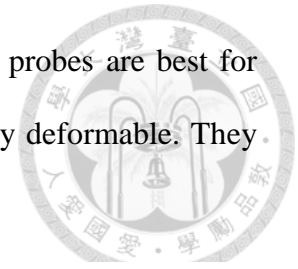
For a measurement task, the core technique is to trigger the probing system mounted on the end-effector of CMMs, so that the rapid feedback is able to return

present coordinate instantly. The typical 3D "bridge" CMM is composed of three axes, X, Y and Z. These axes are orthogonal to each other in a typical three-dimensional coordinate system. Each axis has a scale system that indicates the location of that axis. The machine reads the input from the touch probe, as directed by the operator or programmer. The machine then uses the X, Y, Z coordinates of each of these points to determine size and position with micrometer precision typically. By precisely recording the X, Y, and Z coordinates of the target, the accuracy of a workpiece can be measured and analyzed via regression algorithm for the construction of features.

In Fig. 1-1, three basic types of measurement machines are listed: (a) Bridge CMM is well known for its stability. And it provides cost-effective solution for dimensional quality control, (b) Gantry CMM is suited to inspection for large workpieces because synchronous gantry control leads to high speed movement. It can achieve larger workspace for measurement and efficient performance, (c) Horizontal Arm CMM is the most flexible type. The advantage is particular apparent when inspecting sheet metal parts in the car industry, or other large-volume components in the aerospace, ship, defense industries. Its open structure permits direct access to the workpiece and therefore significantly eases loading and unloading. (d) Shop floor conditions may suggest that accurate measurement results on the factory floor would be impossible to obtain. However, using specialized shop floor hardened CMM, measurement accuracy is ensured even in the production environment.

Some researchers and companies tried to develop different sensors to reach different goals. Mainly the probe is the most popular sensor and they can be mainly divided into 2 classes, touch trigger type and non-contact type. As Fig. 1-2 shown, touch trigger probes consist of a stylus attached to a bearing plate connected to pressure sensors inside the probe housing. They generate an electrical signal each time they

contact a point on the workpiece. On the other hand, Non-contact probes are best for parts that are either more complex, smaller, high-precision or easily deformable. They are either laser-based or vision-based.



(a)



(b)



(b)



(d)

Fig. 1-1 Common types of CMM

(a) NIKON ALTO Bridge CMM (b) Perceptron COORD3 MCT Gantry CMM (c) Hexagon DEA MERCURY Horizontal Arm CMM (d) Hexagon TIGO shop floor CMM

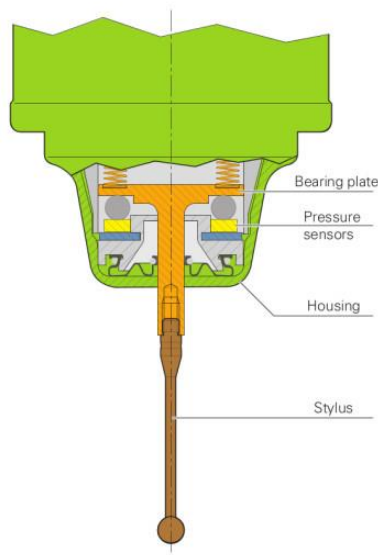


Fig. 1-2 Inner structure of a standard probe

Take some useful sensors or peripherals for example: (a) ultra-compact probe features strain gauge technology. It delivers unrivalled submicron performance when applied to complex 3D shapes and contours, (b) probe head provides extra degree of freedom, which provides larger workspace and decrease time-cost furthermore. Besides, with adaptive control, the touch trigger probe is able to acquire data quickly by using “sliding” touch onto a surface. (c) laser sensor uses a concentrated beam of light instead of a stylus. The beam acts as an optical switch so that when it is projected onto the part, the position can then be read by triangulation through a lens in the probe receptor. (d) For vision sensor, the 2D light sensitive CCD and the integral LED illumination allow an area of the measured component to be illuminated and viewed. All the products mentioned above are listed in Fig. 1-3.

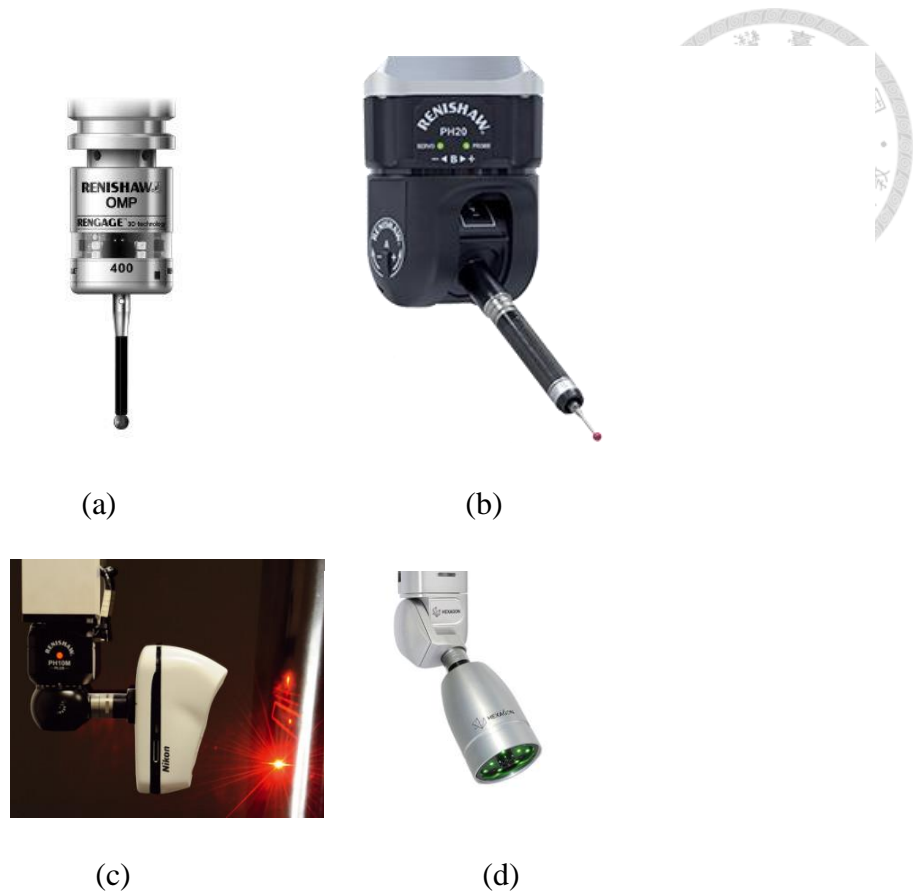


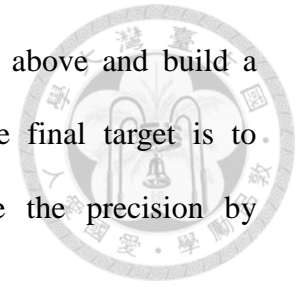
Fig. 1-3 sensors or peripherals for CMM

(a) Renishaw OMP400 probe (b) Renishaw PH20 probe head (c) Nikon InSight L100 laser scanner. (d) Hexagon HP-C-VE vision sensor

1.2 Freeform Surface Inspection and Validation

It's undeniable that the freeform surface inspection technique is thought to be one of the most valuable challenges in of measurement applications. Because of complex structure and irregular shape, it's very hard to decide measured points by human intuitive. Therefore the geometrical characteristics of a surface should be analyzed to sample out to be a point set which can fully describe the whole surface. However, from geometrical data extraction, measured points sampling, trajectory planning, probing raw data to workpiece validation, each step occupies significantly important standing and

has its own consideration. This thesis tries to solve the problem above and build a complete inspection system for complex surface inspection. The final target is to evaluate a real workpiece with complex surface, and validate the precision by comparing the measured data with CAD model.



1.3 CAD Model Representation

In this thesis, the CAD model of measured object is presented in IGES format. All geometrical data is extracted from the file and do some geometrical characteristics analysis. Besides, the mesh-type data format, STL format, is still used to describe other geometrical properties in the measurement process. Both the data type is briefly introduced below.

1.3.1 IGES File Specification

The Initial Graphics Exchange Specification (IGES) outlines a file format for the transfer of geometry data and CAD models. It can be used to represent both Boundary-Representation (B-Rep) and Constructive Solid Geometry (CSG) geometries, as well as two dimensional CAD diagrams. The file is an ASCII text based format, having each line be exactly 80 characters long. The file is split into five sections, denoted by the specific upper case letter in the 73rd column. Those sections are Start (S), Global (G), Data Entry (D), Parameter Data (P), and Terminate (T) sections. The Data Entry and Parameter Data sections are commonly abbreviated Data Entity (DE) and Parameter Data (PD), respectively. For DE, there are around 150 different defined entities in IGES, and the corresponding PD gives the geometrical presentation for the DE. Here gives a simple example of 2D geometry with its IGES file as shown in Fig. 1-4.

1H,,1H;,4HSL0T,37H\$1\$DUA2:[IGESLIB.BDRAFT.B2I]SLOT.IGS;;	S	1	Global & Start
17HBravo3 BravoDRAFT,31HBravo3->IGES V3.002 (02-Oct-87),32,38,6,38,15,	G	1	
4HSL0T,1.,1,4HINCH,8,0.08,13H871006.192927,1.E-06,6.,	G	2	
31HD. A. Harrod, Tel. 313/995-6333,24HAPPLICON - Ann Arbor, MI,4,0;	G	3	
116 1 0 1 0 0 0 0 0 0 1D	1D	4	Data Entry
116 1 5 1 0 0 0 0 0 0 0D	0D	1	
116 2 0 1 0 0 0 0 0 0 1D	1D	2	
116 1 5 1 0 0 0 0 0 0 0D	0D	3	
100 3 0 1 0 0 0 0 0 0 1D	1D	4	
100 1 2 1 0 0 0 0 0 0 0D	0D	5	
100 4 0 1 0 0 0 0 0 0 1D	1D	6	
100 1 2 1 0 0 0 0 0 0 0D	0D	7	
110 5 0 1 0 0 0 0 0 0 1D	1D	8	
110 1 3 1 0 0 0 0 0 0 0D	0D	9	
110 6 0 1 0 0 0 0 0 0 1D	1D	10	
110 1 3 1 0 0 0 0 0 0 0D	0D	11	
116,0.,0.,0.,0.,0,0,0;	1P	12	Parameter Data
116,5.,0.,0.,0.,0,0,0;	3P	1	
100,0.,0.,0.,0.,1.,0.,-1.,0,0;	5P	2	
100,0.,5.,0.,5.,-1.,5.,1.,0,0;	7P	3	
110,0.,-1.,0.,5.,-1.,0.,0,0;	9P	4	
110,0.,1.,0.,5.,1.,0.,0,0;	11P	5	
S 1G 4D 12P 6	T	6	Terminate

(a)

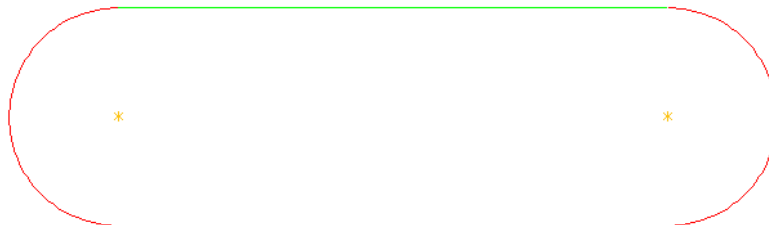


Fig. 1-4 IGES file example

(a) IGES content (b) geometry presented by (a)

1.3.2 IGES File Specification

An STL file is a triangular representation of a 3D surface geometry. The surface is tessellated logically into a set of oriented triangles (facets). Each facet is described by the unit outward normal and three points listed in counterclockwise order representing the vertices of the triangle. While the aspect ratio and orientation of individual facets is governed by the surface curvature, the size of the facets is driven by the tolerance controlling the quality of the surface representation in terms of the distance of the facets from the surface. A STL file can be presented in ASCII or in binary form. A simple

example of a STL object is shown in Fig. 1-5.

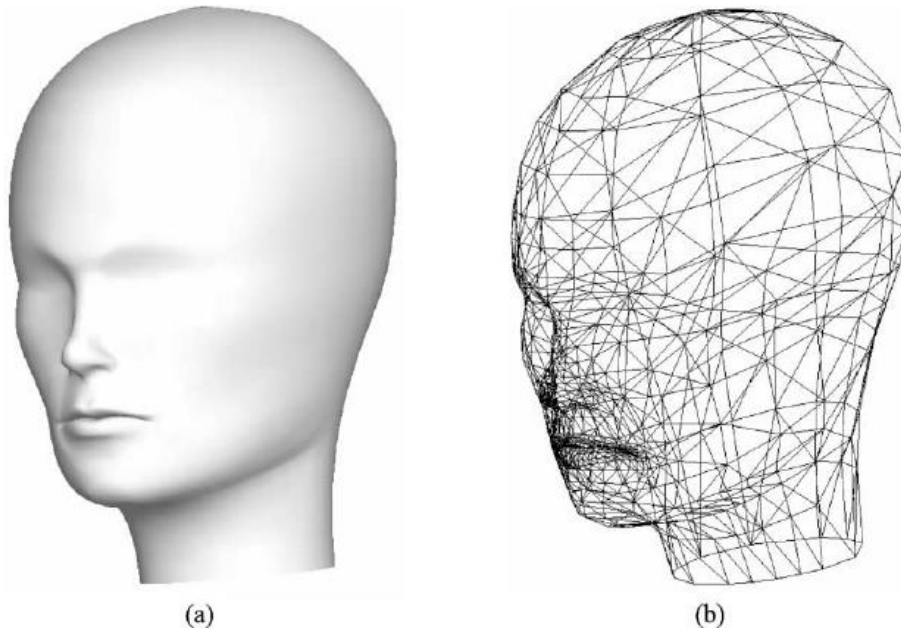


Fig. 1-5 STL object example

(a) CAD model (b) STL file of (a)

1.4 Proposed Measurement and Validation Procedure

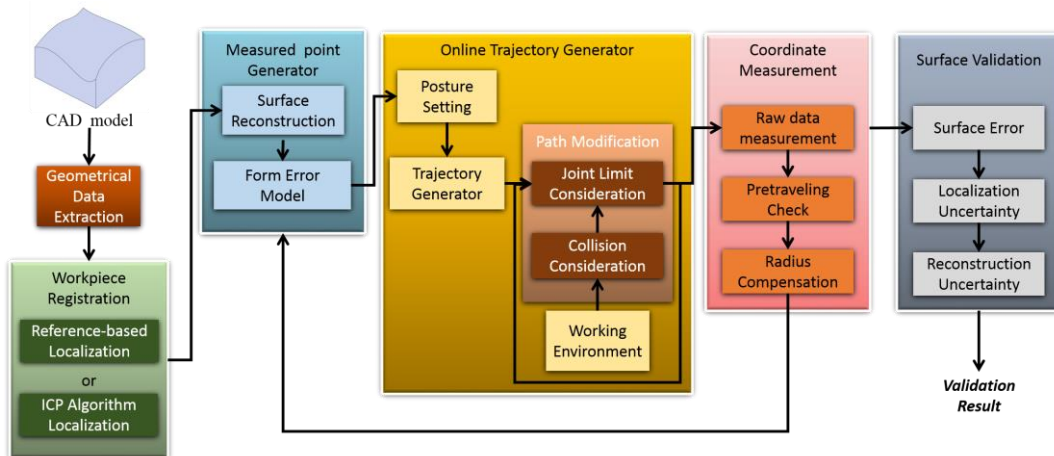


Fig. 1-6 proposed freeform surface measurement and validation process

Chapter 2 Scenario

2.1 Scenario Introduction

In this thesis, a measured workpiece with complex surface is placed onto the backside of CMM platform, where there still exist other objects on the platform. These objects are not measured target in this thesis so that we take them as obstacles in the workspace. The CMM we use in this thesis is a CNC robot with 7 axes and 6 degree of freedom (DoF). A touch trigger probe is mounted on the end-effector to probe the measured points and obtain coordinate data feedback. Each part mentioned above will be introduced in the following sections. The scenario is shown in Fig. 2-1.

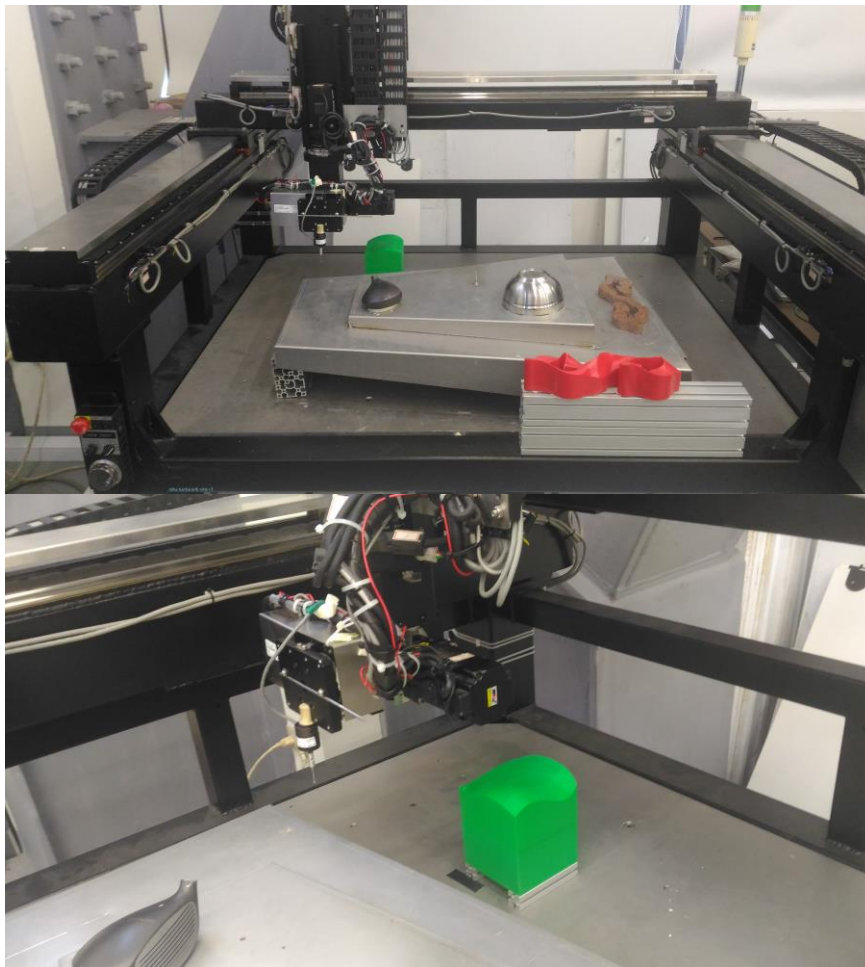


Fig. 2-1 Scenario of measurement task

2.2 Coordinate Measurement Machine in iCeIRA



We combine manipulator and gantry-type CNC machine tool to develop a new page in the machining field. Robot with gantry-type machine tool has many advantages like high stiffness, high accuracy and easy to calculate its coordinate system, for example, Fig. 2-2(a). Robot arm is connected by joints that allowing either translational (linear) or rotational motion as shown in Fig. 2-2(b). Combining the advantages of robot arm and gantry-type CNC machine tool has extra joints to apply and has more workspace than traditional type. The CNC robot is also designed and implemented in iCeIRA laboratory.



(a) (b)
Fig. 2-2 Gantry-type machine and manipulator

(a) Gantry-type machine tool with linear motor. (b) Multi-axis robot arm.

2.2.1 Mechanism Design of Multi-axis CNC Robot

We build a multi-axis robot with gantry-type machine tool to implement the inspection experiment and our algorithm. The outward appearance is shown in Fig. 2-3. It totally has 7 motors and 6 degree of freedom. It consists of three degrees of freedom

at linear axis including X, Y and Z axis to control translation and three rotational axes which orthogonal to each other at the manipulator to control orientation. The x and y axes are using linear motor. The z axis and other three rotational axes are using AC servo motor which is manufactured by Delta Electronics as figure, Inc. The CAD diagram and specifications of AC servo motor and reduction ratio are shown in Fig. 2-4 and Fig. 2-5.

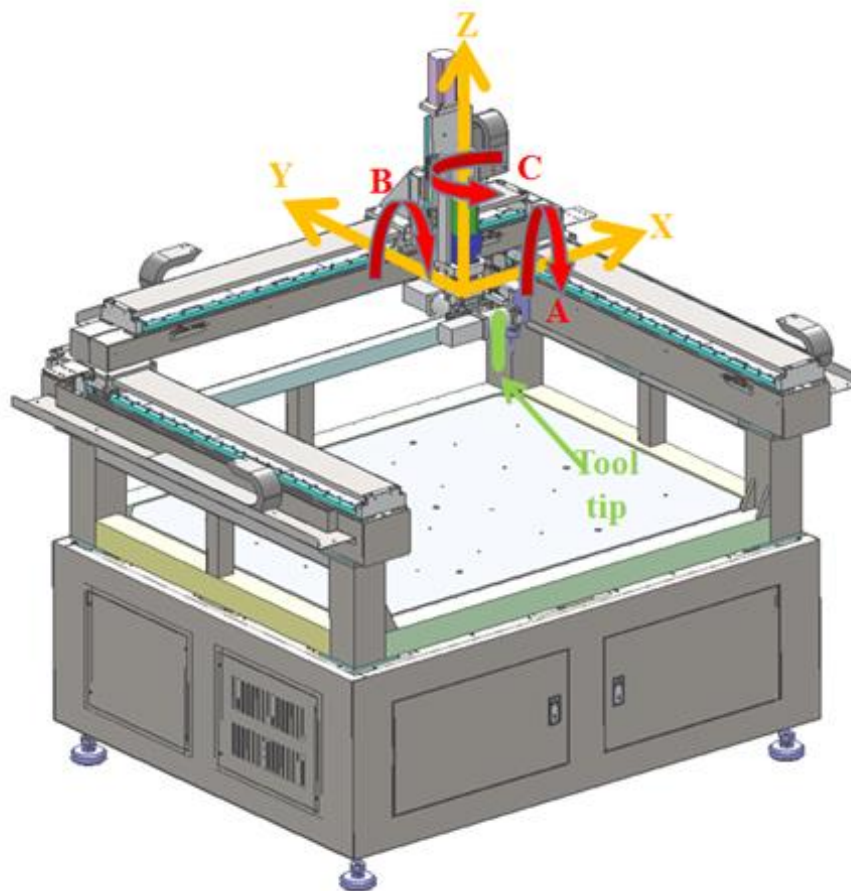


Fig. 2-3 The CAD diagram of the multi-axis machine tool with gantry-type and the its axis definition.

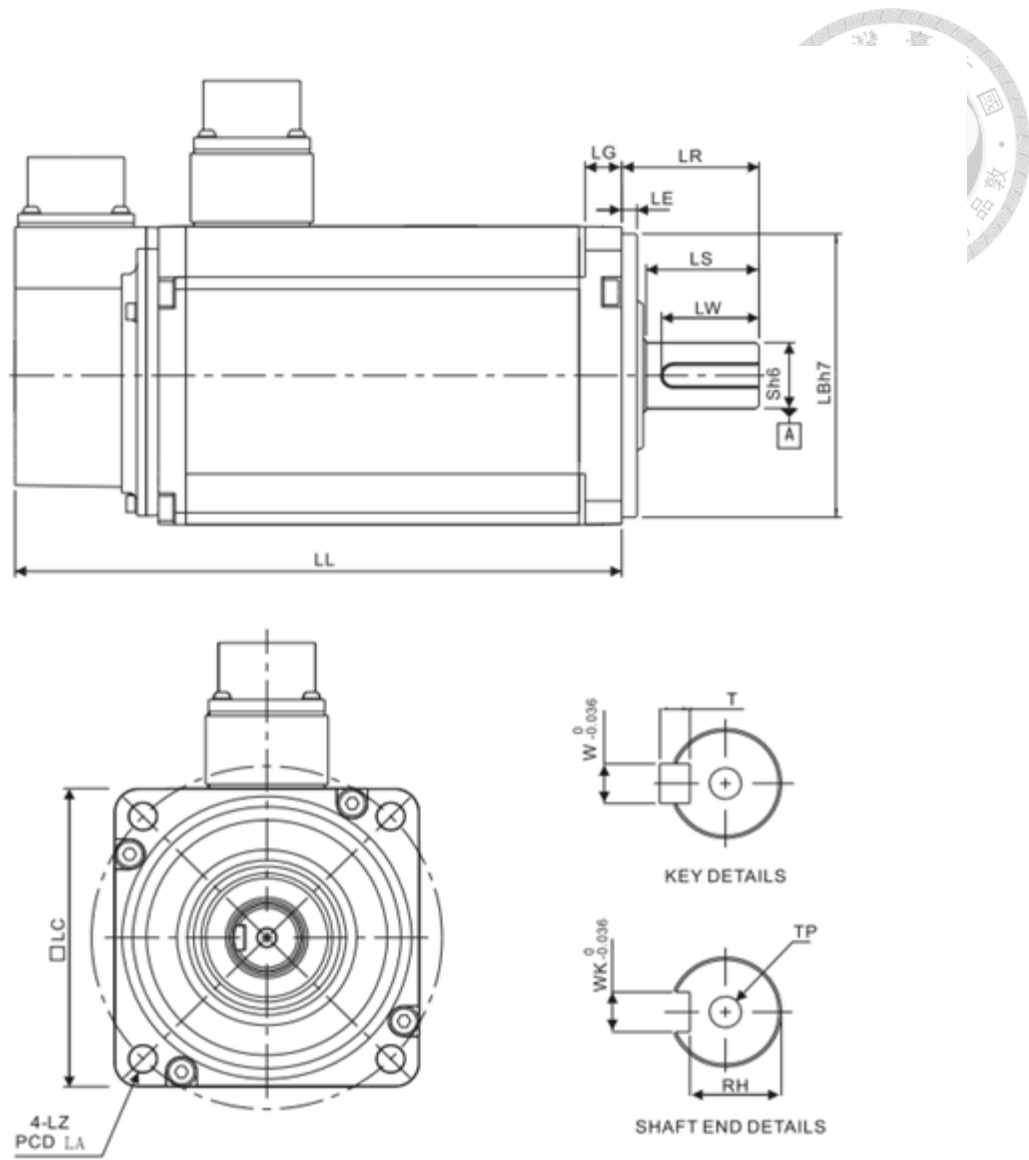


Fig. 2-4 Engineering drawing of AC servo motor

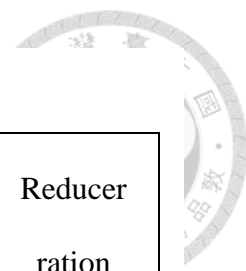


Table 2-1 Specifications of raw, pitch and yaw.

Joint	Rated power (kW)	Rated torque (N-m)	Maximum Torque (N-m)	Rated rotating speed (r/min)	Reducer ration
Raw	0.1	0.64	1.92	3000	10
Pitch					
Yaw					
Z					X

The diagram of reducer is shown in Fig. 2-5. The other characteristic is that most of crucial components like reducer, AC servo motor and screw are made in Taiwan. We want to convey a concept that we can build up a series outstanding machine tool industry by ourselves. We can obtain good effect by using planetary reduction gear and cost lower price than using harmonic drive.

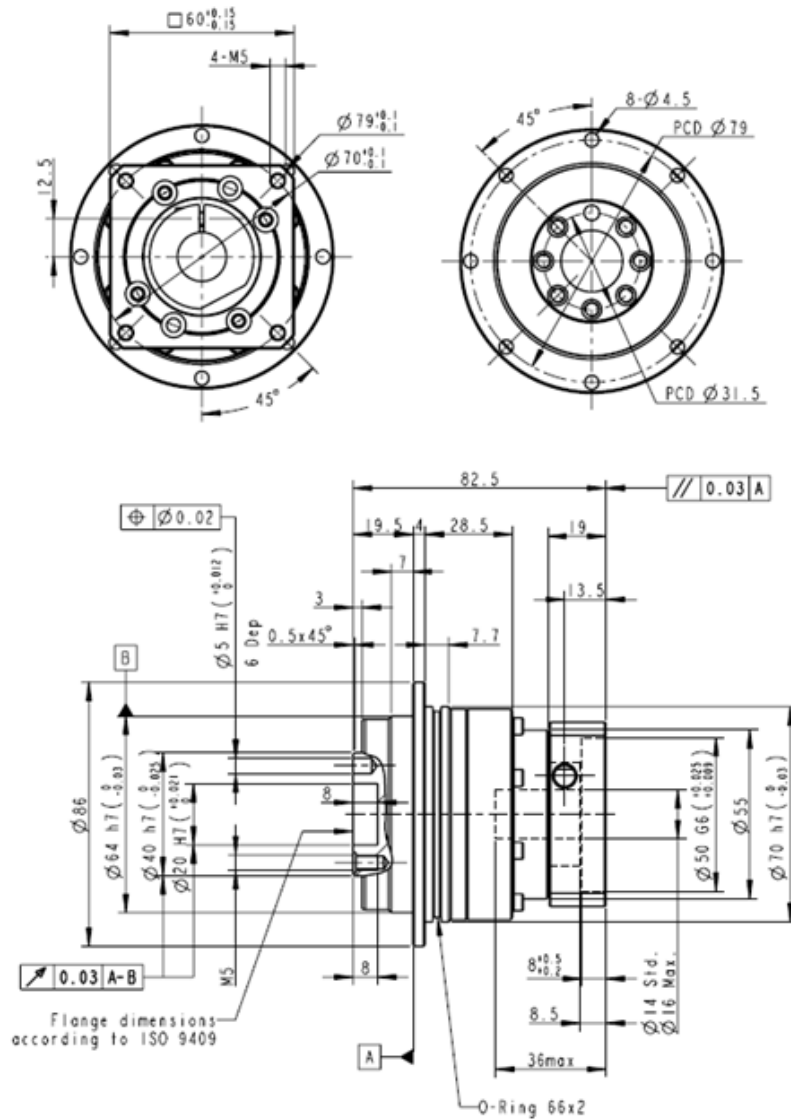
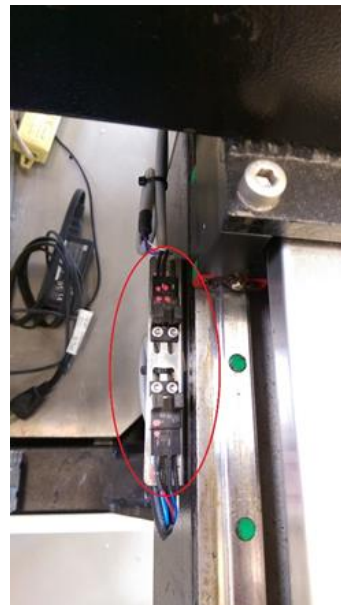
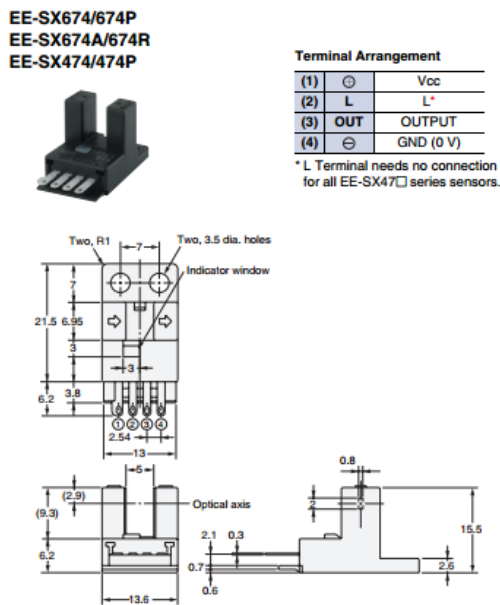


Fig. 2-5 The Engineering drawing of the reducer which is manufactured by APEX



2.2.2 Anti-collision sensor and Home sensor

In order to avoid the motors collision or exceed the workspace, CNC robot is equipped with several sensors at every axis. Besides, home sensors are also mounted to define zero point for each axis. All sensors are the same slot-type photomicrosensor, which works as optical switches to control the signal output. The specification is shown in Fig. 2-6(a). For X and Y axes, two sensors are equipped at the end of linear motor for defining the boundary for each axis. Besides, a home sensor is mounted next to one boundary sensor to set the zero point. The remained axes, Z, A, B and C, have home sensors only. Fig. 2-6(b) shows the sensors located on Y axis.



(a)

(b)

Fig. 2-6 slot-type photomicrosensor

(a) Specification of sensor (b) location of sensors on Y axis

2.2.3 High accuracy probe

We integrate a high accuracy probe into CNC robot to be a second end-effector, so that a coordinate measuring machine (CMM) that we will mention behind the conclusion is born. The illustration is shown in Fig. 2-7. The probe has a precision ruby tip for reduced wear and increased accuracy. The shank is 3/8 inch diameter, to accommodate the capacities of the small desktop mills.

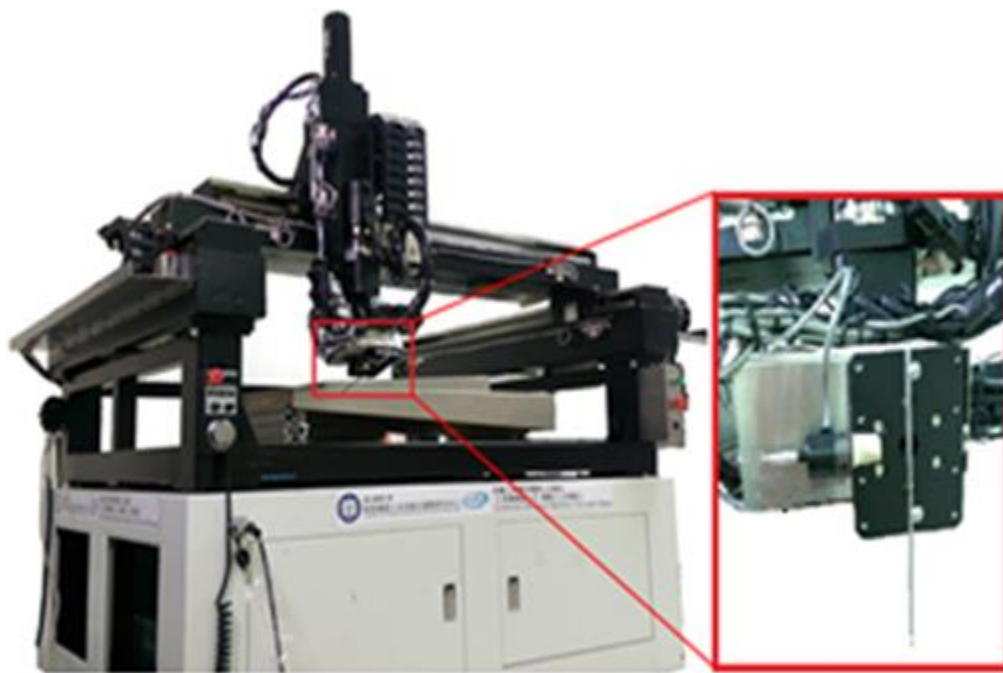


Fig. 2-7 Integration of a high accuracy probe into CNC robot to reach the functionality of CMM



2.3 Measured Workpiece

To build a complex surface to validate, we need to design a workpiece with some properties. First, any evident characteristics that can be identified by human intuitive is not appeared. Since an obvious feature can be easily utilized to infer measured points, which is not fitting to our purpose in this thesis. Second, the surface should be smooth enough, which means that every point onto the surface should be continuous. This condition is applied to fit the definition of “curvatures” surface. The condition is achieved by a continuous mathematic representation. Third, the surface should be bounded into a finite range. The largest range should not exceed the workspace of CMM. On the basis of the restriction above, we utilized the famous CAD software, SolidWork, to design the measured workpiece. The complex surface onto the surface is randomly tuned as the Fig. 2-8(a) and the real object is fabricate by 3DP as shown in Fig. 2-8(b).

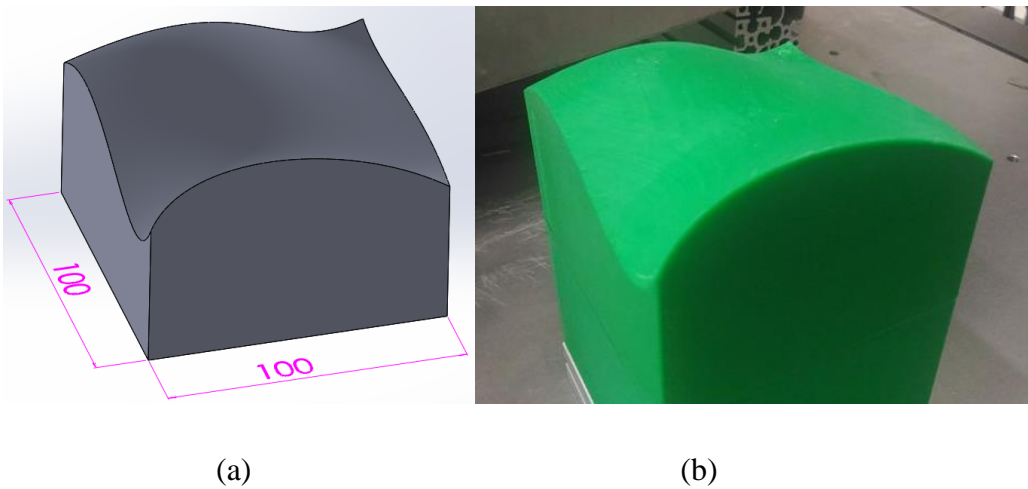


Fig. 2-8 measured workpiece

(a) CAD model of measured workpiece (b) real object

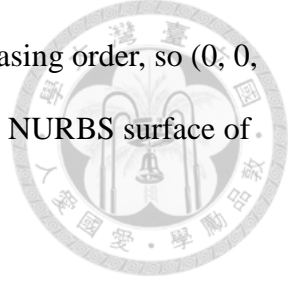
2.4 Geometrical Data Extraction



Before inspection process is implemented on the measured surface, we need to extract some geometrical data from CAD model, which is also the first step in the proposed measurement procedure. The data is used to be material for inspection process. The extraction method mainly depends on the characteristics and mathematical model of the surface. For our workpiece, the surface is constructed by Non-uniform Rational B-Splines (NURBS) which is a mathematical model commonly used in computer graphics for generating curves or surfaces. The feature of this model is that it can be efficiently handled by computer programs and yet allow for easy human inter action. Moreover, for a surface composed of several NURBS patches, the geometric continuity is promised perfectly with different levels including positional continuity (G^0), tangential continuity (G^1), curvature continuity (G^2).

2.4.1 NURBS Specification

A NURBS curve is defined by its order, a set of weighted control points, and a knot vector. NURBS curves and surfaces are generalizations of both B-splines and Bézier curves and surfaces, the primary difference being the weighting of the control points, which makes NURBS curves "rational". The Control points determine the shape of the curve. Typically, each point of the curve is computed by taking a weighted sum of a number of control points. For a curve of k degree, the weight of any control point is only nonzero in $k+1$ intervals of the parameter space. The knot vector is a sequence of parameter values that determines where and how the control points affect the NURBS curve. The number of knots is always equal to the number of control points plus curve



degree plus one. The values in the knot vector should be in nondecreasing order, so (0, 0, 1, 2, 3, 3) is valid while (0, 0, 2, 1, 3, 3) is not. The general form of a NURBS surface of degree (p,q) is shown.

$$S(u,v) = \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u)N_{j,q}(v)w_{i,j}} \quad (2-1)$$

where $N_{i,p}$ and $N_{j,q}$ are the B-Spline basis function, $P_{i,j}$ are control points, and the weighted $w_{i,j}$ of $P_{i,j}$ is the last ordinate of the homogeneous point $P_{i,j}^w$. For a B-Spline generalization, the basis function is

$$N_{i,0}(t) = \begin{cases} 1, & \text{if } t_i \leq t < t_{i+1} \text{ and } t_i < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (2-2)$$

$$N_{i,j}(t) = \frac{t-t_i}{t_{i+j}-t_i} N_{i,j-1}(t) + \frac{t_{i+j+1}-t}{t_{i+j+1}-t_{i+1}} N_{i+1,j-1}(t) \quad (2-3)$$

where $T = \{t_0, t_1, \dots, t_m\}$ is the set of knot vector with $t_i \in [0,1]$, and define the control points P_0, \dots, P_n . Define the degree as

$$p \equiv m - n - 1 \quad (2-4)$$

The “knot” $t_{p+1}, \dots, t_{m-p-1}$ are called internal knots.

2.4.2 Data Extraction from CAD model

In our case, the workpiece is designed by famous CAD software, SolidWorks. In the beginning, we design a cube with $100mm \times 100mm \times 100mm$ then the top face of the cube is deformed randomly by the built-in function in SolidWorks. The CAD model is exported in IGES format and we use the MATLAB to analyze and extract the data by IGES toolbox. The toolbox offers a lot of convenient functions to build a complete database for MATLAB, including analysis, modification, deletion, and so on.

At first, the IGES file is imported in the MATLAB by the function “iges2matlab”, and then we try to extract the data onto the surface. By the definition of NURBS, we set the size of knot vector set in both parametric directions to be 200. Therefore the 200×200 points onto the surface can be obtained as the Fig. 2-9.

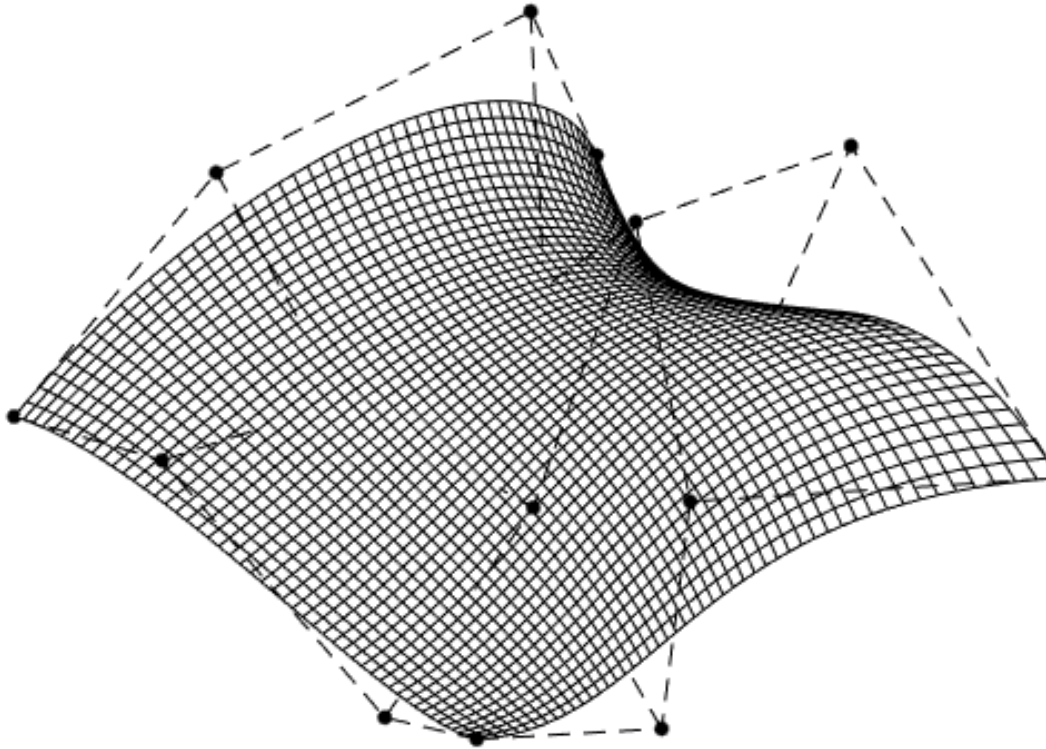


Fig. 2-9 NURBS surface onto the workpiece

The points set are denoted as S_p in our program, of which size is $200 \times 200 \times 3$. The points can also be defined as

$$S(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix}, (u_1 \leq u \leq u_N, v_1 \leq v \leq v_M) \quad (2-5)$$

where $N = M = 200$ and $u_1 = v_1 = 0$, $u_N = v_M = 1$. Moreover, we still get the database in mesh-type as Fig. 2-10, which definition is same as STL specification we mentioned in section 1.3. The triangular-form datatype features efficient

programmability for computer graphic process.

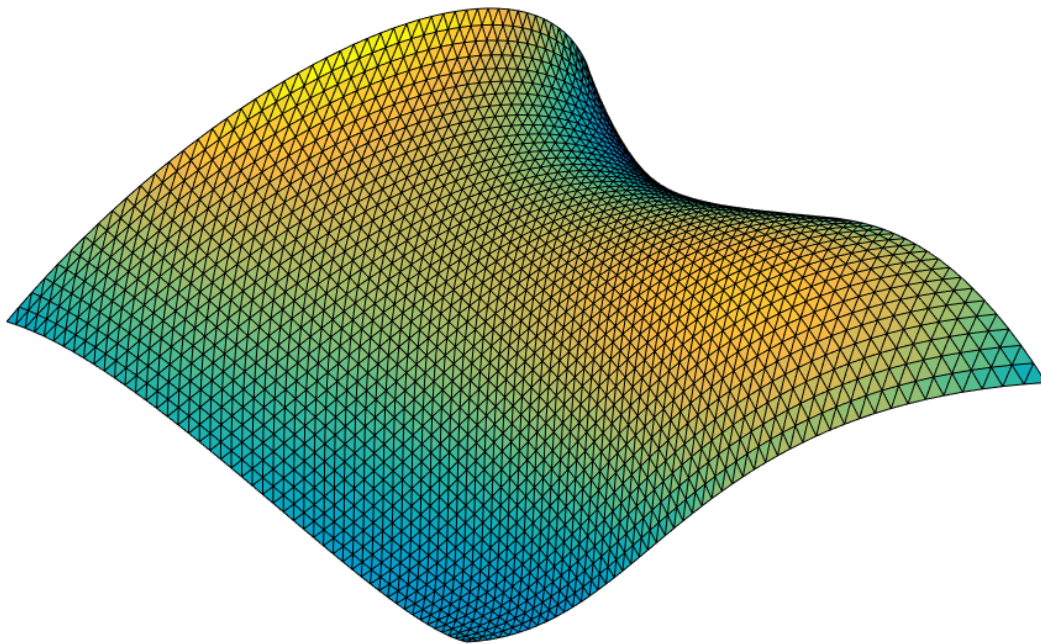
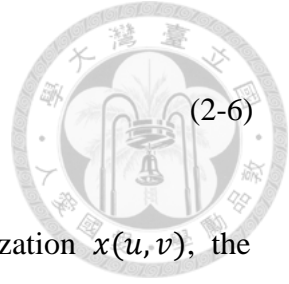


Fig. 2-10 NURBS surface in mesh-type

2.4.3 Geometrical Characteristics Computation

To probe the measured points in normal direction and make some geometrical analysis, we need to infer some geometrical values about the characteristics of the surface, such as principal curvature, Gaussian curvature, mean curvature, principal angle, and normal vector. The following steps are the description about the mathematical inference about these values. All data is obtained from the mesh-type surface.

The normal vector is the vector perpendicular to the surface at the given point, which is commonly denoted N . For a plane specified by $f(x, y, z) = ax + by + cz + d = 0$, the normal vector is given by



$$N = \nabla f = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2-6)$$

where ∇f denotes the gradient. For the surface with parameterization $x(u, v)$, the normal vector is given by cross product of the partial derivatives.

$$N = \frac{\partial x}{\partial u} \times \frac{\partial x}{\partial v} \quad (2-7)$$

The result is shown as Fig. 2-11, all normal vector has two opposite directions. Here we choose the outward-pointing normal vector for displaying, while the inward-pointing normal vector is used as probing direction for implementing inspection.

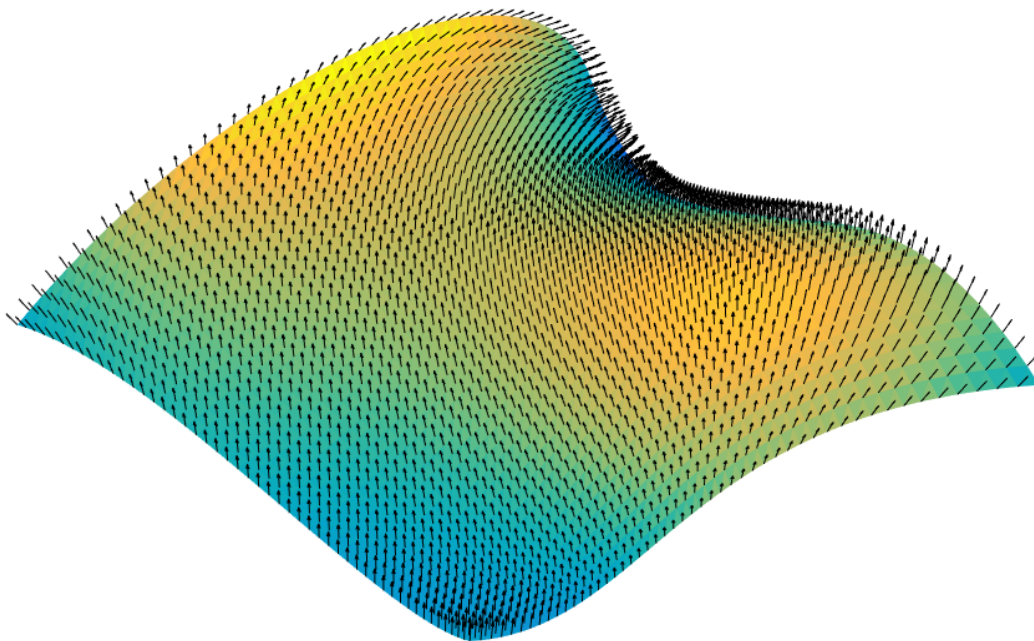
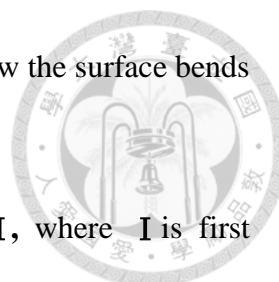


Fig. 2-11 Normal vector of the designed surface

Besides normal vectors, curvature is also an important characteristic of the surface. Intuitively, curvature is the amount by which a surface deviates from being a flat plane, or a curve from being straight. The principal curvature is certainly useful for a surface. It measures the maximum and minimum bending of a surface at each point, and it also



derives their principal directions respectively. Physically it means how the surface bends by different amounts of different directions at the point.

The principal comes from the eigenvalues of the matrix, $\mathbf{I}^{-1} \mathbf{\Pi}$, where \mathbf{I} is first fundamental form and $\mathbf{\Pi}$ is second fundamental form.

$$\mathbf{I} = \begin{bmatrix} X_u \cdot X_u & X_u \cdot X_v \\ X_v \cdot X_u & X_v \cdot X_v \end{bmatrix} \quad (2-7)$$

$$\mathbf{\Pi} = \begin{bmatrix} X_{uu} \cdot N & X_{uv} \cdot N \\ X_{vu} \cdot N & X_{vv} \cdot N \end{bmatrix} \quad (2-8)$$

where the subscript u, v means partial derivative and N is unit normal vector of the point.

Fig. 2-12 show the principal directions $\overrightarrow{U_{max}}$ which represents the maximum curvature κ_1 and $\overrightarrow{U_{min}}$ which represents the maximum curvature κ_2 .

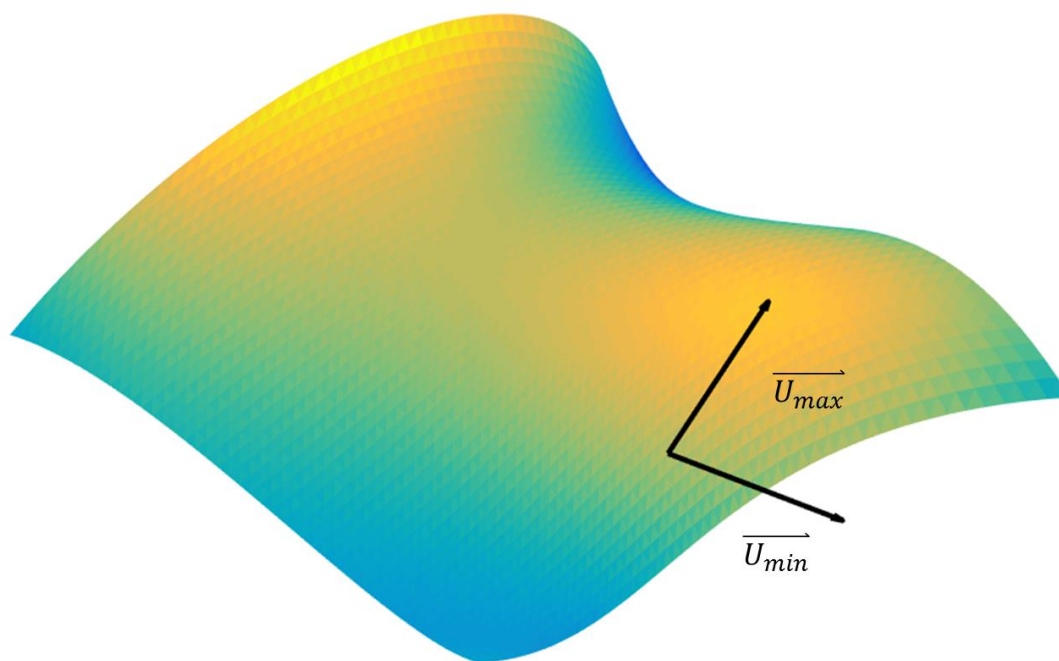
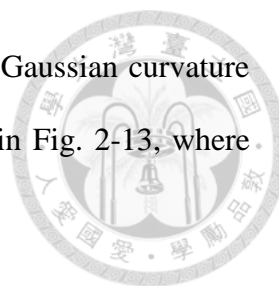


Fig. 2-12 Principal direction of a point

Besides, we also can utilize the principal curvature to infer the Gaussian curvature and Mean curvature. Gaussian curvature is an intrinsic measure of curvature, depending



only on distances that are measured in the surface. The sign of the Gaussian curvature can identify the characteristics of the surface. The result is shown in Fig. 2-13, where the values are displayed in color form.

$$K = \kappa_1 \kappa_2 = \frac{eg - f^2}{EG - F^2} \quad (2-9)$$
$$H = \frac{1}{2}(\kappa_1 + \kappa_2)$$

where E, F, G are first fundamental coefficients and e, f, g are coefficients of the second fundamental form.

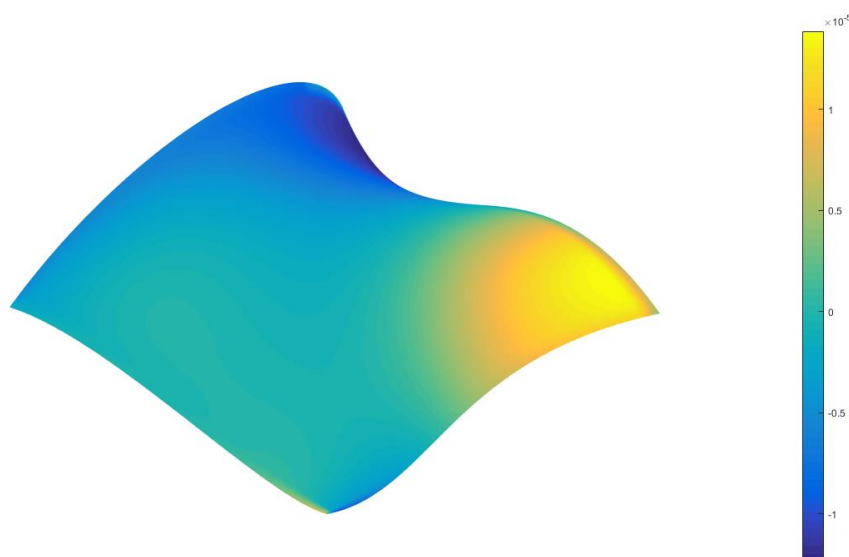
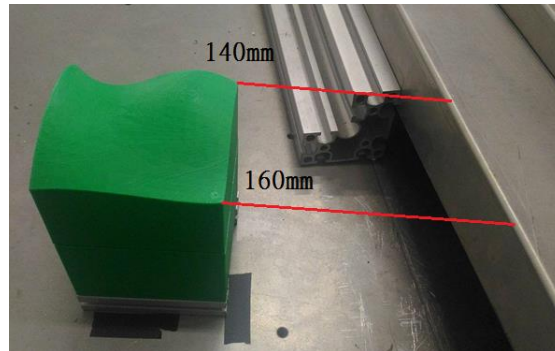


Fig. 2-13 Gaussian curvature for the surface

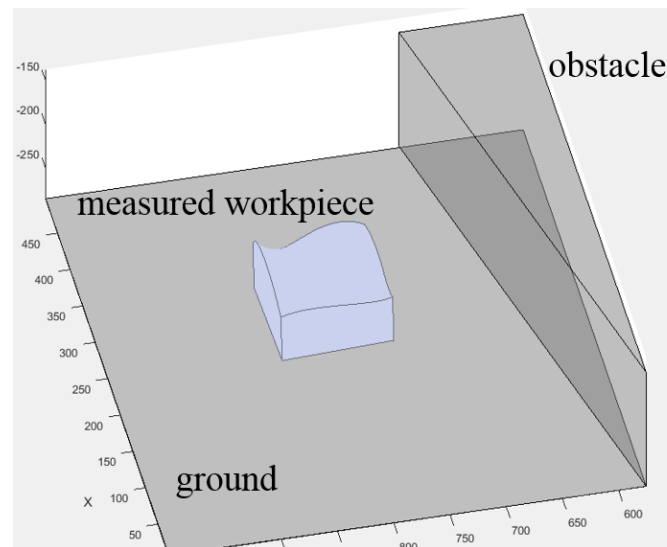
2.5 Simulation Workspace

In our case, there exists some other works we need to finish besides measurement because the physical condition should be taken into consideration. The iCeIRA CNC robot is a precise but fragile machine. Any collision may lead to enormous damage to the machine which makes the accuracy lower. To prevent the accidents and avoid all cases that causes such the tragedy. A simulation workspace should be done and

visualized to let the user certainly know what motions or postures cause dangers. We try to build a complete simulation workspace with physical parameters in reality. To save computation time and cost, the simulation range is limited into the space around the measured workpiece, which all postures of the CNC robot will be inside in. The simulation is shown in Fig. 2-14.



(a) Real working environment and physical parameter

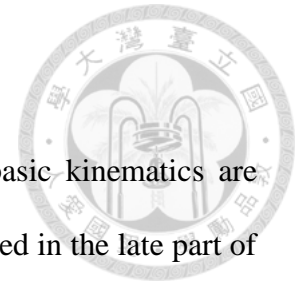


(b) Simulation workspace

Fig. 2-14 Simulation workspace of real environment

We take the localized measured workpiece as the reference of the environment distribution. The localization method will be discussed in Chapter 4.

Chapter 3 CMM Kinematics



In this chapter, mathematic of 3D spatial descriptions and basic kinematics are described. The kinematics of the iCeIRA CNC robot is also introduced in the late part of the section.

3.1 Spatial Descriptions and Transformation

We will use ${}^A V$ or ${}^A v$ to describe a point or a vector in this thesis, so we will not always use \rightarrow to represent a vector. The upper-title means the vector is in terms of coordinates frame $\{A\}$. The subtitle can represent a specific point of a position; represent a vector from point A to point B like P_{AB} ; or it can describe a position or a vector as a specific point or other specific meaning.

We can represent the orientation of a coordinate frame $\{B\}$ by its unit vectors expressed in terms of the reference coordinate frame $\{A\}$. Each unit vector has three elements and they form the columns of a 3×3 orthonormal matrix ${}^A R_B$. The matrix is also a rotation transformation matrix that can transform points from frame $\{B\}$ to reference frame $\{A\}$.

A 4×4 homogeneous transformation matrix has the structure like

$${}^A T_B = \begin{bmatrix} {}^A R_B & {}^A P_{B,o} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-1)$$

where “A” and “B” represent coordinate frames; “ B,o ” means it is an original point of frame $\{B\}$. The matrix can easily represent the relation including rotation and translation from frame $\{A\}$ to $\{B\}$ and it can transform points from frame $\{B\}$ to frame $\{A\}$. Often, we will use notation ${}^A T_B$ or T_{AB} to represent the transformation matrix and use $T = (R,P)$ to represent the matrix T is composed by rotation matrix R and position vector P. In order to represent a point in frame $\{B\}$ in terms of $\{A\}$, the operation is



$$\begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A R_B & {}^A P_{B,o} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B P \\ 1 \end{bmatrix} \quad (3-2)$$

It is seen that implements the equation

$$\begin{aligned} {}^A P &= {}^A R_B {}^B P + {}^A P_{B,o} \\ 1 &= 1 \end{aligned} \quad (3-3)$$

It is obviously mapping the point P from frame $\{B\}$ to $\{A\}$. Also, the inverse of the transformation matrix represents the inverse operation of this mapping, which is

${}^B T_A = ({}^A T_B)^{-1}$, ${}^B T_A {}^A T_B = I$. The inverse of ${}^A T_B$ can be derived as

$${}^A T_B^{-1} = {}^B T_A = \begin{bmatrix} {}^A R_B^T & -{}^A R_B^T {}^A P_{B,o} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-4)$$

3.1.1 Three-Angle Representation

Any 3x3 rotation matrix can be specified by just three parameters. A well-known method is ZYX Euler Angle representation:

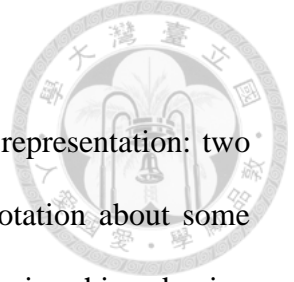
$$R = R_x(\alpha)R_y(\beta)R_z(\gamma) \quad (3-5)$$

The inverse formula is:

$$\text{if } R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \text{ then}$$

$$\begin{aligned} \beta &= \text{atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\ \alpha &= \text{atan2}(r_{32}/c\beta, r_{33}/c\beta) \\ \gamma &= \text{atan2}(r_{21}/c\beta, r_{11}/c\beta) \end{aligned} \quad (3-6)$$

A fundamental problem with the three-angle representation is singularity. This occurs when the rotational axis of the middle term in the sequence becomes parallel to the rotation axis of the first or third term. This is the same problem as gimbal lock.



3.1.2 Angle–Axis Representation

Another method to describe Rotation is equivalent angle–axis representation: two coordinate frames of arbitrary orientation are related by a single rotation about some axis in space. Converting from angle and vector to a rotation matrix is achieved using Rodrigues' rotation formula

$${}^A R_B = I_{3 \times 3} + \sin \theta S({}^A u) + (1 - \cos \theta)({}^A u {}^A u^T - I_{3 \times 3}) \quad (3-7)$$

noting that, the outer product ${}^A u {}^A u^T = [S({}^A u)]^2 + I_{3 \times 3}$. So another equivalent expression is

$${}^A R_B = \sin \theta [S({}^A u)] + \cos \theta [S({}^A u)]^2 + {}^A u {}^A u^T \quad (3-8)$$

The meaning of this equation is rotating {B} relative {A} about the unit vector u by an angle θ according to the right-hand rule. Where $S(u)$ is a 3x3 skew-symmetric matrix of the 3x1 vector u . A matrix S is defined as a skew-symmetric matrix if and only if $S^T + S = 0$. It has zero diagonal and the property that $v^T S(v) = 0, \forall v$. For the 3x3 case:

$$S(v) = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \quad (3-9)$$

Finally, it can be shown that the rotation matrix is

$$R(u, \theta) = \begin{bmatrix} u_x u_x v\theta + c\theta & u_x u_y v\theta - u_z s\theta & u_x u_z v\theta + u_y s\theta \\ u_x u_y v\theta + u_z s\theta & u_y u_y v\theta + c\theta & u_y u_z v\theta - u_x s\theta \\ u_x u_z v\theta - u_y s\theta & u_y u_z v\theta + u_x s\theta & u_z u_z v\theta + c\theta \end{bmatrix} \quad (3-10)$$

Where $c\theta = \cos\theta$, $s\theta = \sin\theta$, $v\theta = 1 - \cos\theta$, and ${}^A u = [u_x \quad u_y \quad u_z]^T$. The inverse operation can be computed, if



$${}^A R_B(u, \theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3-11)$$

then

$$\theta = \arccos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \quad (3-12)$$

and

$${}^A u = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \quad (3-13)$$

The θ has a value between 0 and 180 degrees. For any solution $({}^A u, \theta)$, there is another pair solution $(-{}^A u, -\theta)$, which results in the same orientation in space. Also, another problems of this method is that, when θ near to 0 or 180 degrees, the axis becomes ill-defined

3.1.3 Unit Quaternions

Quaternions have been controversial since they were discovered by W.R. Hamilton over 150 years ago but they have great utility in robotics, computer vision, computer graphic and aerospace navigation applications. The quaternion is an extension of the complex number – a hyper-complex number – and is written as a scalar plus a vector

$$\begin{aligned} Q &= w + v \\ &= w + xi + yj + zk \end{aligned} \quad (3-14)$$

where $w \in \mathbb{R}$, $v \in \mathbb{R}^3$ and the orthogonal complex numbers, i, j and k are defined such that

$$i^2 = j^2 = k^2 = ijk = -1 \quad (3-15)$$

We will denote a quaternion as

$$Q = w \langle x, y, z \rangle \quad (3-16)$$

To represent rotations we use unit-quaternions. These are quaternions of unit magnitude, that is, $|Q|=1$ or $w^2 + x^2 + y^2 + z^2 = 1$. The unit-quaternion has the special property that it can be considered as a rotation of θ about the unit vector \hat{n} which are related to the quaternion components by

$$w = \cos \frac{\theta}{2}, \quad v = \left(\sin \frac{\theta}{2} \right) \hat{n} \quad (3-17)$$

and is similar to the angle-axis representation of section 2.1.2. The transformation operation of rotation matrix is left-multiplication of matrix. The same operation in quaternion is known as Hamilton product, we denote it as $Q_2 \otimes Q_1$ and

$$Q_2 \otimes Q_1 = w_2 w_1 - v_2 \cdot v_1 \langle w_2 v_1 + w_1 v_2 + v_2 \times v_1 \rangle \quad (3-18)$$

It can be expressed as a matrix-vector product where

$$Q_2 \otimes Q_1 = \begin{bmatrix} w_2 & x_2 & y_2 & z_2 \\ -x_2 & w_2 & -z_2 & y_2 \\ -y_2 & z_2 & w_2 & -x_2 \\ -z_2 & -y_2 & x_2 & w_2 \end{bmatrix} \begin{bmatrix} w_1 \\ x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (3-19)$$

As we can see, computing the Hamilton product requires 16 multiplications and 12 additions. But computing two orthonormal rotation matrices requires 27 multiplications and 18 additions. Quaternion is more efficiency.

According to Eq. (2-17), the quaternion can be algebraically manipulated into an equivalent matrix to represent the same rotation, that is

$$R = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2(z^2 + x^2) & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2(x^2 + y^2) \end{bmatrix} \quad (3-20)$$

and if the rotation matrix R is given, the matrix can be converted to a quaternion using

this basic form:

$$\begin{aligned}
 w &= \sqrt{r_{11} + r_{22} + r_{33} + 1} / 2 \\
 x &= (r_{32} - r_{23}) / w \\
 y &= (r_{13} - r_{31}) / w \\
 z &= (r_{21} - r_{12}) / w
 \end{aligned}
 \tag{3-21}$$



but it tend to be unstable and un-accuracy when $trace(R)+1$ of the rotation matrix is zero or very small. a better algorithm to compute the quaternion from a rotation matrix is shown in Fig. 3-1.

Algorithm of computing a Quaternion converted from a Rotation Matrix
<pre> tr(0) = R(0,0)+R(1,1)+R(2,2)+1; tr(1) = R(0,0)-R(1,1)-R(2,2)+1; tr(2) = -R(0,0)+R(1,1)-R(2,2)+1; tr(3) = -R(0,0)-R(1,1)+R(2,2)+1; for i=0:3 { if(tr(i) < 0) tr(i) = 0; s(i)= 0.5*sqrt(tr(i)); } if(s(0)>=s(1) && s(0)>=s(2) && s(0)>=s(3)){ w = s(0); x = sign(R(2,1)-R(1,2))*s(1); y = sign(R(0,2)-R(2,0))*s(2); z = sign(R(1,0)-R(0,1))*s(3); } else if(s(1)>=s(0) && s(1)>=s(2) && s(1)>=s(3)){ w = sign(R(2,1)-R(1,2))*s(0); x = s(1); y = sign(R(1,0)+R(0,1))*s(2); z = sign(R(0,2)+R(2,0))*s(3); } else if(s(2)>=s(0) && s(2)>=s(1) && s(2)>=s(3)){ w = sign(R(0,2)-R(2,0))*s(0); x = sign(R(1,0)+R(0,1))*s(1); y = s(2); z = sign(R(2,1)+R(1,2))*s(3); } else if(s(3)>=s(0) && s(3)>=s(1) && s(3)>=s(2)){ w = sign(R(1,0)-R(0,1))*s(0); x = sign(R(0,2)+R(2,0))*s(1); y = sign(R(2,1)+R(1,2))*s(2); z = s(3); } </pre>

Fig. 3-1 the algorithm pseudo code of computing a quaternion convert from a rotation matrix

This approach avoids square looting very small number to reduce the floating point error. But the computed result may not be exactly unit quaternion, so we still need to

normalize the result. The normalization is achieved by ensuring that the quaternion norm is unity, a straightforward division all elements by the quaternion norm

$$Q' = \frac{Q}{|Q|} \quad (3-22)$$

It is easier to normalize a quaternion rather than a non-orthogonal matrix. To summary, the advantages of using quaternion to represent rotation are list:

1. Quaternions doesn't have gimbal lock problem but Euler angle representation does.
2. Quaternions composed from four numbers but rotation matrix need nine numbers.
3. Similar to rotation matrix, the product of two quaternions can represent twice rotations but the computation is less.
4. Easier interpolation between quaternions.
5. Quaternion, matrix operations will accumulate rounding error or integration error, so we need to do each of its normalization and orthogonalization. But "Restoring" a rotation matrix is computationally more expensive than normalizing a quaternion.

3.2 Differential Motion

A body rotating in 3D space has an angular velocity which is a vector quantity $\omega = [\omega_x \ \omega_y \ \omega_z]^T$. The direction of this vector defines the instantaneous axis of rotation, that is, the axis about which the coordinate frame is rotating at a particular instant of time. In general, this axis changes with time. The magnitude of the vector is the rate of rotation about the axis. There is a well-known expression for the derivative of a time-varying rotation matrix

$$\dot{R}(t) = S(\omega)R(t) \quad (3-23)$$

where $S(\cdot)$ is a skew-symmetric matrix introduced before.

$$S(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (3-24)$$



Consider the approximation of derivative

$$\begin{aligned} R(t + \Delta t) &\approx \dot{R}\Delta t + R(t) \\ &\approx S(\omega)R(t)\Delta t + R(t) \approx (S(\omega)\Delta t + I_{3 \times 3})R(t) \end{aligned} \quad (3-25)$$

which describes how the orthonormal rotation matrix changes as a function of angular velocity.

Consider a coordinate frame that undergoes a small rotation from R_0 to R_1 . Let the small rotation $\Delta\theta = \omega\Delta t$, $\Delta\theta = [\Delta\theta_x \ \Delta\theta_y \ \Delta\theta_z]^T$, $\omega = [\omega_x \ \omega_y \ \omega_z]^T$. We can write (3-25) as

$$R_1 = (S(\omega)\Delta t + I_{3 \times 3})R_0 = (S(\Delta\theta) + I_{3 \times 3})R_0 \quad (3-26)$$

and rearrange it as

$$S(\Delta\theta) = R_1 R_0^T - I_{3 \times 3} \quad (3-27)$$

and then apply the inverse operation of $S(\cdot)$, defined as $\text{vex}(\cdot)$, to both sides

$$\Delta\theta = \text{vex}(R_1 R_0^T - I_{3 \times 3}) \quad (3-28)$$

The detail of vex operator in 3×3 case is

$$\mathbf{v}_{3 \times 1} = \text{vex}(S_{3 \times 3}) = \frac{1}{2} \begin{bmatrix} s_{32} - s_{23} \\ s_{13} - s_{31} \\ s_{21} - s_{12} \end{bmatrix} \quad (3-29)$$

The $\Delta\theta = \omega\Delta t$ is a vector that represents a small rotation about the **world** x -, y - and z -axes.

If we give two homogeneous transformation matrices T_0, T_1 to represent two poses (position and orientation) that has small deference, then we can represent the deference between them as a 6×1 vector composed by deference of two position vectors and deference of two rotation matrix

$$\delta = \Delta(T_0, T_1) = \begin{bmatrix} P_1 - P_0 \\ \text{vex}(R_1 R_0^T - I_{3 \times 3}) \end{bmatrix}, \delta \in \mathbb{R}^{6 \times 1} \quad (3-30)$$

Where $T_0 = (R_0, P_0)$, $T_1 = (R_1, P_1)$, In practice, this equation can be used to compute the differential of motion in 3D space. Let $T_0 = T(t)$, $T_1 = T(t+\Delta t)$ and $T = (R, P)$, than

$$\begin{bmatrix} v \\ \omega \end{bmatrix} \approx \begin{bmatrix} dP/d\Delta t \\ dR/d\Delta t \end{bmatrix} = \frac{\delta}{\Delta t} \quad (3-31)$$

where v and ω is instantaneous linear velocity and angular velocity at pose T and time t .

The character “d” will mean “small deference” or deferential conveniently in this thesis.

The inverse of Eq. (2-30) is

$$T_1 = \Delta^{-1}(T_0, \delta) = \left[\begin{bmatrix} S(dR) & dP \\ 0_{1 \times 3} & 0 \end{bmatrix} + I_{4 \times 4} \right] T_0, \delta = \begin{bmatrix} dP \\ dR \end{bmatrix} \quad (3-32)$$

In practice, this equation can be used to compute the integration of motion in 3D space.

The example is trajectory generation. It will introduce in chapter 4.

3.3 CMM Kinematics

3.3.1 Forward Kinematics of CMM

A serial-link manipulator comprises a set of links in a chain and connected by joints. Each joint has one degree of freedom. Motion of the revolute joint changes the relative angle of its neighboring links while the prismatic one changes relative translation. A systematic way of describing the geometry of a serial chain of links and joints is Denavit-Hartenberg (D-H) notation. Once these D-H link coordinate frames are assigned, transformation between adjacent coordinate frames can be represented by a single standard 4x4 homogeneous transformation matrix.

Fig. 3-2 shows the definition of standard D-H parameters. The method of building

a D-H representation is the following procedure:

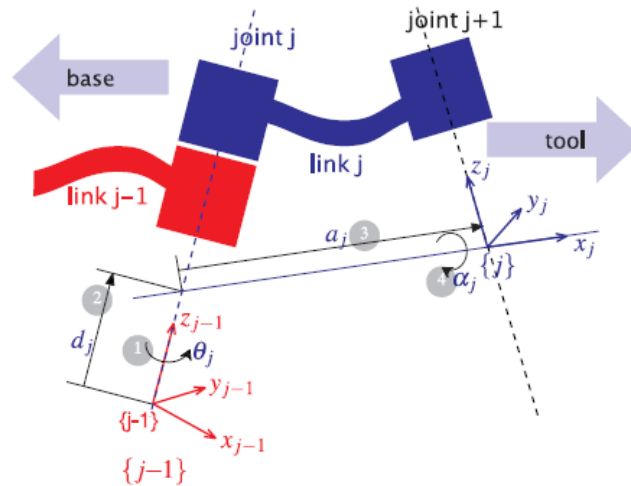


Fig. 3-2 definition of standard D-H parameters

First, the coordinate frames are assigned to the links by the following rules:

1. $\hat{z}_{j-1} \rightarrow j^{th}$ joint axis
2. $\hat{x}_j \perp \hat{z}_{j-1}$ and point away from it
3. $\hat{y}_j = \hat{z}_j \times \hat{x}_j$
4. $\{0\}$: $\hat{z}_0 \rightarrow 1^{st}$ joint, $\{N\}$: $\hat{x}_N \perp \hat{z}_{N-1}$, \hat{z}_N align with \hat{z}_{N-1}

Then, we can determine the D-H parameter according to these frames by the following rules:

- θ_j : the angle from \hat{x}_{j-1} to \hat{x}_j measured about \hat{z}_{j-1}
- d_j : the distance from \hat{x}_{j-1} to \hat{x}_j measured along \hat{z}_{j-1}
- a_j : the shortest distance from \hat{z}_{j-1} to \hat{z}_j measured along \hat{x}_j
- α_j : the offset angle from \hat{z}_{j-1} to \hat{z}_j measured about \hat{x}_j

The transformation from link coordinate frame $\{i-1\}$ to frame $\{i\}$ is defined in terms of elementary rotation and translation as

$${}^{i-1}T_i(\theta_i, d_i, a_i, \alpha_i) = T_{R,z}(\theta_i)T_z(d_i)T_x(a_i)T_{R,x}(\alpha_i) \quad (3-33)$$

which can be expanded as

$${}^{i+1}T_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-34)$$

For a revolute joint θ_i is the joint variable and d_i is constant, while for prismatic joint d_i is variable and θ_i is constant and $\alpha_i = 0$.

The forward kinematics (FK) is often expressed in functional form with the end-effector pose as a function of joint coordinates (the coordinate frame of the end-effector is usually same as the last N^{th} coordinate frame). Using the product of the individual link transformation matrices for an N-axis manipulator is

$${}^0T_E = {}^0T_1(q_1) {}^1T_2(q_2) \cdots {}^{i-1}T_i(q_i) \cdots {}^{N-1}T_N(q_N) \quad (3-35)$$

where “ q ” represents the joint space variable $q = [q_1 \ q_2 \ \dots \ q_i \ \dots \ q_N]^T$. The 0T_E include a rotation matrix and a position vector just describes the pose of the end-effector.

3.3.2 Velocity Relationship: The Manipulator Jacobian

If we write the FK in functional form as

$$x = f(q) \quad (3-36)$$

and taking the derivative we write

$$v = \frac{df(q)}{dq} = J(q)\dot{q} \quad (3-37)$$

which is the instantaneous FK where $v = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$ is spatial velocity and comprises translational and rotational velocity components. The matrix $J(q) \in \mathbb{R}^{6 \times N}$ is the manipulator Jacobian.

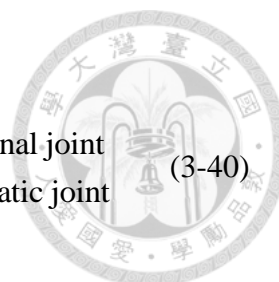
Jacobian matrices are useful in robot arm inverse kinematics, trajectory planning and control. We can compute Jacobian matrices numerically by using Eq. (3-32) to approximate the derivative of transformation matrix with respect to joint coordinates. We perturb i^{th} joint by a small but finite amount dq_i , and let T_1 in Eq.(3-32) as $T(q_i+dq_i)$ and $T_0 = T(q_i)$, then, the vector $J_i \in \mathbb{R}^{6 \times 1}$ which is i^{th} column of the Jacobian matrix can be approximate as

$$J_i \approx \frac{\Delta(T(q_i), T(q_i + dq_i))}{dq_i} \quad (3-38)$$

In practice, we will prefer to compute Jacobian matrices in analytic form for efficiency in programing. In fact, we can get the symbolic form forward kinematic 0T_E and symbolic form of a Jacobian matrix by symbolic computation ability of the powerful software MATLAB. The derivation of a Jacobian matrix has two parts. First part is $J_v \in \mathbb{R}^{3 \times N}$, the translational part of the Jacobian matrix. Obviously, each column of J_v is partial differential of translational FK function with respect to each joint variable. After we compute the FK ${}^0T_E = ({}^0R_E, {}^0P_E)$ in terms of symbolic joint coordinate $q \in \mathbb{R}^{N \times 1}$, we can get the analytical $J_v(q)$ which is

$$J_v(q) = \left[\frac{\partial {}^0P_E}{\partial q_1} \quad \frac{\partial {}^0P_E}{\partial q_2} \quad \dots \quad \frac{\partial {}^0P_E}{\partial q_i} \quad \dots \quad \frac{\partial {}^0P_E}{\partial q_N} \right] \quad (3-39)$$

by input `jacobian(${}^0P_E, q$)` in MATLAB. The `jacobian()` is a build-in function in MATLAB. Second part is $J_\omega \in \mathbb{R}^{3 \times N}$, the rotational part of the Jacobian matrix. The angular velocity of the end-effector is the sum of angular velocity of each rotation joints, we can write as ${}^0\vec{\omega}_E = {}^0\vec{\omega}_1 + {}^0\vec{\omega}_2 + \dots + {}^0\vec{\omega}_N$ (${}^0\vec{\omega}_i = \mathbf{0}_{3 \times 1}$ as it is prismatic joint). Because the direction of angular velocity of i^{th} rotation joint is ${}^0R_{i-1} \hat{z}_{i-1} = \hat{z}_{i-1} = {}^0R_{3,i-1}$, it equal to the third column of the $(i-1)^{\text{th}}$ rotation matrix relative to base, we can derive the



velocity relation in matrix form, which is

$$J_\omega(q) = [\sigma_1^0 \hat{z} \quad \sigma_2^1 \hat{z} \quad \cdots \quad \sigma_i^{i-1} \hat{z} \quad \cdots \quad \sigma_N^{N-1} \hat{z}], \sigma_i \begin{cases} = 1 \text{ for rotational joint} \\ = 0 \text{ for prismatic joint} \end{cases} \quad (3-40)$$

Then, we can write Eq. (2-37) as

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \dot{q} \quad (3-41)$$

3.4 Forward Kinematics Model of iCeIRA CNC robot

iCeIRA CNC robot is a gantry-typed 6 axis machine. The high degree of freedom enables measurement task can be achieved with appropriate probe and control strategy. Compared with common CMMs, the extra degree of freedom leads to more flexible machine posture for a probing point. Fig. 3-3 shows the assigned DH link frames and parameters of the CNC robot.

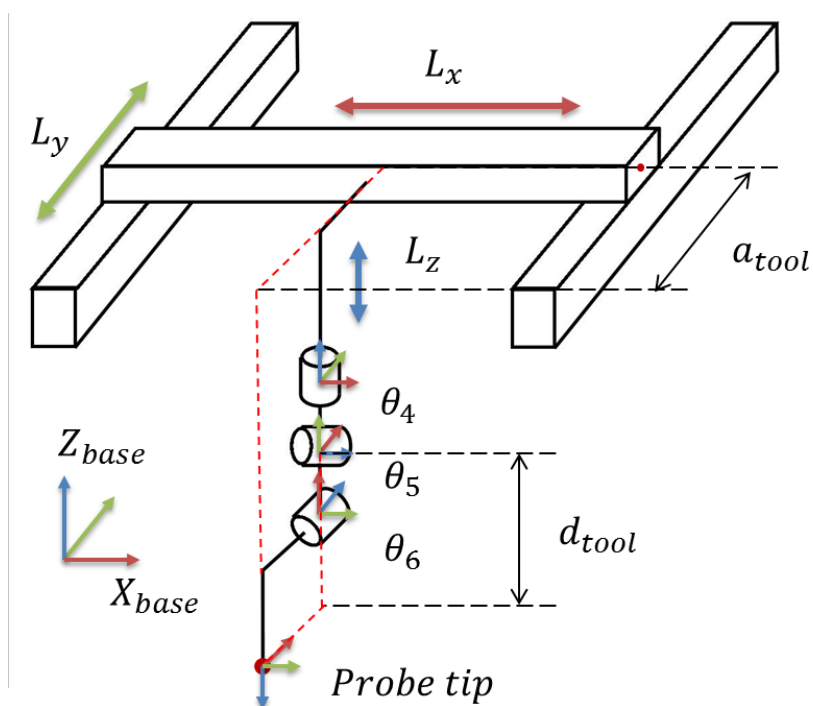


Fig. 3-3 the assigned DH link frames and DH parameters of iCeIRA CNC robot

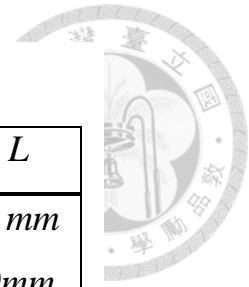


Table 3-1 link parameter of iCeIRA CNC robot

i	$\theta_i(\text{rad})$	d_i	a_i	a_i	U	L
1	-	L_x	90°	0	1000mm	0 mm
2	-	0	0°	L_y	1000mm	0mm
3	-	$-L_z$	-90°	0	60mm	45mm
4	$\theta_4 + 0$	0	90°	0	170°	175°
5	$\theta_5 + 0$	0	90°	0	90°	-10°
6	$\theta_6 + 0$	d_{tool}	180°	a_{tool}	0°	-135°

U : joint angle upper bound

L : joint angle lower bound

The detail geometry parameter and numerical value are list in chapter 5. Table 3-1 is the DH table according to the DH frames in Fig. 3-1. The first three joints are prismatic joints and the other three joints are rotational joint. Also, the joint limits are listed.

The joint 1, 2, 3 are prismatic joint which control the translation of gantry. The joint 4, 5, 6 are rotational joint, and their axes intersect at a single point at the center of joint 5. We can view this kind of structure as a single point at the intersection. In geometry, this kind of mechanism is called “vertical spherical joint” which have three DoF provide roll-pitch-yaw rotation work on the point of intersection. In industrial design field, this geometrical structure is called spherical joint architecture. The feature of this design is useful and convenient for computation of inverse kinematics which we will discuss in the next section.

3.5 Inverse Kinematics Model of iCeIRA CNC robot

Compared with the forward kinematics of which functionality is to obtain spatial information (position and orientation) of the end-effector while joint angles are taken as inputs, inverse kinematics (IK) is just contrary. To control the robot more intuitively, the IK solution allows users to control the robot by defining state of end-effector. In other words, the corresponding joint angles are determined by IK solution while the position and orientation of the TCP is defined.

3.5.1 Numerical Solution - The Jacobian Transpose Method

In general case, there may be no analytic IK solution can be found for a joint configuration of a robot with high DoFs. Thus, the numerical method has been introduced to solve the problem for general robots. Certainly, the computing time is longer than analytic solution due to the iterations.

Using the Jacobian linearization method, the joint velocity can be mapped to the velocity in Cartesian space with Jacobian matrix, as shown in Eq. (2-37). The Jacobian matrix J may not be a square matrix when the DoFs is not equal to 6. Thus, pseudo-inverse is used to solve the joint velocity for the desired velocity in Cartesian space. As shown in Eq. (3-1)

$$\dot{q} = J^\dagger \dot{x}, J^\dagger = (J^T J)^{-1} J^T \quad (3-42)$$

The procedure of an IK solver with dynamic gain is shown in Fig. 3-4. The dynamic gain can increase converge speed of the iteration when the initial value and changing rate are adjusted properly.

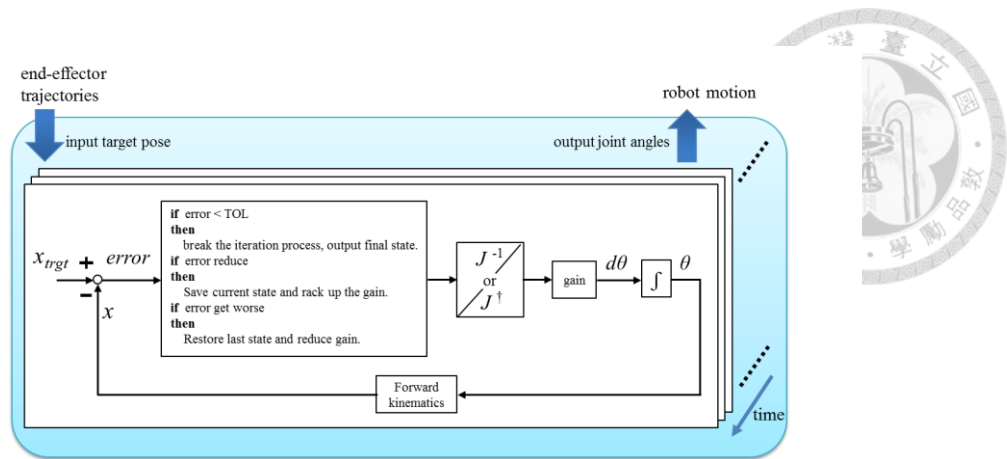


Fig. 3-4 The procedure of numerical method for solving IK

In the figure, the end tip target postures are taken as inputs to the IK solver. Firstly, the robot's pose is calculated with FK, and then the IK solver updates the joint angles with Pseudo-inverse calculation. These two step are repeated till the end tip reach the target error pose in a acceptable error. During the iteration procedure, if the new joint angles lead a bigger error than the last step, the gain decreased dynamically. If it gets smaller error, the gain increased vice versa. In this procedure, the motion is solved one-by-one with the input trajectory.

When solving IK, singularity configurations may cause discontinuous trajectory or a failure of the IK solver. The state of the solver may not converge. The damped least squares method (DLS) and the weighted least-norm (WLN) method [12] solve this problem. The WLN method can also avoiding joint limit during a movement in a path.

3.5.2 Analytic Solution – Closed-Form Solution

We have known that there is a closed-form solution of a 6-axis robot if the three wrist axes intersect at a single point we called wrist center. This means that motion of the wrist joints only changes the end-effector orientation but not translation of wrist center, so we can decouple the inverse kinematics problem into two simpler problems as inverse translation kinematics and inverse orientation kinematics. Such a mechanism is

known as a spherical wrist and almost all industrial robots with high DoFs have such wrists structure.

In our case, the IK solution is much simpler because of intuitional translation controlled by prismatic joints (gantry-type structure). We define a rotation center of which position is equivalent to translation of gantry. This step simplify IK problem because it geometrically divides IK into translation part and orientation part. For FK, as long as position of rotation center is defined, the orientation is controlled by remained three rotational joints. Finally the tool offset is considered to obtain the pose of tool center point (TCP) which is mounted on the end-effector. Inversely, the first step is to solve $\theta_4, \theta_5, \theta_6$ to fit the orientation of the TCP. By solving the FK in symbolic form, the rotation from the base frame to TCP frame can be calculated.

$${}^{base}R_{tool} = \begin{bmatrix} -c_5s_4 & c_4c_6 - s_4s_5s_6 & -c_4s_6 - c_6s_4s_5 \\ c_4c_5 & c_6s_4 + c_4s_5s_6 & -s_4s_6 + c_4c_6s_5 \\ s_5 & -c_5s_6 & -c_5s_6 \end{bmatrix} \quad (3-43)$$

The TCP frame and position are respectively denoted as R_{tool} and P_{tool} . The symbol "c" and "s" present cosine and sine function respectively, and the numerical subscript presents the θ_i . With the analytic equation, the inverse orientation kinematics can be solved.

$$\begin{aligned} \theta_4 &= \text{atan2}(-{}^{base}r_{tool\ 11}, {}^{base}r_{tool\ 23}) \\ \theta_5 &= \text{atan2}({}^{base}r_{tool\ 31}, \sqrt{{}^{base}r_{tool\ 11}^2 + {}^{base}r_{tool\ 21}^2}) \\ \theta_6 &= \text{atan2}(-{}^{base}r_{tool\ 32}, -{}^{base}r_{tool\ 33}) \end{aligned} \quad (3-43)$$

After solving $\theta_4, \theta_5, \theta_6$, we can infer the translation kinematics solution by inversely infer the position of rotation center. The equation is shown below.

$${}^{base}T_4 = {}^{base}T_{tool} {}^{base}T_4^{-1} \quad (3-44)$$

where the ${}^{base}T_4^{-1}$ is the function of $\theta_4, \theta_5, \theta_6$. And finally we can obtain the gantry

translation, which are values of position part of the ${}^{base}T_4$.

$$Lx = {}^{base}_4r_{14}$$

$$Ly = {}^{base}_4r_{24}$$

$$Lz = {}^{base}_4r_{34}$$



(3-45)

Here shows the simulation results in Fig. 3-5. We use MATLAB to simulate the iCeIRA CNC robot and the path is given with different frame and position.

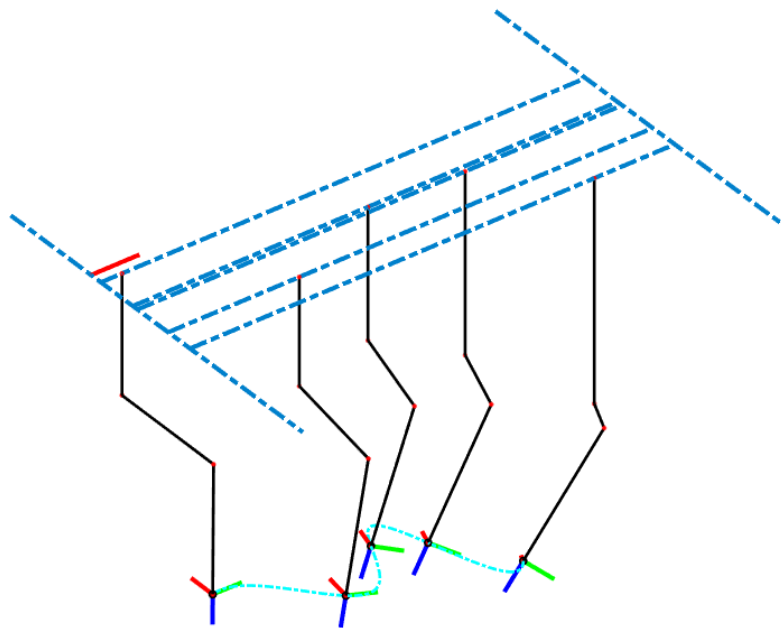
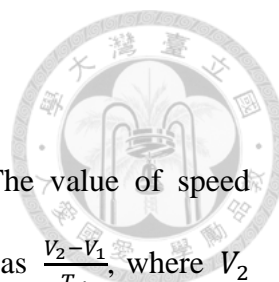


Fig. 3-5 The IK result working on an irregular path



3.6 Linear Blending Motion

Blending motion is a velocity-related control problem for The value of speed change from the current segment to the next segment is calculated as $\frac{V_2-V_1}{T_A}$, where V_2 is the next segment moving speed, V_1 is current segment moving speed and T_A is acceleration time. The illustration is shown in. In this paper, the calculated acceleration will check by algorithm, in other words, the acceleration is restricted by acceleration limit. Acceleration limit is depended on the allowable acceleration value on robot with gantry-type machine tool. For example, if the acceleration $\frac{V_2-V_1}{T_A}$ is larger than allowable value, the speed of the segment, V_1 , will be recalculated.

The blending motion will be used in some applications, such as five axis CNC machine or milling machine. The end-effector of machine tool must move quickly and smoothly in tasks, so the trajectory of end-effector couldn't have any piecewise stop between each segments. The piecewise stop phenomenon sometimes creates vibration on machine tool. In brief, blending move is a significant technique to apply in this paper.

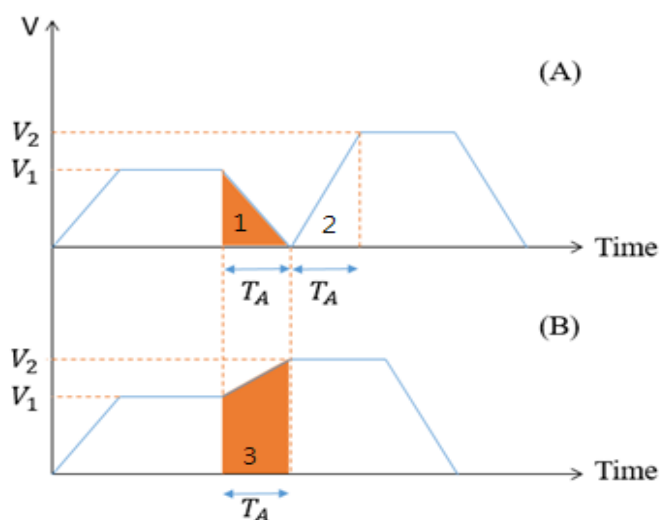
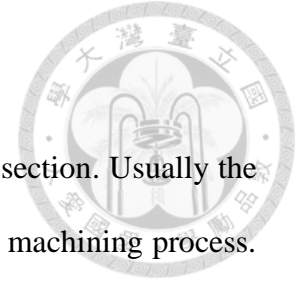


Fig. 3-6 Linear blending motion

Chapter 4 Workpiece Registration



In this chapter, the workpiece registration is introduced in this section. Usually the workpiece registration is also regarded as workpiece localization in machining process.

We will talk about some useful methods with different principals in this section.

4.1 Description

For a CMM, it must have a zero point in machine coordinate system (*MCS*) such that the TCP can be commanded to reach any position in the workspace. However, for a workpiece that we would like to inspect, we usually plan a posture for placing the workpiece onto the CMM platform, which is in workpiece coordinate system (*WCS*). With the ideal above, we can intuitively know that it's impossible that the workpiece locates on the given position without any error. Therefore, the workpiece localization is in needed before starting any measurement process. It's also why the workpiece registration is called localization in the measurement process.

The localization method is trying to find the transformation matrix, T_L , from machine coordinate system (*MCS*) to workpiece coordinate system (*WCS*), so it is also called workpiece localization as shown in Fig. 4-1.

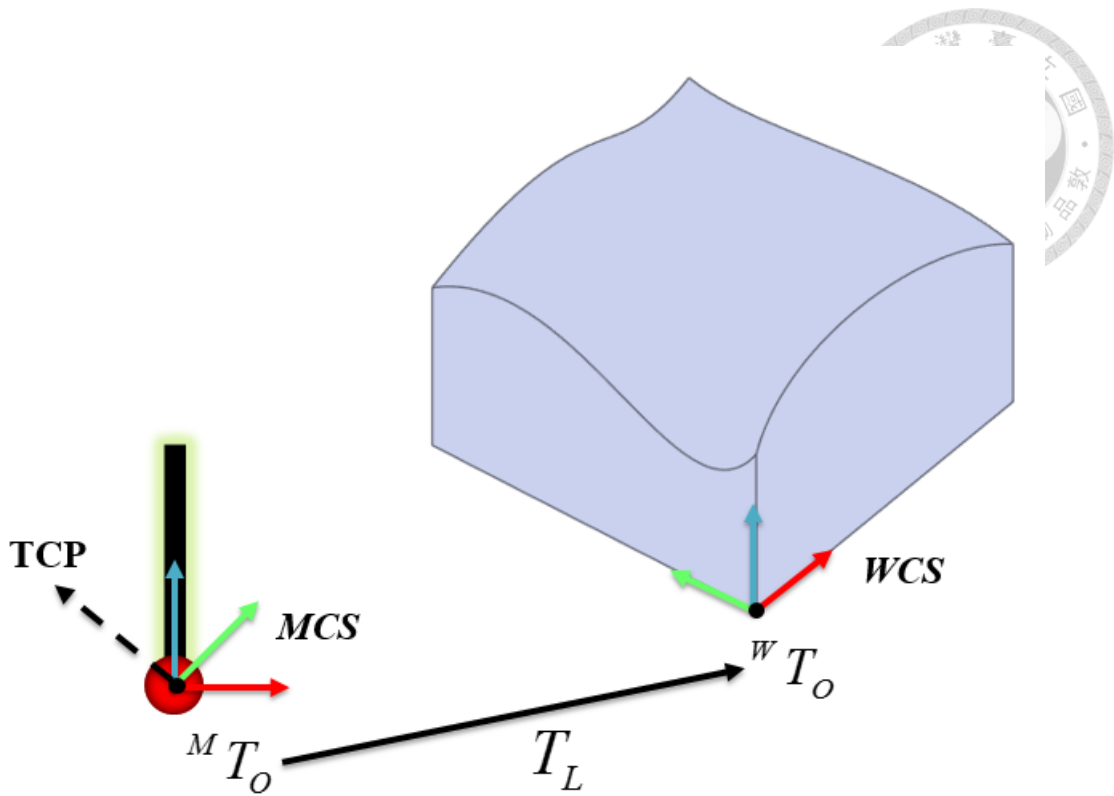


Fig. 4-1 Workpiece Registration

With the localization, the position of a measured point can be defined precisely in *MCS*, which means the CMM is able to touch the target position and obtain accurate data feedback. However, it also means that the accuracy of the localization heavily affects the precision of the measurement raw data. If the localization error is too large, all raw data will be untrusted as shown in Fig. 4-2. As the figure shown, if the localization error is too large, it may lead to inaccurate coordinate transformation and get the wrong result like *WCS**. By observation, the positions of all points onto the surface are certainly imprecise compared with the real *WCS*.

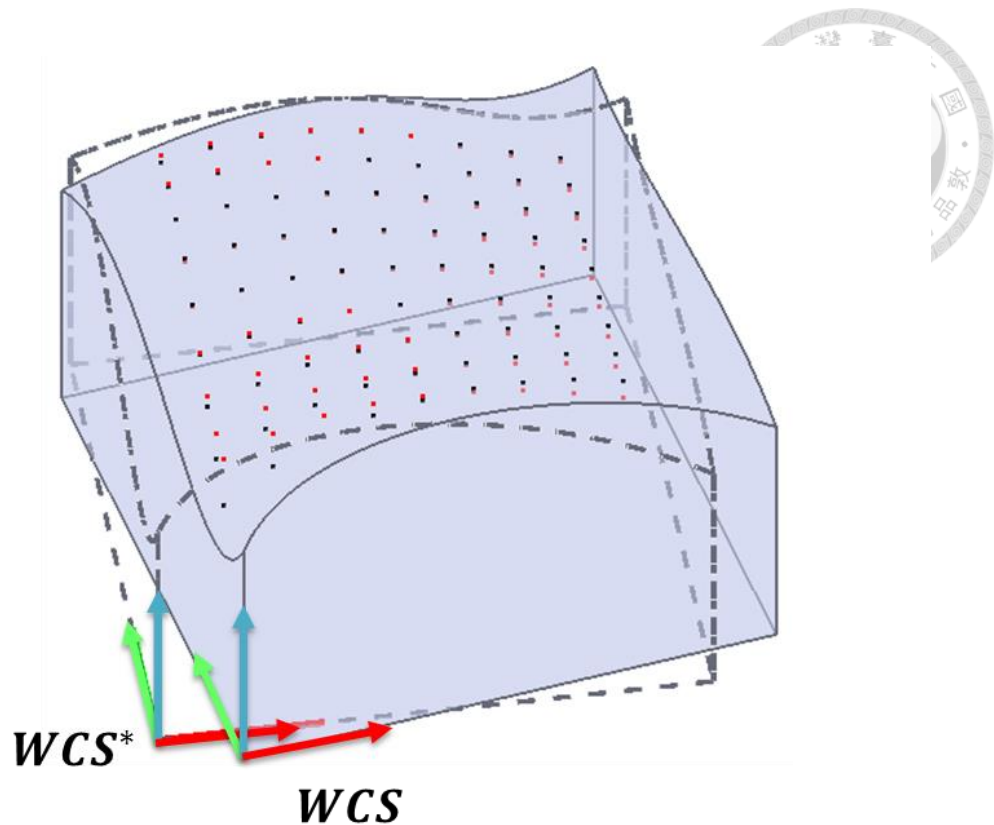


Fig. 4-2 Effect of inaccurate localization

4.2 Reference-based Registration

For a workpiece with some referenced characteristics, such as flat surfaces or protruding point, the localization method will be easy and intuitive. Take the model we design for example, the four flat surfaces around the workpiece are great references for localization. Because the original shape of the workpiece is a cube, so that we can intuitively infer that the surfaces around the workpiece can be used as a great reference.

As long as three points are touched and obtain their corresponding TCP position, it's easy to use the normal vector to describe the surface.

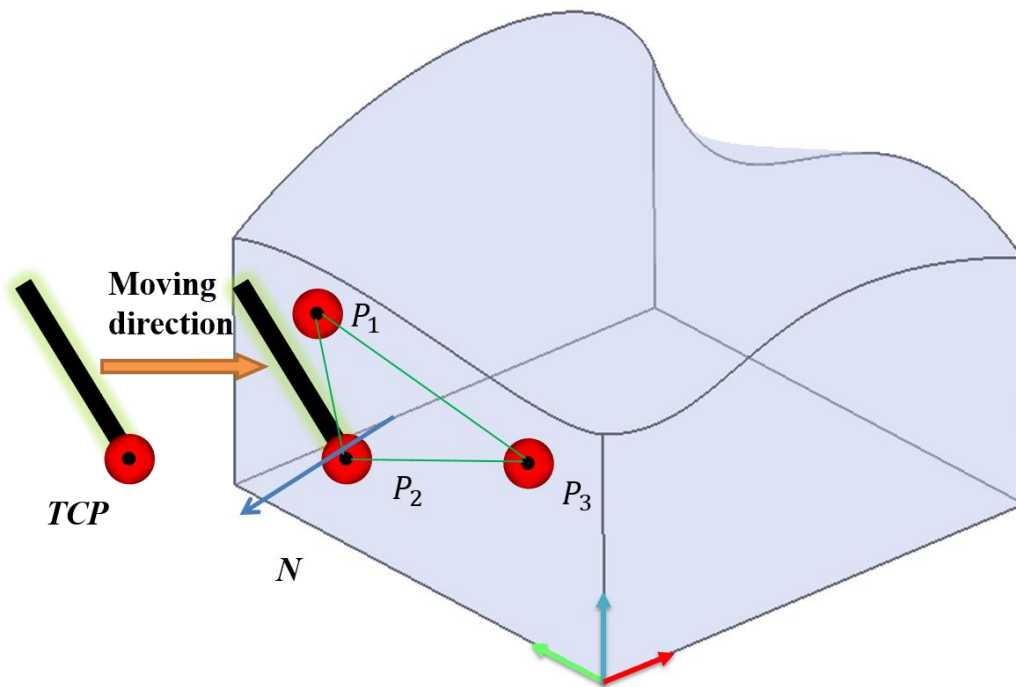
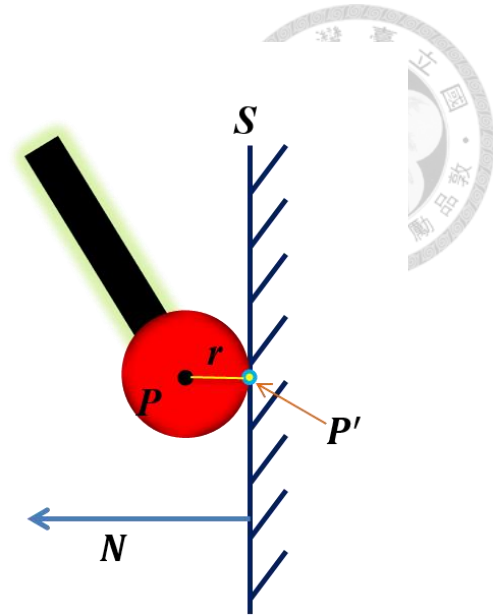
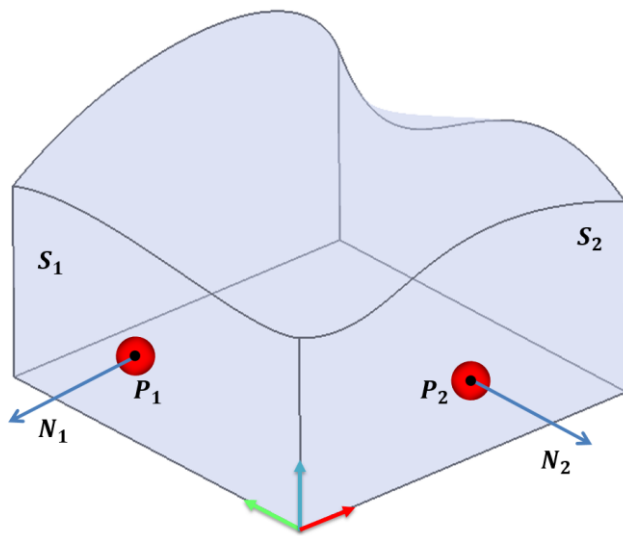


Fig. 4-3 Obtain normal vector of the surface

As the Fig. 4-3 shown, we moving the TCP along a horizontal direction and the TCP position is recorded as the probe is triggered by touching the surface. Assume the TCP positions are P_1 , P_2 and P_3 , we can obtain the normal vector N by the cross product of two vectors.

$$N = \frac{(P_2 - P_1) \times (P_3 - P_1)}{|P_2 - P_1| |P_3 - P_1|} = \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} \quad (4-1)$$

By implementing the same strategy on the adjacent surface, we can obtain the normal vector of the other surface. Here the normal vector of the first surface is denoted N_1 and the one of adjacent surface is denoted N_2 . We assume the normal vectors are all in outward form as shown in Fig. 4-4.



(a) Describe each surface by probe (b) lateral view of probe

Fig. 4-4 normal vectors of the referenced surface

To describe the surface in the parametric form, we need to find the real touch point onto the surface by a translation with the amount of the probe radius in the direction of the normal vector. Assume the probe radii is denoted r , the touched point is

$$\begin{aligned} P_1' &= P_1 - rN_1 \\ P_2' &= P_2 - rN_2 \end{aligned} \quad (4-2)$$

and the surface description are

$$\begin{aligned} S_1 &\equiv N_{1x}(P_x - P_{1x}') + N_{1y}(P_y - P_{1y}') + N_{1z}(P_z - P_{1z}') = 0 \\ S_2 &\equiv N_{2x}(P_x - P_{2x}') + N_{2y}(P_y - P_{2y}') + N_{2z}(P_z - P_{2z}') = 0 \end{aligned} \quad (4-3)$$

where P_x, P_y, P_z express the coordinate of any point located onto the surface.

By finishing the works above, the WCS can be calculated. In our knowledge, the WCS is represented by the transformation matrix wT_o , which is presented

$${}^wT_o = \begin{bmatrix} {}^wR_o & {}^wP_o \\ 0 & 1 \end{bmatrix} \quad (4-4)$$

First, we observe the geometric features of the cube and the orientation part of the

WCS is

$${}^w R_o = [-N_1, -N_2, N_1 \times N_2] \quad (4-5)$$

For the translation part, a probing-based strategy is implemented by touching the tip of the workpiece to calculate the ${}^w P_o$. It's trivial that the direction of the intersection vector is equal to cross-product of the N_1 and N_2 . We pick an arbitrary point p_0 , and plug into the equation to solve the unknown. Basically, we desire that the resulting point p is as close to the chosen point p_0 as possible. Thus, we could write a distance

$$\|p - p_0\| = (p_x - p_{0x})^2 + (p_y - p_{0y})^2 + (p_z - p_{0z})^2 \quad (4-6)$$

Incorporating the other points in the similar fashion, and writing this constraint using Lagrange multipliers into an objective function results in:

$$J = \|p - p_0\| + \lambda(p - p_1)^2 + \mu(p - p_2)^2 \quad (4-7)$$

Using the standard Lagrange framework, one establishes a nice matrix, in the form:

$$M = \begin{bmatrix} 2 & 0 & 0 & N_{1x} & N_{2x} \\ 0 & 2 & 0 & N_{1y} & N_{2y} \\ 0 & 0 & 2 & N_{1z} & N_{2z} \\ N_{1x} & N_{1y} & N_{1z} & 0 & 0 \\ N_{2x} & N_{2y} & N_{2z} & 0 & 0 \end{bmatrix} \quad (4-8)$$

This matrix can now be used in a system of linear equations

$$M \begin{bmatrix} p_x \\ p_y \\ p_z \\ \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} 2p_{0x} \\ 2p_{0y} \\ 2p_{0z} \\ p_1 \cdot N_1 \\ p_2 \cdot N_2 \end{bmatrix} \quad (4-9)$$

With the linear equation, the vector v goes through the point p is obtained where v is equal to the third column vector of the ${}^w R_o$. We move the probe along v and obtain the triggered TCP position, ${}^w P_p$, as Fig. 4-5 shown. And the ${}^w P$ can be calculate by the

equation.

$${}^w P_o = {}^w P_{tp} - (r+50) \mathbf{v} \quad (4-10)$$

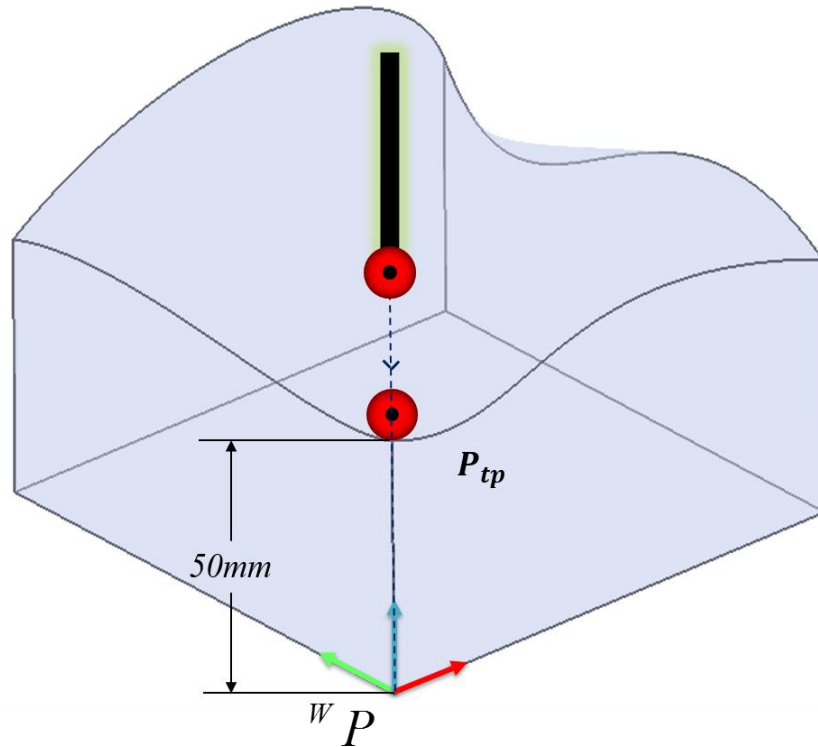


Fig. 4-5 localize the translation of WCS by probing strategy

Finally, the WCS is obtained by the procedure above. And the measured points onto the surface can be determined into the MCS by transformation.

4.3 Algorithm-based Registration

In the section 4.1, we describe an efficient and precise localization method based on geometric characteristics of the workpiece. However, some workpieces do not have such regular shape to do localization by human intuition. These kinds of workpieces usually have its own locators to do localization. A suitable locator is able to fix the position of the workpiece on the CMM platform. Nonetheless, a locator only fits a corresponding workpiece. For the same type of workpieces with different size, shape,

material, a different locator should be designed and manufacture again. Therefore, an algorithm-based localization method is necessary for a CMM to find the WCS by itself.

We implement the Iterative Closet Point (ICP) algorithm to do localization. The algorithm tries to search a transformation matrix which can minimize the difference between two clouds of points as shown in Fig. 4-6. We use the points extracted from the CAD model as the original point set, which is described in section 2.4. The other point set is composed with the position data form probing system. As the number of probing data set is larger than a threshold value, the ICP algorithm is implemented to determine if the transformation error is small enough. If the result is satisfying, the matrix will be recorded and stop the algorithm. If not, just continue probing and implement the algorithm. Ideally, a perfect-matched case leads to zero errors

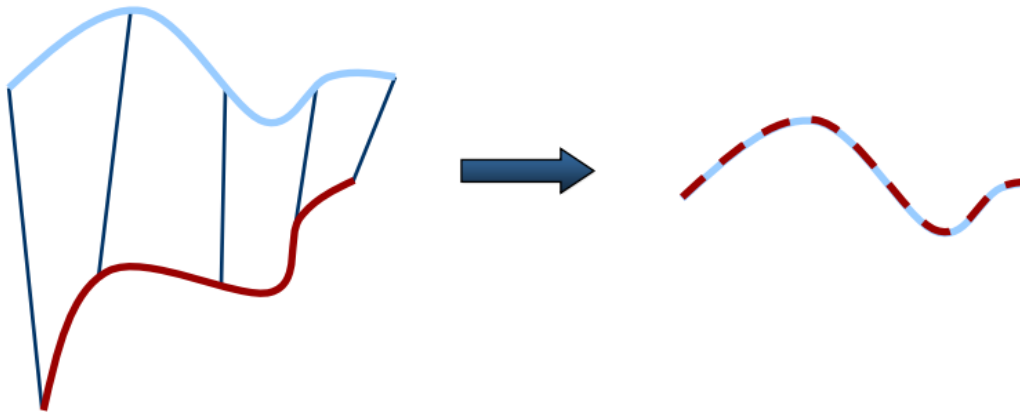


Fig. 4-6 schematic diagram of ICP algorithm

The objective function that describes the difference of two point set can be written as

$$E(\mathbf{R}, \mathbf{T}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - \mathbf{R}p_i - \mathbf{T}\|^2 \quad (4-11)$$

where E means the error combined with original CAD and transformed probing data. x_i

and p_i denote the i -th point of original CAD data set X and transformed probing data set P respectively. N_p is number of probing points. R and T are element that construct the transformation, which denotes rotation part and translation part respectively. In the aspect of optimization, the mathematical derivation is shown as following. The first step is to find the mass center of both points set, then subtract the corresponding mass center from every point in the two point sets.

$$\begin{aligned}\mu_x &= \frac{1}{N_x} \sum_i^{N_x} x_i & X' &= \{x_i - \mu_x\} = \{x_i'\} \\ \mu_p &= \frac{1}{N_p} \sum_i^{N_p} p_i & P' &= \{p_i - \mu_p\} = \{p_i'\}\end{aligned}\quad (4-12)$$

After the data preprocessing is finished, the singular value of the rotation matrix is obtained by Singular Value Decomposition (SVD).

$$W = \sum_i^{N_p} x_i' p_i'^T = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V \quad \sigma_1 \geq \sigma_2 \geq \sigma_3 \quad (4-13)$$

If the $rank(W)=3$, the R and T are solved

$$\begin{aligned}R &= UV^T \\ T &= \mu_x - R\mu_p\end{aligned}\quad (4-14)$$

And the minimum error is calculated

$$E(R, T) = \sum_i^{N_p} (\|x_i'\|^2 + \|y_i'\|^2) - 2(\sigma_1 + \sigma_2 + \sigma_3) \quad (4-15)$$

The procedure above is the solution of traditional ICP algorithm. To improve the efficiency of localization, how to choose good pairs of corresponding points in the two datasets is an important issue. We implement the point sampling method proposed in [32]. This paper focuses on geometrical stability and efficient sampling for minimizing the ICP error. The distribution of the points is analyzed to optimize the sampling result

based on the numerical analysis. Here we show the result of the proposed method in Fig. 4-7.

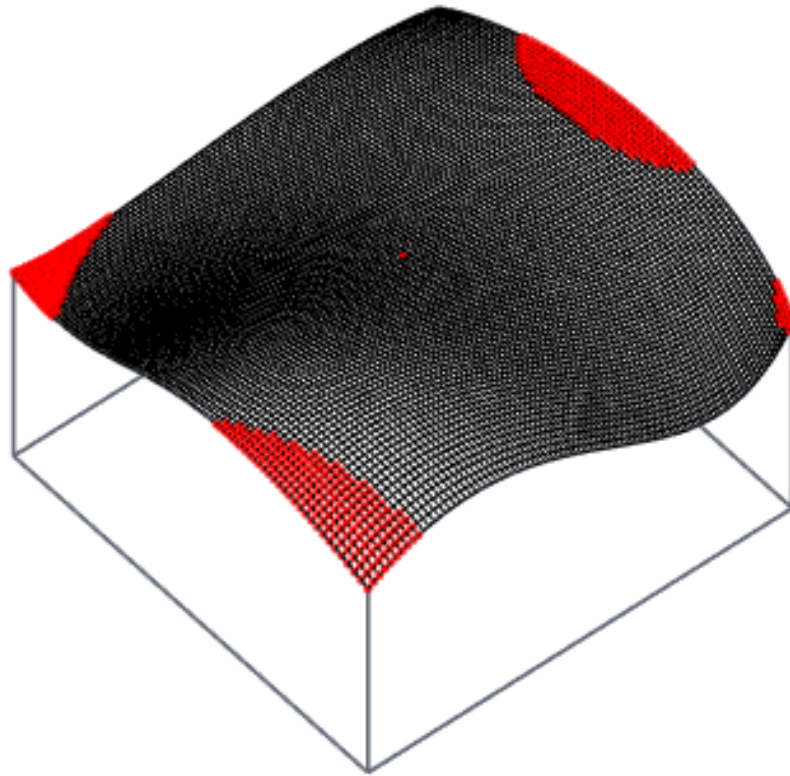
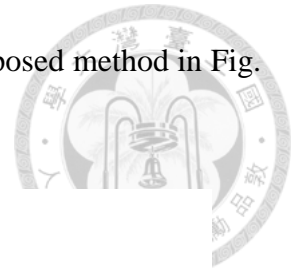


Fig. 4-7 Result of geometrically stable sampling for ICP algorithm

Chapter 5 Freeform Surface Measurement and Validation

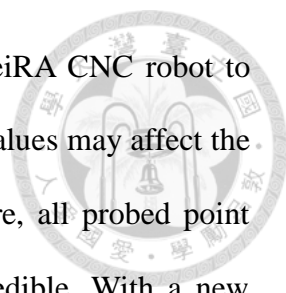


In this chapter, a complete procedure of freeform surface measurement and validation is purposed, which is the core part in this thesis. Moreover, all geometric data in this section are discussed in the machine coordinate system (*MCS*) because of preprocessing done in the previous sections.

5.1 Concept of Proposed Measurement Process

For a traditional CMM with probing system, the application can be mainly divided into two types. One is feature measurement and the other is validation measurement. The former one tries to touch few measured points to observe the geometric characteristic. This application is usually implemented during a machining or fabrication process, and it can provide on-machine real-time inspection information about the state of the machining workpiece. The latter one is used to validate the precision of a workpiece fabricated from CNC machine. Usually more measured points are required for this application because it need a lot of data to improve the credibility of the validation.

In this thesis, we propose a novel freeform surface validation system. The procedure is shown in Fig. 1-6. At first, the four points at the corner of the surface will be probed to build the boundary and build the first reconstructed surface. According the comparison result of the CAD model surface and reconstructed surface, the form-error model is applied to sample the next measured point out. This model aims to search a new single measured point to minimize the difference between the CAD model and the next reconstructed surface. With the new measured point, the probing trajectory is



implemented to plan the most efficient and safety path for the iCeIRA CNC robot to touch the point and obtain real position data. However, some error values may affect the position data feedback because of the physical problems. Therefore, all probed point data should be determined by the error rule until the value is credible. With a new position data feedback, a new reconstructed surface is generated and continue the steps we mention above until the comparison error is lower than a valid threshold value. Finally, the surface validation is implemented to challenge the measured position data and give a final value about how similar the real workpiece is compared with the CAD model.

5.2 Measured point generation

To improve the measurement efficiency, we develop the measured point generator to choose a single point as the measured target from the designed surface S . In this part, two inputs are in needed. The first one is designed surface from the CAD model, which is denoted as S . It also can be taken as the large point set that digitize the designed surface, which is denoted as $S(u, v)$ mentioned in section 2.4. The other input is the measured point set $MP \equiv \{mp_1, mp_2, \dots, mp_n\}$, which is obtained from CMM and probing system. As the number of measured points is more than or equal to 4 ($n \geq 4$), a surface can be reconstructed, which is denoted as $S^*(u, v)$. The number of the points that construct S^* will be the same as the one of S . Therefore both the surface can be compared to earn an error map, which is used to determine the next measured point in the generator. The error map is denoted as $E_S(u, v)$.

$$E_S(u, v) = |S(u, v) - S^*(u, v)| \quad (5-1)$$

We intuitively choose the maximum of E_S as the target, and the corresponding parametric value (u_t, v_t) is easily obtained.

$$(u_t, v_t) = \arg \max_{u \in [0,1], v \in [0,1]} E_s(u, v)$$



Therefore the point $S(u_t, v_t)$ is chosen as the next measured point.

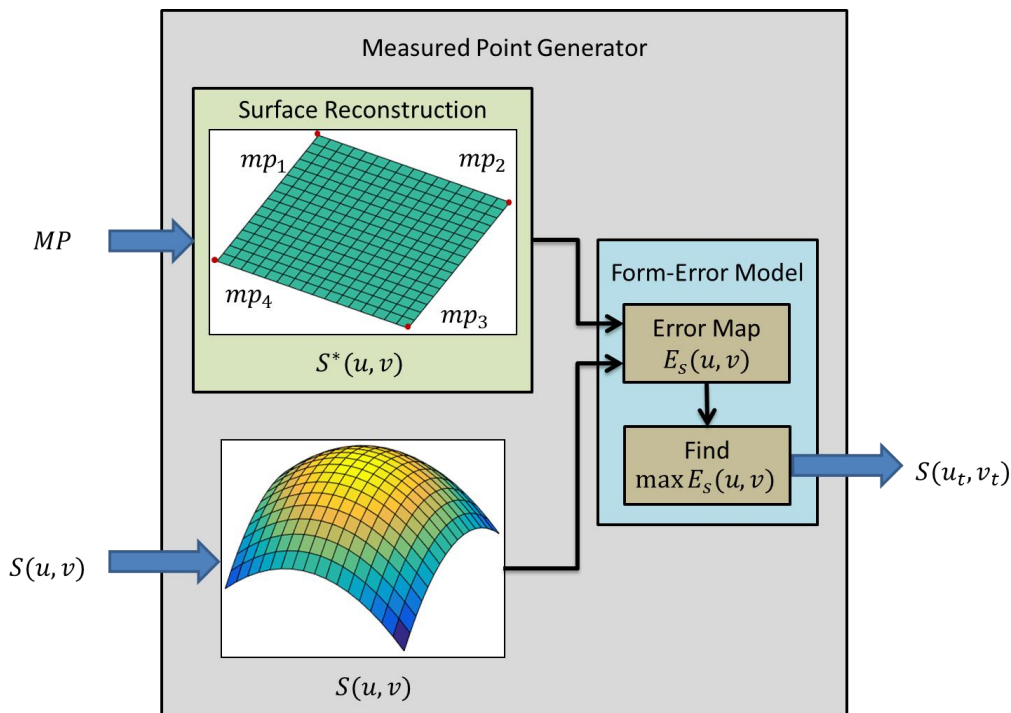
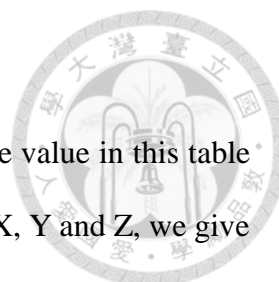


Fig. 5-1 Schematic diagram of the measured point generator

5.3 Working Consideration

To manipulate the CMM safely, a lot of factors should be taken into consideration. Here we discuss the most important two factors for control, joint limit and obstacles in the workspace. Usually the CAD model of the CMM is necessary, but the location of some objects in the space cannot be defined because some parameters are lost, or even cannot be obtained anymore. Therefore we estimate these parameters by rough measurement. And some other parameters in this section is defined to know how to safely control the CMM. We also build a series of error code to inform the user if a motion is valid or not.



5.3.1 Joint Limit Consideration

The joint limit of iCeIRA CNC robot is shown in Table 5-1. The value in this table is deliberately modified into the safe range. For the translation axis, X, Y and Z, we give a small value to prevent the robot trigger the limit sensor and stop the motion. For the rotation axis, A, B, and C, we try to test the flexibility on each axis several times by controlling the rotation. Finally we get the limit values below through observation.

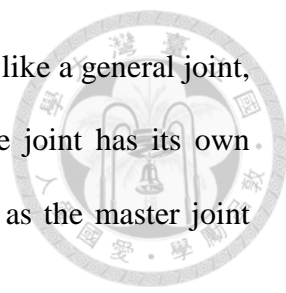
Table 5-1 Joint Limit of iCeIRA CNC robot

	Lower bound	Upper bound
X	40 mm	900 mm
Y	0 mm	890 mm
Z	-45 mm	60 mm
A	-120°	120°
B	-10°	90°
C	-45°	90°

5.3.2 Collision Detection

To reach the functionality of the collision detection, we need to build the 3D model of the CMM robot in advance. We roughly measure the size and physical parameters of the CMM and the tool and observe the link relationship with joints. The model is mainly divided into 6 parts. The parameters about these parts are all added into the simulation environment and are visualized.

The original joints of iCeIRA CNC robot are already built in the simulation and we use it to construct the skeleton model for the fake joints, as shown in Fig. 5-2. These fake joints are employed for presenting the CMM model, and they are visualized in 3D



form. All these fake joints have their own transformation matrix just like a general joint, but they don't provide any degree of freedom anymore. Each fake joint has its own master joint, which means its position and orientation are changed as the master joint rotates.

The mesh-type cube model is implemented as a function to build the CMM 3D model. However, it's possible that a cube cannot fully describe a link of the robot. Maybe some part of CMM is larger than the size we measure so the cube does not cover it all. Therefore we add a small thickness to build the potential field that can ensure the field completely cover the whole hardware.

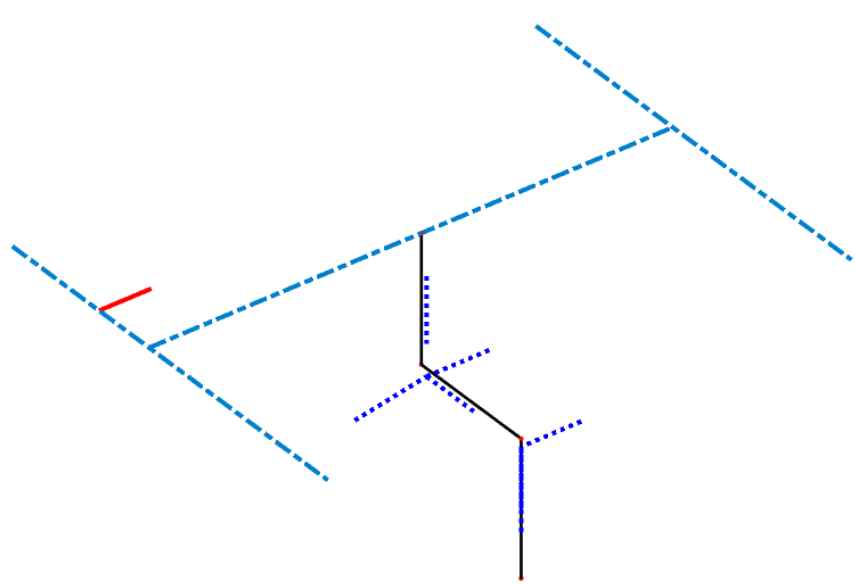


Fig. 5-2 Skeleton of the iCeIRa CNC robot 3D model

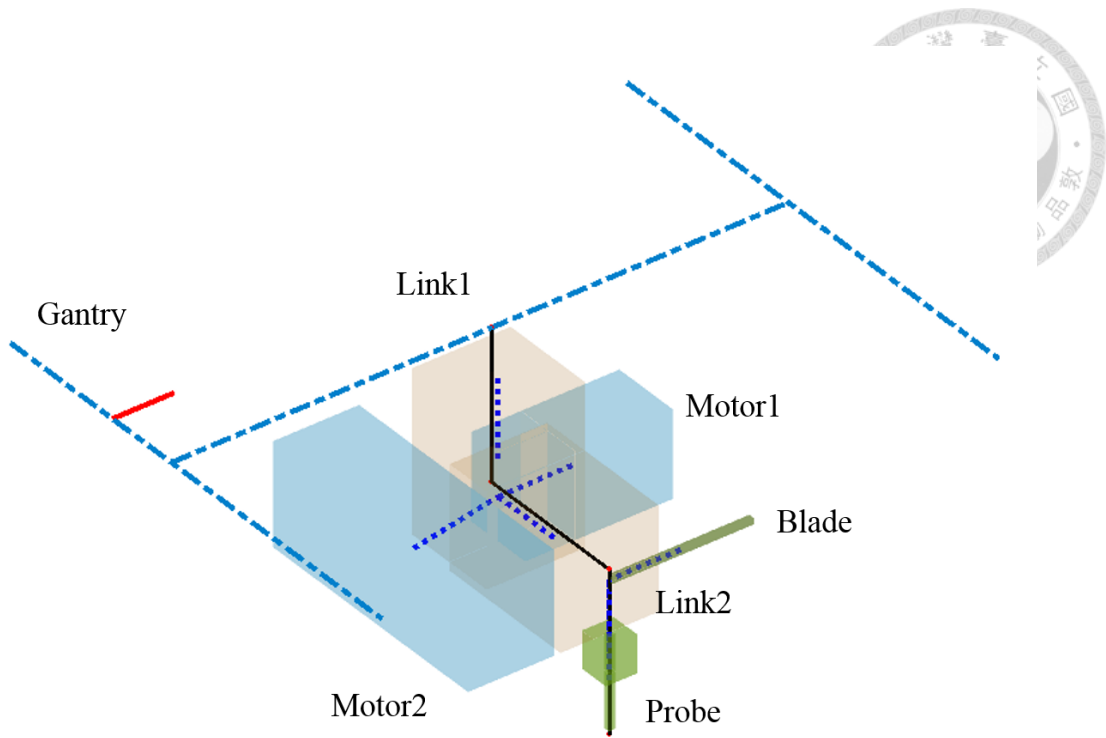
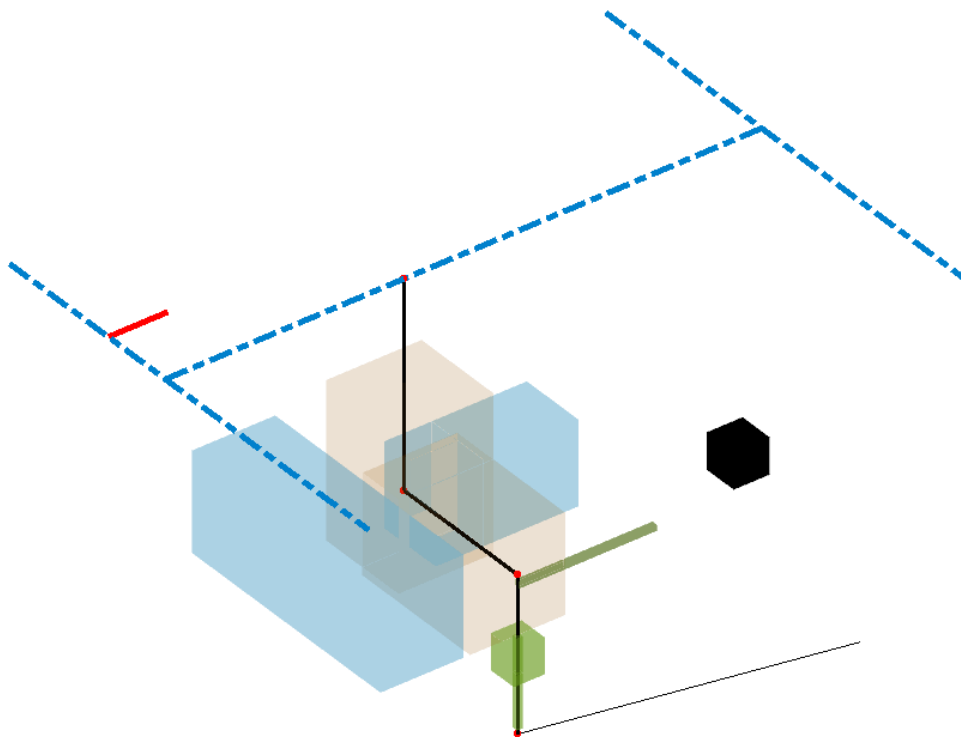
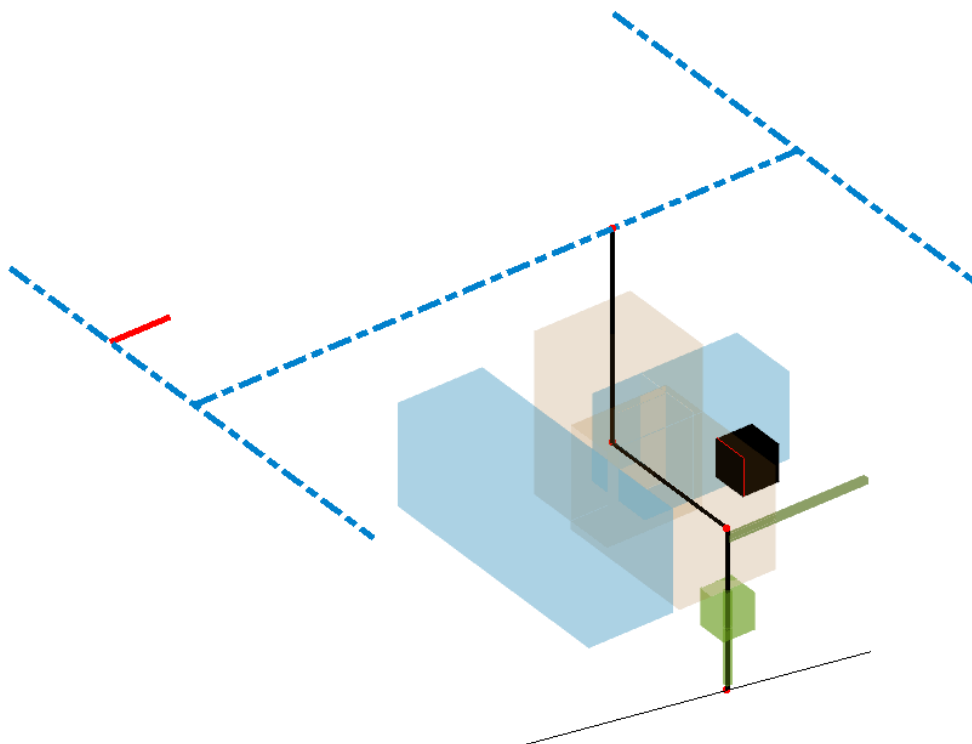


Fig. 5-3 Skeleton of the iCeRa CNC robot

With the 3D model of CMM, the collision detection algorithm is implemented to detect if collisions happen. As we know, both the obstacles and the 3D model of CMM are all built in mesh format. Here we implement the algorithm proposed in [31]. This algorithm is very popular for collision avoidance application because of its high computation speed. Compared with computing the intersection of two primitive shape which may cost a lot of time, it choose only check if two triangles intersect. This method help us quickly help us to check if any obstacles intersect the CMM. Because the obstacles and CMM model are all built by mesh, which are combined with many triangles, we can detect if the collision is happened by checking if an intersection of two triangles exists. Here the detection result is shown in Fig. 5-4.

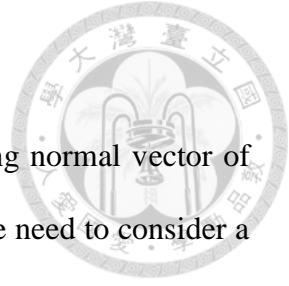


(a) CMM is going to run a given straight path



(b) Collision happened on Link 2 is detected

Fig. 5-4 example of collision detection where the collision happen on Link 2



5.4 Measurement Trajectory Generation

In this section, we will introduce a probing path planning along normal vector of the measured point. To generate a safe and efficient probing path, we need to consider a lot of factors in reality.

5.4.1 Probing Posture Determination

For a measured point, infinite probing postures exist ideally because of high degree of freedom the iCeIRA CNC robot provides. As the Fig. 5-5 shown, with the last three motors that control the orientation of the TCP, any rotation angles along the probing direction are permitted. However, we still need to consider the joint limit and collision problem.

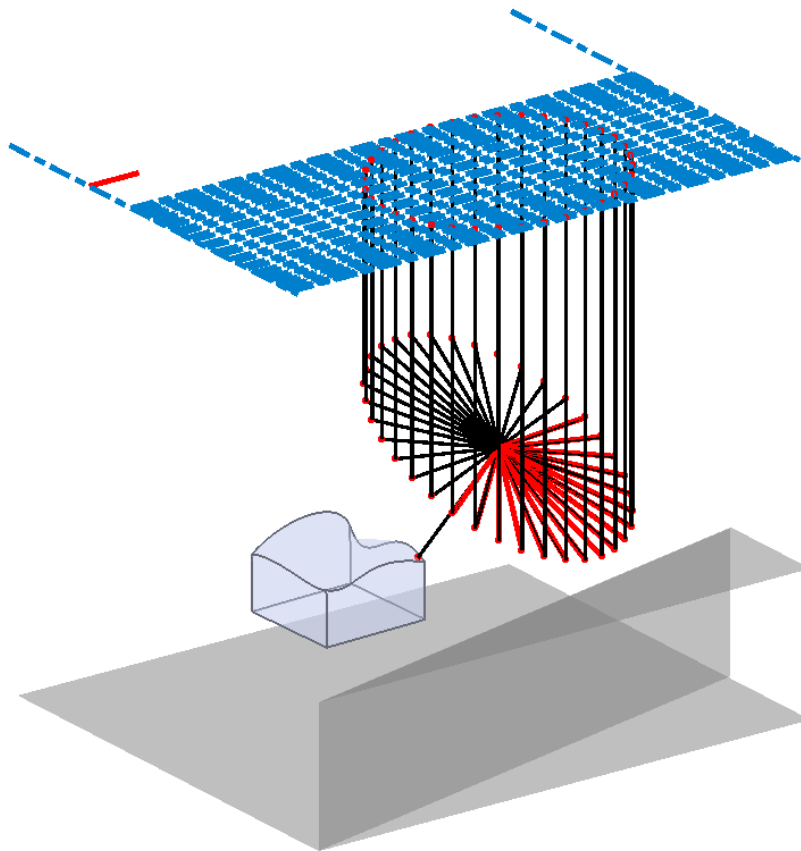
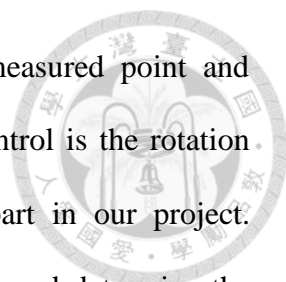


Fig. 5-5 Skeleton of the iCeIRA CNC robot



All postures are determined by the normal vector of the measured point and rotation angle along the normal vector. The only factor we can control is the rotation angle because measurement along normal vector takes a core part in our project. Furthermore, we try to use the Euler angle to present the posture and determine the posture of the next measured point. We try to find the posture which leads to minimum value of the equation (5-3) and choose it as the final result.

$$E_2 = \arg \min_{\theta \in [0, 2\pi]} |E_1 - R_{N_2, \theta} E_2| \quad (5-3)$$

where E_1 means the Euler angle of the former measured point, and E_2 is the one which we need to decide for the current measured point. $R_{N_2, \theta}$ is employed as the rotational matrix that rotates counterclockwise θ degrees along the vector N_2 . The improved efficiency is shown in Fig. 5-6. As we observe, the path is much smoother with the same measured points pair because the rotational difference is minimized.

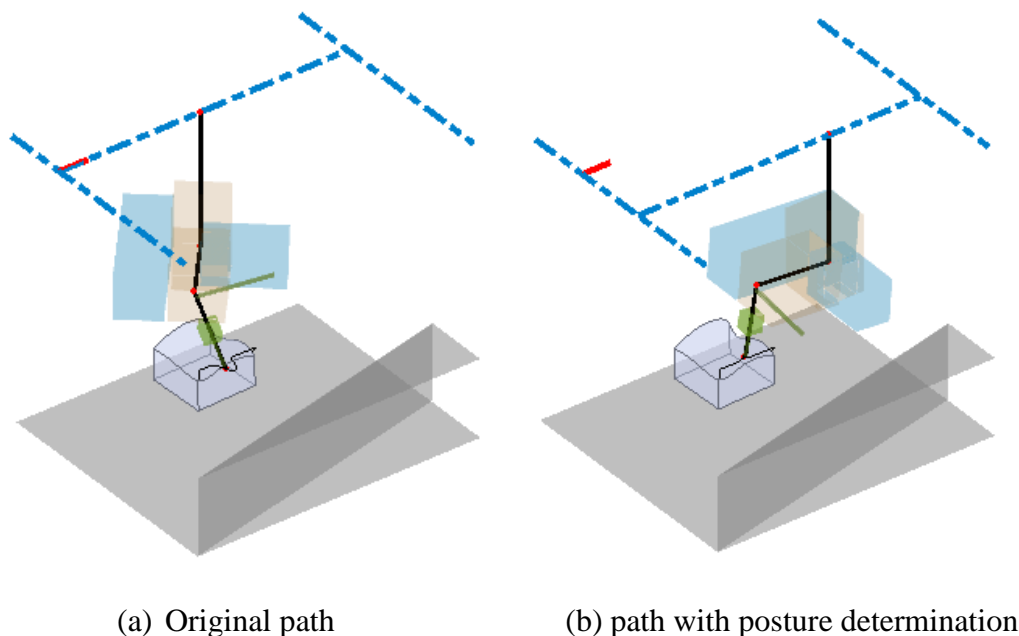
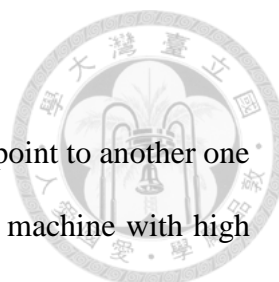


Fig. 5-6 Path with optimized posture determination



5.4.2 Initial Trajectory generation

How to plan a smooth and efficient trajectory from a measured point to another one is a critical problem for coordinate measurement, especially for the machine with high DoF. We develop a measurement trajectory planning strategy based on cooperation of control points and measured points. As the Fig. 5-7 show, a control point should be designed to avoid the collision during the measurement process. Usually the position of a control point is located on the normal direction of the corresponding measured point. Assume the first measured point is called Mp_1 whose orientation (frame) is presented as Mf_1 . Similarly the Mp_2 and Mf_2 denote the second measured point and its frame.

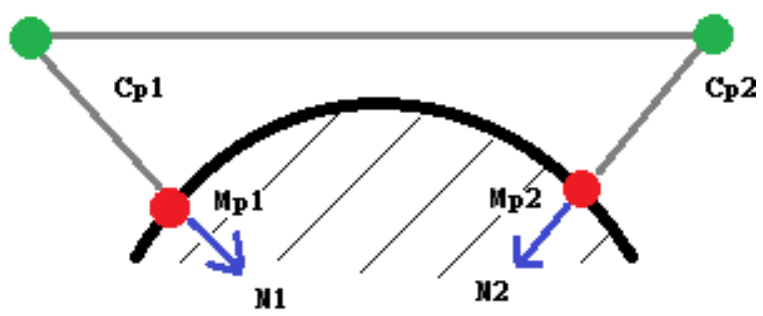
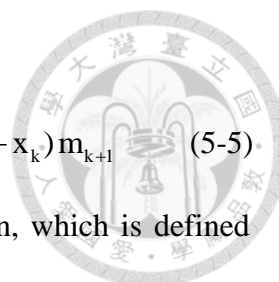


Fig. 5-7 Trajectory planning with control points

Generally, the path is divided into three segments, Mp_1 to Cp_1 , Cp_1 to Cp_2 , and Cp_2 to Mp_2 . To generate a complete path for the CMM, the cubic hermite spline is implemented to interpolate the points onto the trajectory. It has the property that provides the unique third-degree polynomial path between two points. The mathematical expression is shown.

$$p(t) = (2t^3 - t^2 + 1)p_0 + (t^3 - 2t^2 + t)m_0 + (-2t^3 + 3t^2)p_1 + (t^3 - t^2)m_1 \quad (5-4)$$

where p_0 is the starting point at $t = 0$ and an ending point p_1 at $t = 1$ with starting tangen m_0 at $t = 0$ and ending tangent m_1 at $t = 1$. This is the fundamental expression of the cubic hermite spline, and the other type of expression is shown for the



interpolation on an arbitrary interval.

$$p(x) = h_{00}(t)p_k + h_{10}(t)(x_{k+1} - x_k)m_k + h_{01}(t)p_{k+1} + h_{11}(t)(x_{k+1} - x_k)m_{k+1} \quad (5-5)$$

where $t = (x - x_k)/(x_{k+1} - x_k)$ and h refers to the basis function, which is defined as following. Any $p(x)$ is the linear combination of these basis functions.

$$\begin{aligned} h_{00}(t) &= 2t^3 - t^2 + 1 \\ h_{10}(t) &= t^3 - 2t^2 + t \\ h_{01}(t) &= -2t^3 + 3t^2 \\ h_{11}(t) &= t^3 - t^2 \end{aligned} \quad (5-6)$$

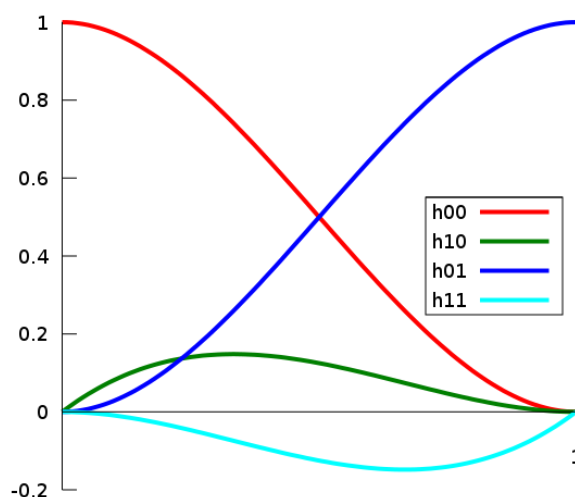


Fig. 5-8 The four Hermite basis functions

The number of interpolated points is related to the velocity curve, distance, time. Therefore usually the less points are created onto the first and third segments compared with the second segment. All segments are straight lines in the 3D space. However, some problems are indicated as shown in Fig. 5-9. The first one is that joint A is out of joint limit and second one is that collision is happened on the probe. We will solve these problem in the next sub-section. We show the joint limit problem in Fig. 5-10, where the limit of joint A is $[-120^\circ, 120^\circ]$.

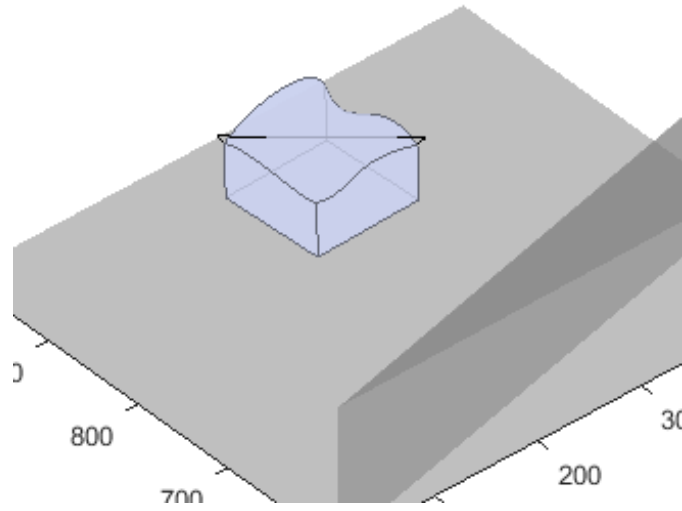
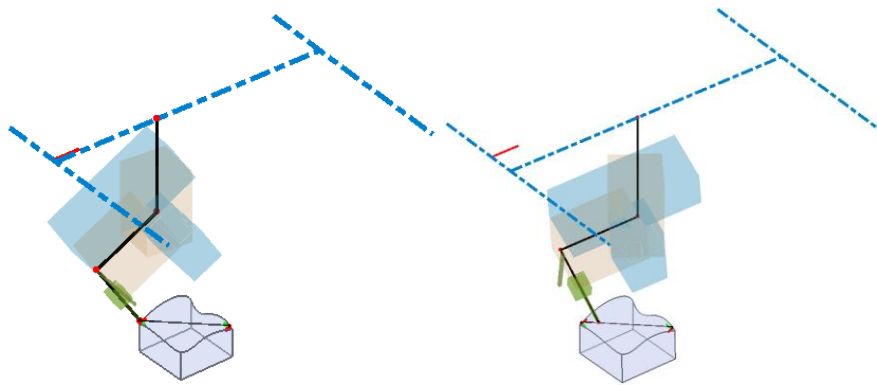
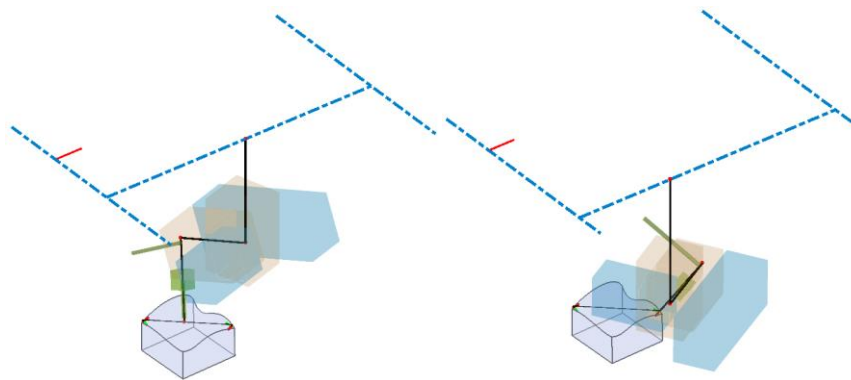


Fig. 5-9 Initial measurement path



(a) TCP reaches Cp_1 ($\theta_4=-95.25$)

(b) joint A is out of limit ($\theta_4=-124.47$)



(c) joint A is out of limit ($\theta_4=-124.47$)

(d) TCP reaches Cp_2 ($\theta_4=119.63$)

Fig. 5-10 CMM motion on initial path



5.4.3 Joint Limit Response

The joint limit problem is taken into consideration for the path planning. We divide the reason that cause joint limit into two aspect, orientation and translation.

First we check orientation because the workspace of translation affects less in our case, where the orientation part depends on A, B and C axes only. The concept is that we believe it's a truth that any posture is feasible for the CMM because of high DoF. Therefore, the core problem is only related to the rotation direction of the joint in our analysis. As the Fig. 5-11 shows, because the interpolation only outputs the interval values from starting angle to target angle, it cannot consider the directionality.

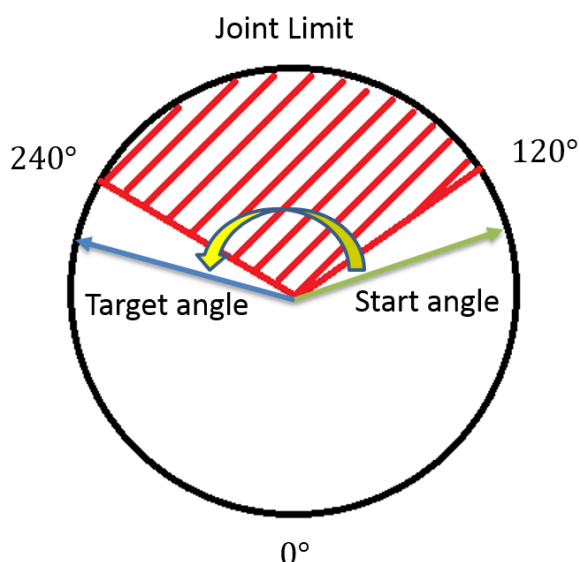


Fig. 5-11 Interpolation problem of joints for orientation part

We implement a detector, which can know which joint has the problem mentioned above in this path. A contrary orientation with the corresponding path will be generated then replace the original path. That's it, the joint limit of the orientation part is solved.

For translation part, X, Y, Z axes, the problem is simpler and more intuitive. The reason is only the IK solution exceeds the joint limits, and can be solved by

$$\delta = \delta - \min(\delta - \delta_{UB}, \delta - \delta_{LB}) \quad \delta = \{X, Y, Z\} \quad (5-7)$$

Here we shows the result of modified path which is affected by joint limit.

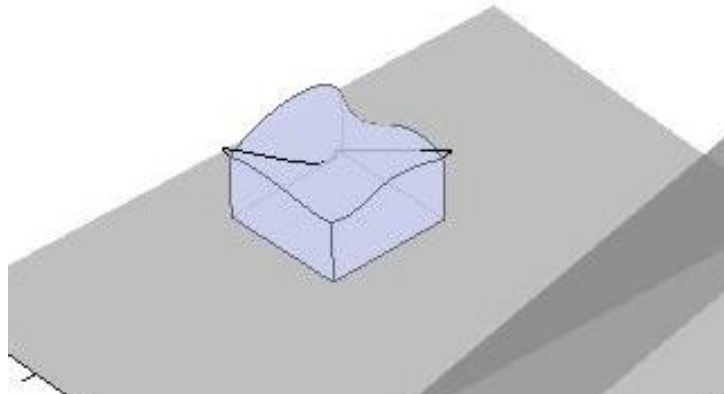
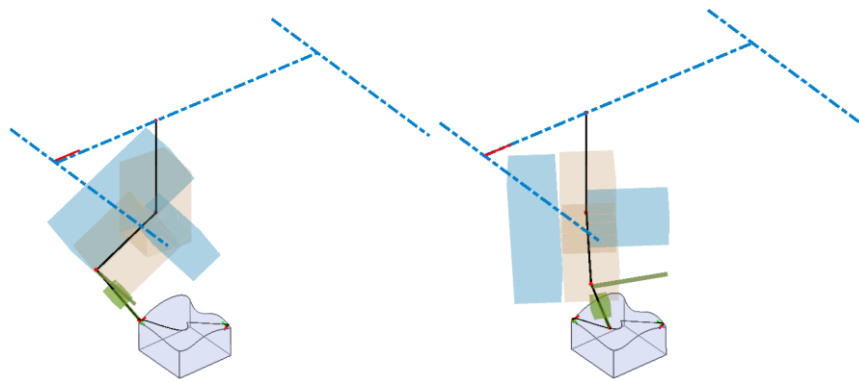
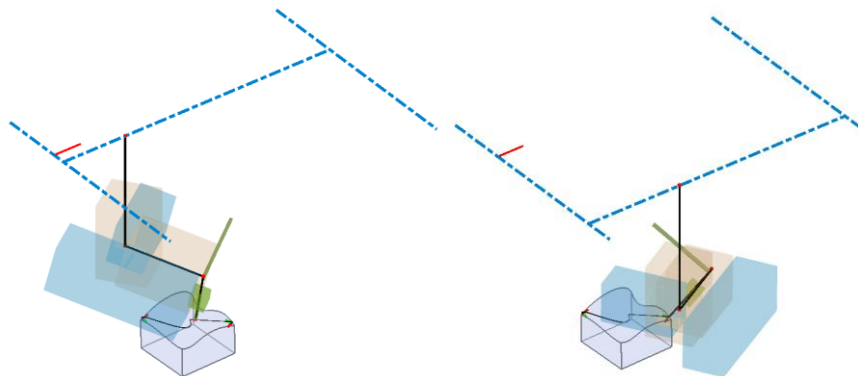


Fig. 5-12 Path modified by joint limit



(a) TCP reaches Cp_1 ($\theta_4=-95.25$) (b) CMM motion is valid but unsafe($\theta_4=-26.64$)



(b) CMM motion is valid but unsafe ($\theta_4=41.97$) (d) TCP reaches Cp_2 ($\theta_4=119.63$)

Fig. 5-13 CMM motion on path with joint limit response

However, the collision is happened on the probe during the motion. The solution is introduced in the next section.



5.4.4 Collision Response

To avoid the collision during the CMM motion, we implement the Configuration space (C-space) with the Depth-based Search (DFS) algorithm to search a safe trajectory to replace the collision path.

A C-space is a physical system that can indicate the space occupied by the robot and obstacles in the joint space. Let q denote a point in a configuration space Q , and denote the workspace obstacles as WO_i . Therefore configuration space obstacle QO_i is the set of configuration spaces q at which the robot intersects WO_i , that is

$$QO_i = \{q \in Q \mid R(q) \cap WO_i \neq \emptyset\} \quad (5-8)$$

Therefore the remained configuration space is that permits robot motion, which is called free configuration space (free space) Q_{free} .

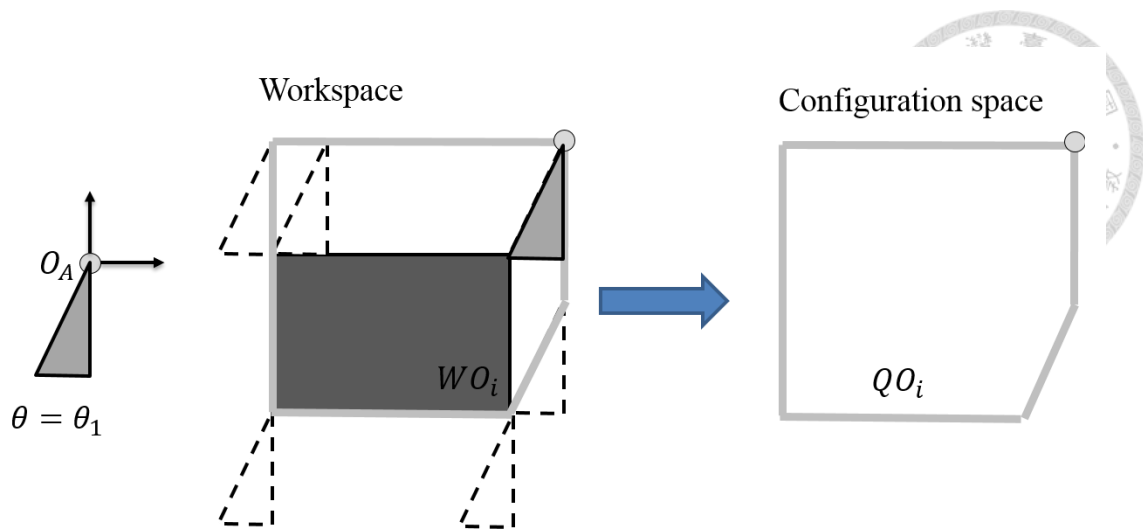
$$Q_{free} = Q \setminus (\cup QO_i) \quad (5-9)$$

Usually the polygonal representation is a great way discuss the C-space, here we take the robot translating in 2D space as example, where the mobile robot also has a rotational DoF. The Minkowski sum is used to describe the configuration space between a robot and an obstacle. The Minkowski sum of two sets P and Q , denoted by $P \oplus Q$, is defined as

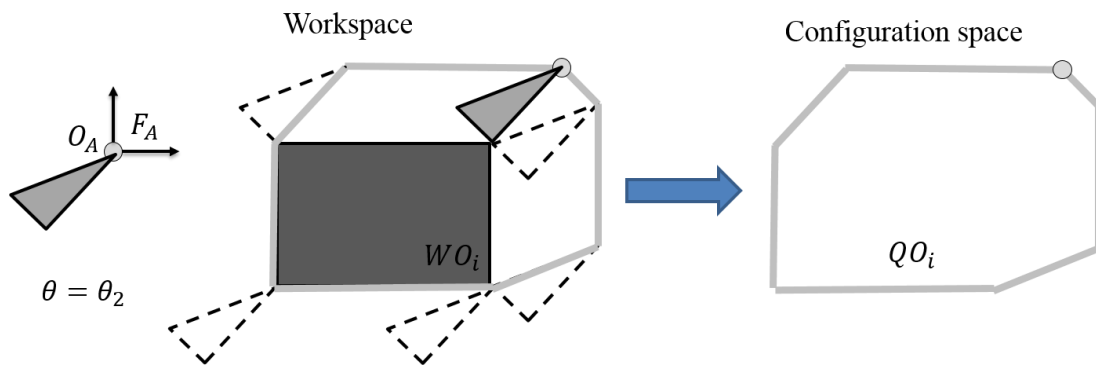
$$P \oplus Q = \{p+q \mid p \in P, q \in Q\} \quad (5-10)$$

Similarly, the Minkowski difference is defined as

$$P \ominus Q = \{p - q \mid p \in P, q \in Q\} \quad (5-11)$$



(a) Workspace to C-space when robot's $\theta = \theta_1$



(b) Workspace to C-space when robot's $\theta = \theta_2$

Fig. 5-14 Example of workspace to C-space for a 2D translating robot

By the concept above, we can also build a C-space for the iCeIRA CNC robot, where the obstacles are shown in Fig. 2-14. All the objects in the simulation workspace are taken as obstacles, including the ground and the measured object. To save the computation cost, the measured surface is simplified as the shown in Fig. 5-15.

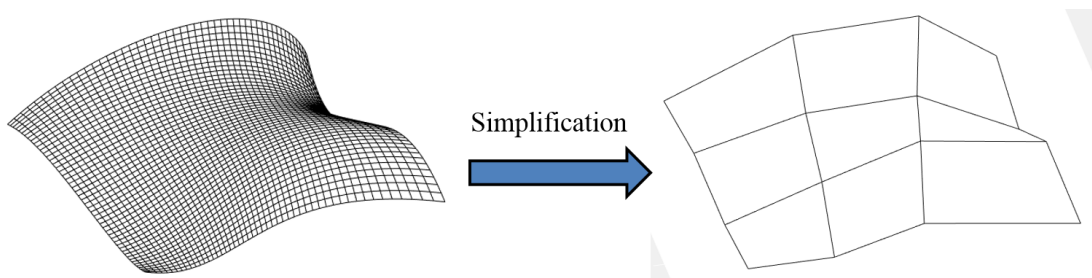
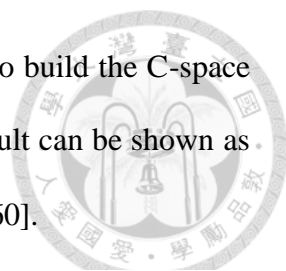


Fig. 5-15 Simplified measured surface which is used for collision avoidance



Here we introduce another simple case that uses a 2DoF robot arm to build the C-space with a simple obstacle. The C-space is equal to DoFs, so that the result can be shown as.

Fig. 5-16. The joint limit for α is $[0,180]$ and the one for β is $[0,360]$.

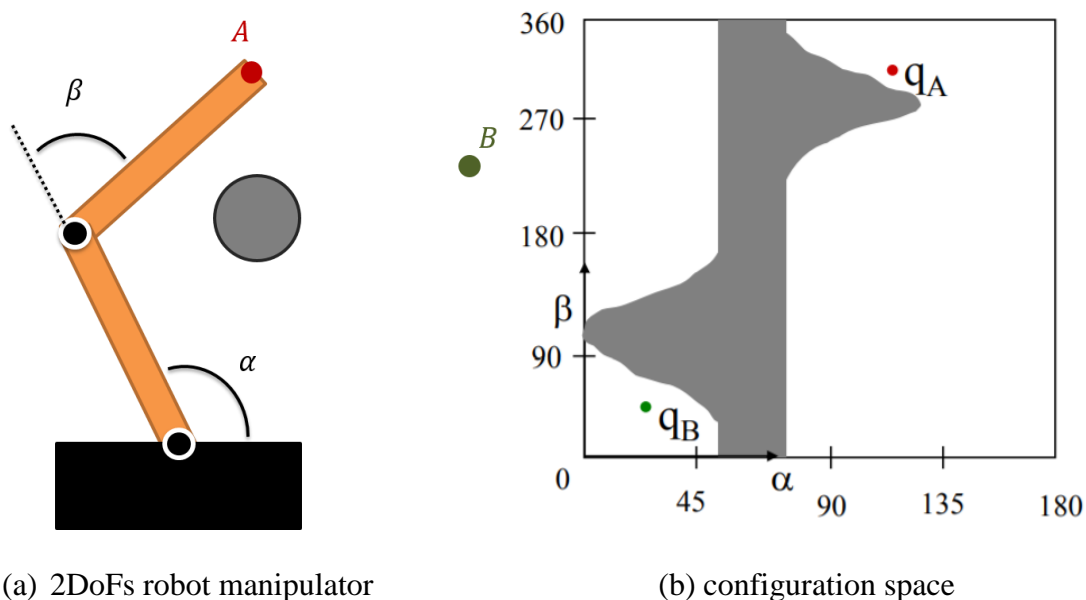


Fig. 5-16 C-space example for a simple 2DoFs robot manipulator

Finally the DFS algorithm is implemented to find the shortest path that can reach the target point without collision. And we also show the result implemented on the iCeIRA CNC robot in the simulation.

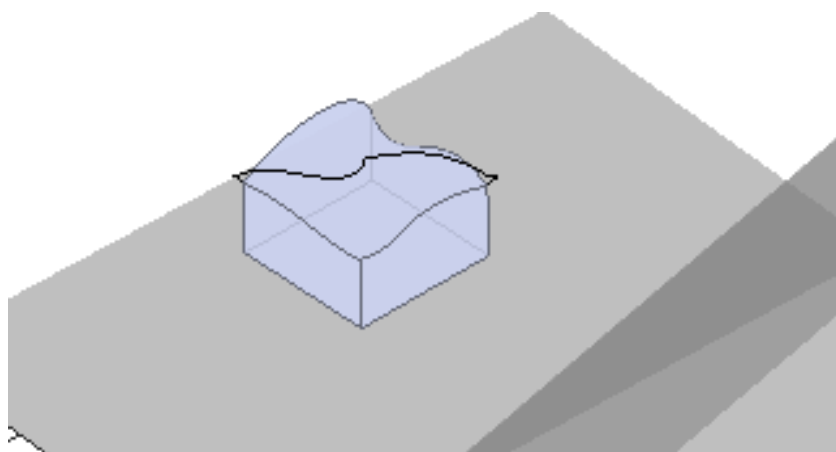
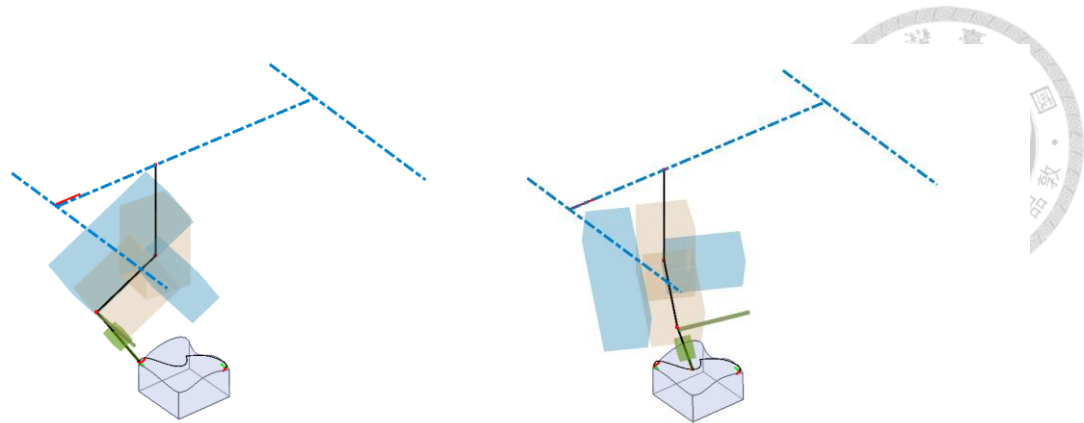
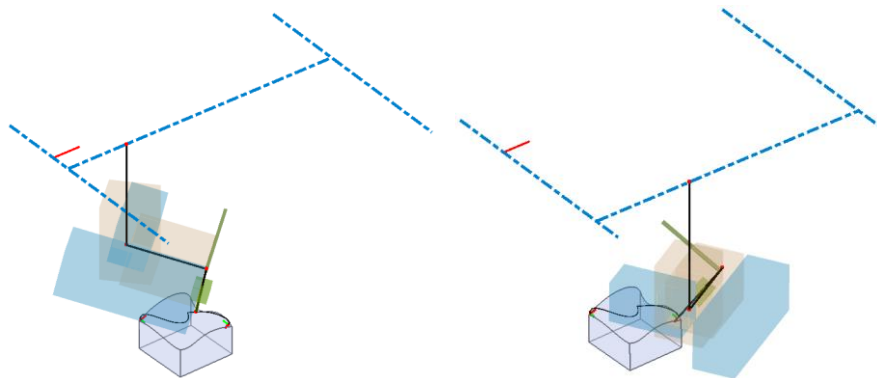


Fig. 5-17 Path modified with consideration of joint limit and collision



(b) TCP reaches Cp_1 ($\theta_4=-95.25$) (b) CMM motion is valid and safe ($\theta_4=-33.50$)



(d) CMM motion is valid and safe ($\theta_4=33.51$) (d) TCP reaches Cp_2 ($\theta_4=119.63$)

Fig. 5-18 CMM motion on path with joint limit and collision response

5.5 Error Consideration

During the measurement operation, we need to consider some errors that may happened physically. In fact, countless error exists in the real world. No matter how we human try to eliminate the error, it's also hard that the error can be lower than $0.01\mu m$ for a machine. Nonetheless, for a probe mounted on the end-effector, a lot of error are brought because the sensor precision is still a big problem. In [15], it discuss a lot of error types that may affect the data precision, as the Fig. 5-19 shows. Here we try to solve two errors that make a great effect during the measurement.

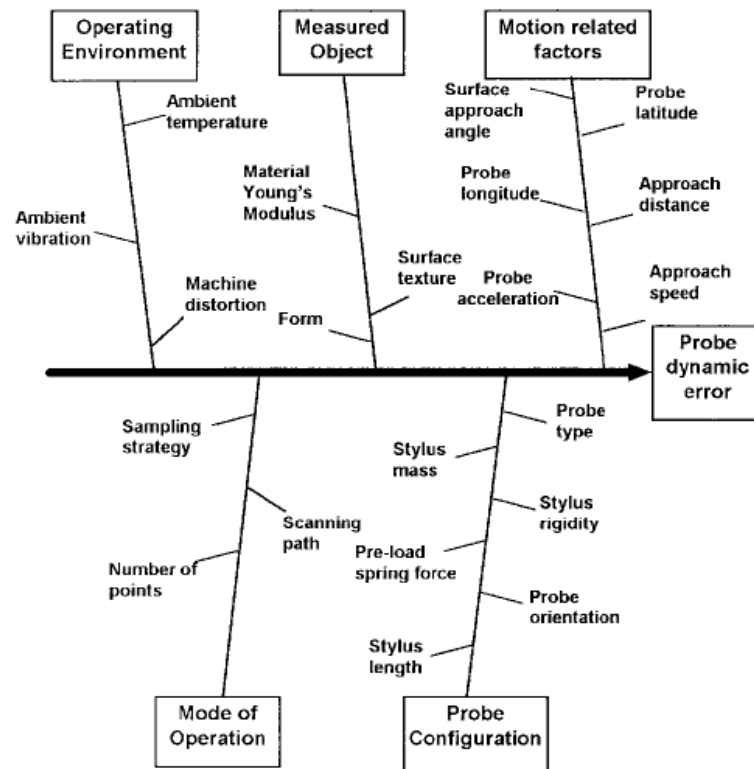


Fig. 5-19 Error in the measurement process

5.5.1 Pretraveling problem

Pretraveling is a physical error leads to inaccuracy of the probing data, which is usually caused by the inaccurate localization or low stiffness of measured object. In our probing strategy, we try to that the end point of the path is equal to measured point. This means the velocity of the probe should be zero ideally while the measured point is touched. Therefore we design a detector the acquire the velocity value during the third of the measurement path, which only covers the path from Cp_2 to Mp_2 . If the velocity is not decreased to a satisfying value when the probe is triggered, we will try to run the path again with lower velocity.

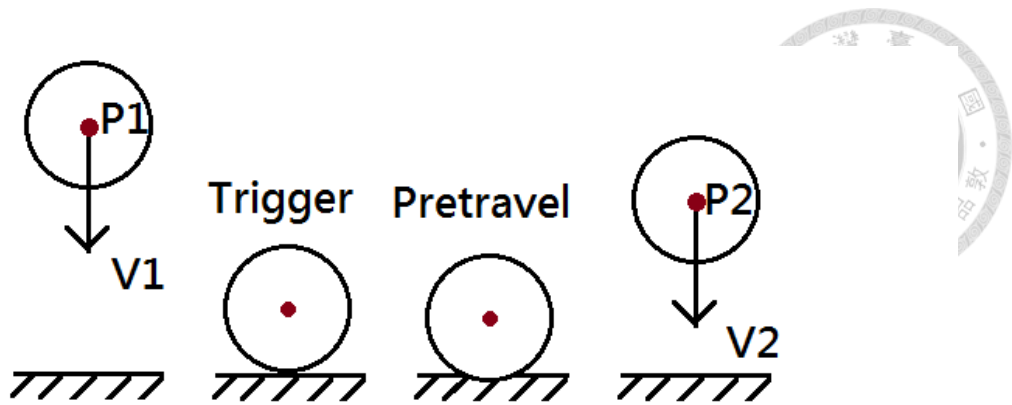


Fig. 5-20 schematic diagram of pretraveling problem

5.5.2 Radius Compensation

Radius compensation aims to reduce the error of data by deducing the radius of the probe from the TCP data to obtain real point position that the probe touches. It seems to be a simple problem but not anymore. It's almost an insoluble problem in the measurement industry. The most important problem is that how to know the correct and precise compensation vector, as the Fig. 5-21. Too many factors can affect the direction of the vector, so that we cannot find the vector precisely. In our case, we simply use the equation (5-12) to do the compensation.

$$p' = p + rv \quad (5-12)$$

where p is the TCP points that obtain by triggering the probe, and p' is the real point that probe touches the surface. The probe radius is denoted as r . And the v is the compensation vector, which is equal to the normal vector of the measured point.

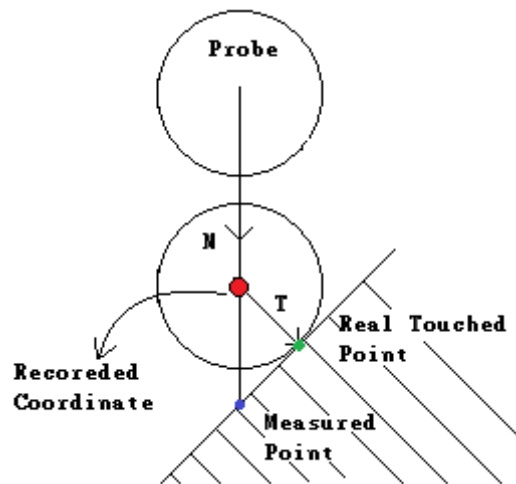


Fig. 5-21 schematic diagram of radius compensation

5.6 Surface Validation

Surface validation is the final step in the measurement process we propose. It aims to quantify how different is between the real workpiece and designed CAD model. In our proposed method, we can't directly trust the correctness of probing data because a lot of errors must exist to disturb the result. First we consider the uncertainty brought from the localization. Compared with the reference-based localization method, algorithm-based localization may lead to more error in the process. Moreover, because we use the surface reconstruction method to achieve the validation, we should also take the reconstruction error into consideration. We try to use the variance of them to decrease the effect caused by the error. As Fig. 5-22 shows, the variations get larger and larger during the process, which means more and more unknown appear.

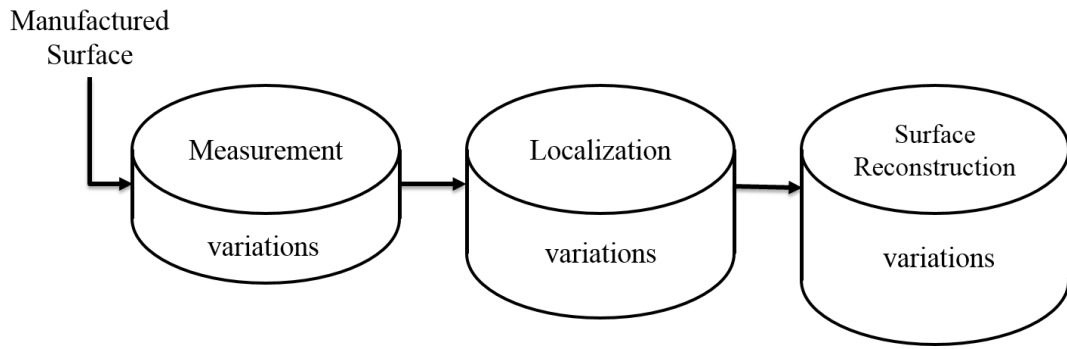


Fig. 5-22 schematic diagram Error caused by different reasons

As we mentioned above, we try to explain the uncertainty that caused by the localization and surface reconstruction. We briefly take them as the variation caused by Gaussian noise and try to theoretically explain the relationship between the measured data and real workpiece. The concept is shown in

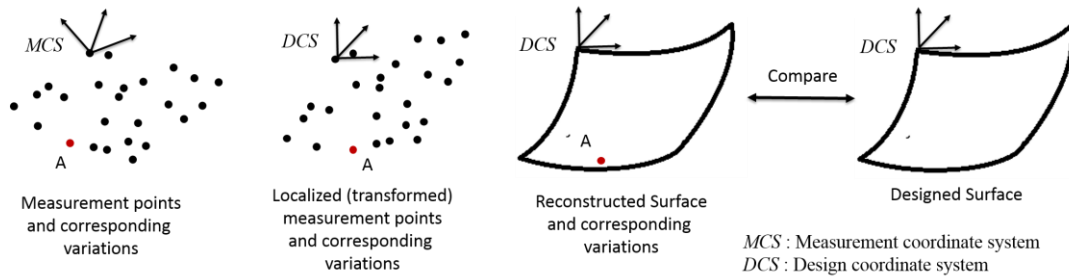


Fig. 5-23 Freeform surface inspection considering the uncertainties in measurement, localization and surface reconstruction

Here we briefly use the equation (13) to represent the measurement error at i -th measurement point by

$$E_i^M = P_i^M - P_i^P \quad (5-13)$$

where E_i^M means a 3×1 vector with the measurement error in X, Y and Z direction, P_i^M is a 3×1 vector with measured X, Y and Z coordinates of the i -th measured point. P_i^P is a 3×1 vector with the true X, Y and Z coordinates of the i -th measured point on the real surface. We will talk about the uncertainties that may affect the equation



(5-13) which leads to more accurate validation result.

5.6.1 Localization Uncertainties

In our case, to transform the *WCS* to *MCS*, a transformation matrix called T_L is used.

Assume there are m measured points are digitized on the manufactured surface, and each of them can be described as a homogeneous coordinate in *WCS*.

$$r^M = \begin{bmatrix} x_0^M & x_1^M & \dots & x_{m-1}^M \\ y_0^M & y_1^M & \dots & y_{m-1}^M \\ z_0^M & z_1^M & \dots & z_{m-1}^M \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (5-14)$$

And we can establish the relationship between the localized measured point and measured points as following.

$$r^L = r^D + \varepsilon = T_L r^M \quad (5-15)$$

Usually we know a transformation matrix T_L has six parameters, including translation (D_x, D_y, D_z) and rotation ($\theta_x, \theta_y, \theta_z$), and T_L can be written as.

$$T_L = DR_x R_y R_z = \begin{bmatrix} c_y c_z & -c_y s_z & s_y & D_x \\ c_z s_x s_y + c_x s_z & -s_x s_y s_z + c_x c_z & -c_y s_x & D_y \\ -c_x c_z s_y + s_x s_z & c_x s_y s_z + c_z s_x & c_x c_y & D_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-16)$$

where R_x , R_y and R_z are rotation matrices about the X , Y and Z axes, and D is the translation matrix, c_γ and s_γ stand for $\cos(\theta_\gamma)$ and $\sin(\theta_\gamma)$ separately. Therefore the coordinate of the localized measurement points x_i^L , y_i^L and z_i^L can be obtained.

$$\begin{aligned} x_i^L &= c_y c_z x_i^M - c_y s_z y_i^M + s_y z_i^M + D_x \\ y_i^L &= (c_z s_x s_y + c_x s_z) x_i^M + (-s_x s_y s_z + c_x c_z) y_i^M - c_y s_x z_i^M + D_y \\ z_i^L &= (-c_x c_z s_y + s_x s_z) x_i^M + (c_x s_y s_z + c_z s_x) y_i^M + c_x c_y z_i^M + D_z \\ i &= 0, 1, \dots, m-1 \end{aligned} \quad (5-17)$$

The sum of squared errors between localized measurement points and their corresponding points on the design surface, L , can be calculated

$$L = \sum_{i=0}^{m-1} [(x_i^L - x_i^D)^2 + (y_i^L - y_i^D)^2 + (z_i^L - z_i^D)^2] \quad (5-18)$$



In another words, we also can obtain the variances of the localized coordinates of the measurement points to express the uncertainties, which is calculated by (Meyer 1975).

$$\begin{aligned} \text{var}(x_i^L) &= \left(\frac{\partial x_i^L}{\partial \theta_x}\right)^2 \text{var}(\theta_x) + \left(\frac{\partial x_i^L}{\partial \theta_y}\right)^2 \text{var}(\theta_y) + \left(\frac{\partial x_i^L}{\partial \theta_z}\right)^2 \text{var}(\theta_z) + \left(\frac{\partial x_i^L}{\partial D_x}\right)^2 \text{var}(D_x) \\ &+ \left(\frac{\partial x_i^L}{\partial x_i^M}\right)^2 \text{var}(x_i^M) + \left(\frac{\partial x_i^L}{\partial y_i^M}\right)^2 \text{var}(y_i^M) + \left(\frac{\partial x_i^L}{\partial z_i^M}\right)^2 \text{var}(z_i^M) \\ &= (-c_z s_y x_i^M + s_y s_z y_i^M + c_y z_i^M)^2 \text{var}(\theta_y) + (-c_y s_x x_i^M - c_y c_z y_i^M)^2 \text{var}(\theta_z) \\ &+ \text{var}(D_x) + (c_y c_z)^2 \text{var}(x_i^M) + (-c_y s_z)^2 \text{var}(y_i^M) + s_y^2 \text{var}(z_i^M) \end{aligned} \quad (5-19)$$

$$\begin{aligned} \text{var}(y_i^L) &= \left(\frac{\partial y_i^L}{\partial \theta_x}\right)^2 \text{var}(\theta_x) + \left(\frac{\partial y_i^L}{\partial \theta_y}\right)^2 \text{var}(\theta_y) + \left(\frac{\partial y_i^L}{\partial \theta_z}\right)^2 \text{var}(\theta_z) + \left(\frac{\partial y_i^L}{\partial D_y}\right)^2 \text{var}(D_y) \\ &+ \left(\frac{\partial y_i^L}{\partial x_i^M}\right)^2 \text{var}(x_i^M) + \left(\frac{\partial y_i^L}{\partial y_i^M}\right)^2 \text{var}(y_i^M) + \left(\frac{\partial y_i^L}{\partial z_i^M}\right)^2 \text{var}(z_i^M) \\ &= ((c_x c_z s_y - s_x s_z) x_i^M + (-c_x s_y s_z - c_z s_x) y_i^M - c_x c_y z_i^M)^2 \text{var}(\theta_x) \\ &+ (c_y c_z s_x x_i^M - c_y s_x s_z y_i^M - s_x s_y z_i^M)^2 \text{var}(\theta_y) \\ &+ ((-s_x s_y s_z - c_x c_z) x_i^M + (-c_z s_x s_y - c_x s_z) y_i^M)^2 \text{var}(\theta_z) \\ &+ \text{var}(D_y) + (c_z s_x s_y + c_x s_z)^2 \text{var}(x_i^M) \\ &+ (-s_x s_y s_z + c_x c_z)^2 \text{var}(y_i^M) + (-c_y s_x)^2 \text{var}(z_i^M) \end{aligned} \quad (5-20)$$

$$\begin{aligned} \text{var}(z_i^L) &= \left(\frac{\partial z_i^L}{\partial \theta_x}\right)^2 \text{var}(\theta_x) + \left(\frac{\partial z_i^L}{\partial \theta_y}\right)^2 \text{var}(\theta_y) + \left(\frac{\partial z_i^L}{\partial \theta_z}\right)^2 \text{var}(\theta_z) + \left(\frac{\partial z_i^L}{\partial D_z}\right)^2 \text{var}(D_z) \\ &+ \left(\frac{\partial z_i^L}{\partial x_i^M}\right)^2 \text{var}(x_i^M) + \left(\frac{\partial z_i^L}{\partial y_i^M}\right)^2 \text{var}(y_i^M) + \left(\frac{\partial z_i^L}{\partial z_i^M}\right)^2 \text{var}(z_i^M) \\ &= ((c_z s_x s_y + c_x s_z) x_i^M + (-s_x s_y s_z + c_x c_z) y_i^M - c_y s_x z_i^M)^2 \text{var}(\theta_x) \\ &+ (-c_x c_y s_z x_i^M - c_x c_y s_z y_i^M - c_x s_y z_i^M)^2 \text{var}(\theta_y) \\ &+ ((c_x s_y s_z + c_z s_x) x_i^M + (c_x c_z s_y - s_x s_z) y_i^M)^2 \text{var}(\theta_z) \\ &+ \text{var}(D_z) + (c_z s_x s_y + c_z s_x)^2 \text{var}(x_i^M) \\ &+ (c_x s_y s_z + c_z s_x)^2 \text{var}(y_i^M) + (c_x c_y)^2 \text{var}(z_i^M) \end{aligned} \quad (5-21)$$

$$i = 0, 1, \dots, m-1$$

Covariance terms are not considered since each X, Y, Z coordinates of a measurement point is assume to be independent of other coordinate. Since all the m measurement



points are used to calculate the six transformation parameters, the influence of only one measurement point on the localization transformation parameters is insignificant.

5.6.2 Reconstruction Uncertainty

The reconstruction uncertainty is related to the reconstruction method. Here we implement the classical B-Spline reconstruction. The uncertainties of reconstruction are as following.

Assume the real position of a point on the manufactured surface is P_i^P and the measured point is P_i^M . After localization, these two points are transformed into P_i^T and P_i^L respectively, as shown in Fig. 5-24.

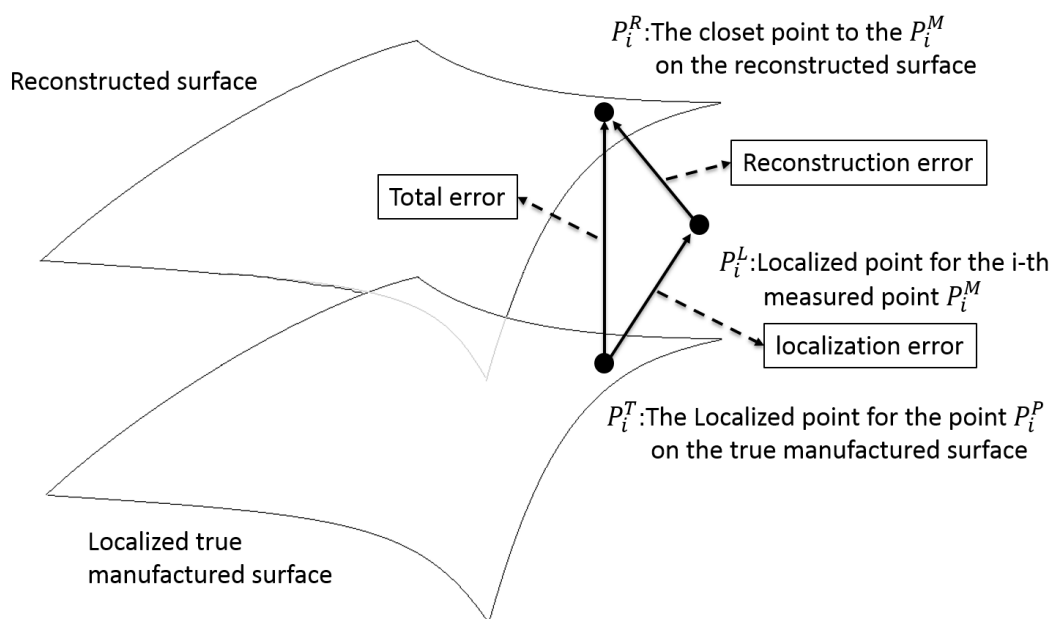


Fig. 5-24 Reconstruction error and localization error

The Cartesian components of the localized measurement error at the i -th measurement point can be defined as follows:



$$\begin{cases} E_{X_i}^L = (P_{X_i}^L - P_{X_i}^T) \sim N(0, V_{X_i}^L) \\ E_{Y_i}^L = (P_{Y_i}^L - P_{Y_i}^T) \sim N(0, V_{Y_i}^L) \\ E_{Z_i}^L = (P_{Z_i}^L - P_{Z_i}^T) \sim N(0, V_{Z_i}^L) \end{cases} \quad (5-22)$$

$$i = 0, 1, \dots, m-1$$

Where $E_{X_i}^L$, $E_{Y_i}^L$ and $E_{Z_i}^L$ are components of localized measurement error, and $N(0, V)$ represents the normal distribution with zeros mean and variance V . The $V_{X_i}^L$, $V_{Y_i}^L$ and $V_{Z_i}^L$ are variance w infer in equation (5-18,19,20).

5.6.3 Surface Validation

After defining the localization error, we then need to define the reconstruction error. The error can be defined as following.

$$\begin{cases} E_X^R = (P_X^R - P_X^L) \sim N(0, V_X^R) \\ E_Y^R = (P_Y^R - P_Y^L) \sim N(0, V_Y^R) \\ E_Z^R = (P_Z^R - P_Z^L) \sim N(0, V_Z^R) \end{cases} \quad (5-23)$$

where the definition is similar to the localization error. Although $V_{X_i}^R$, $V_{Y_i}^R$ and $V_{Z_i}^R$ are unknown, It still can be estimated equation (5-24).

$$V_\gamma^R = \left(\sum_{i=0}^{m-1} (P_{\gamma_i}^R - P_{\gamma_i}^L)^2 \right) / (m-n) \quad (5-24)$$

where γ means X,Y or Z direction. m and n are number of measured points and control points respectively. Here that us denote the sum of resultant error as E_γ^S , and localization error is E_γ^L , and reconstruction error is E_γ^R , as shown in Fig. 5-24. By removing the effect caused by the uncertainties, we can obtain the validation error finally, which represents the difference between real workpiece and designed surface.

$$E_\gamma^S = E_\gamma^L + E_\gamma^R + E_{val} \quad (5-25)$$

The resultant error vector given in equation (5-26) is also can be rewritten as bellowing

$$E_\gamma^S = ((P_\gamma^L - P_\gamma^T) + (P_\gamma^R - P_\gamma^L)) + E_{val} \sim N(0, V_\gamma^L) + N(0, V_\gamma^R) + E_{val} \quad (5-26)$$

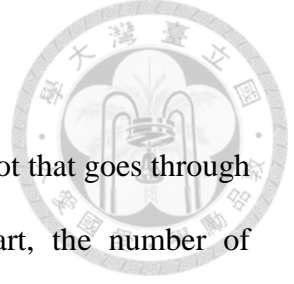
The sum of two normal distribution is also a normal distribution, where the mean of the resultant error is the mean of the contributing normal distributions, and the variance of the resultant normal distribution is the sum of the variances of the contributing normal distributions. Therefore the equation (5-2) can be changed to

$$E_{\gamma}^S = (P_{\gamma}^R - P_{\gamma}^T) + E_{\text{val}} \sim N(0, V_{\gamma}^L + V_{\gamma}^R) + E_{\text{val}} \quad (5-27)$$

And finally, the validation error can be obtained by the equation below.

$$E_{\text{val}} = E_{\gamma}^S - N(0, V_{\gamma}^L + V_{\gamma}^R) \quad (5-28)$$

And we can determine the variance of the E_{val} to express the extent of difference between real workpiece and designed surface.



Chapter 6 Experiment

In our experiment, we run the program on the iCeIRA CNC robot that goes through the procedure in section 1.4. In the geometric data extraction part, the number of parametric values are denoted as n_u and n_v along both parametric direction. Therefore $n_u \times n_v$ points are extracted by the different parametric (u, v) values. Here we denote the points on the designed surface as $p_{i,j}^W$, where W means the points are in WCS , and i, j are point number along the parametric direction, $(0 \leq i \leq n_u - 1, 0 \leq j \leq n_v - 1)$. And we also build the dataset $P_C^W = \{p_{0,0}^W, p_{1,0}^W, \dots, p_{n_u-1,0}^W, p_{0,1}^W, \dots, p_{i,j}^W, \dots, p_{n_u-1,n_v-1}^W\}$. After point extraction, the normal vector of each point is calculated and is denoted as $N_C = \{n_{0,0}^W, n_{1,0}^W, \dots, n_{i,j}^W, \dots, n_{n_u-1,n_v-1}^W\}$.

6.1 Localization Experiment

Initially, the localization is implemented to obtain the precise transformation matrix. We first use the reference-based method to obtain the transformation matrix, denote $T_{L,R}$. Because of using the geometric characteristics, we take the $T_{L,R}$ as a precise localization solution to verify the precision of the algorithm-based transformation matrix, $T_{L,A}$. The ICP algorithm is implemented with 50 points distributed onto the surface, and the result is shown below.

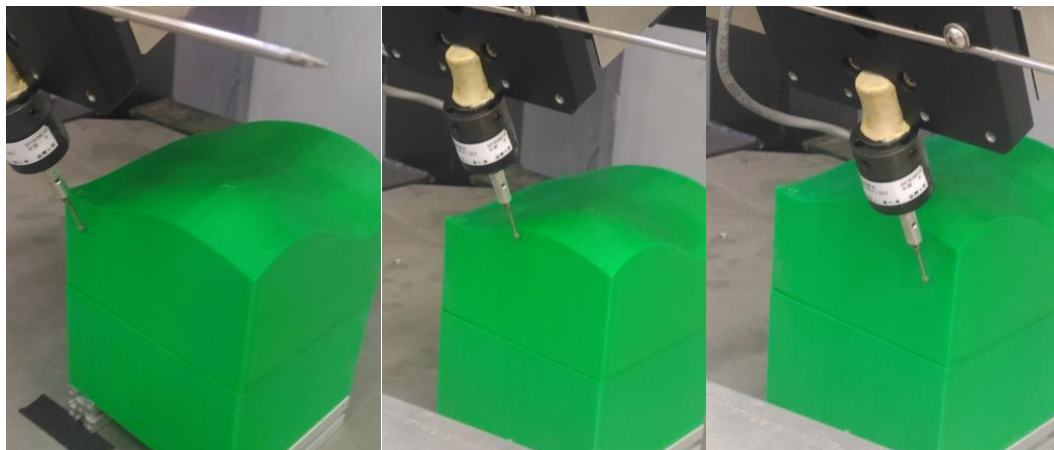
The measured points used for localization is denoted as P_L^W , and their corresponding points onto the designed surface (CAD model) is denoted as P_{CL}^W .



Table 6-1 Localization Result

		Algorithm-based result	Reference-based result	error
Rotation Angle(°)	α	-0.0562	0.041	-0.0972
	β	0.2839	0.033	0.0461
	γ	2.768	3.74	-0.672
Translation(mm)	Δx	2.486	2.172	0.314
	Δy	2.214	1.875	0.339
	Δz	1.015	0.841	0.174

As the Table 6-1 shows, we can know an algorithm-based solution is not precise enough if we cost more time to probe more points to be the dataset. Here we shows the real implementation of the reference-based picture in Fig. 6-1.



(a) first point

(b) second point

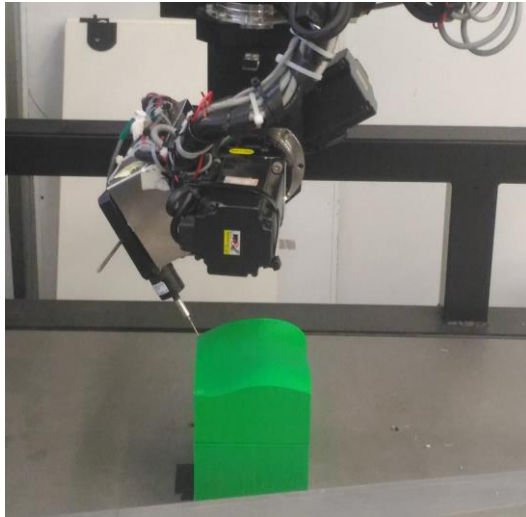
(c) third point

Fig. 6-1 Implementation of reference-based localization

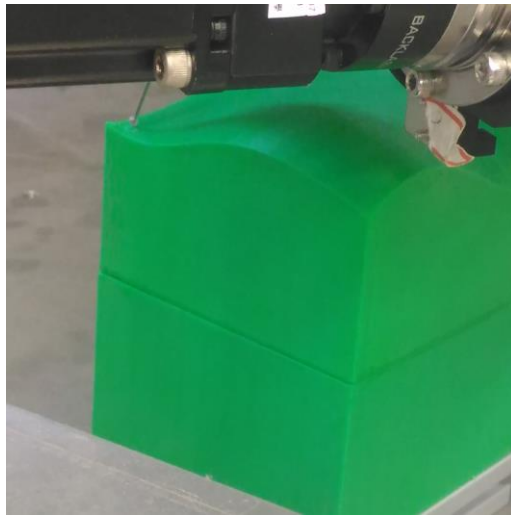
6.2 Freeform Surface Measurement Experiment



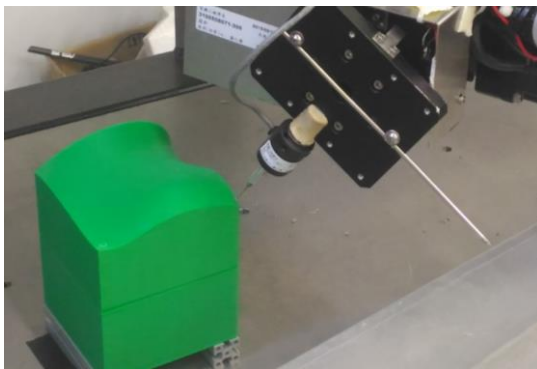
As the proposed measurement procedure, we start the freeform surface measurement after localization. We first extract four corner points as the measured points, and the implementation picture is shown in Fig. 6-2.



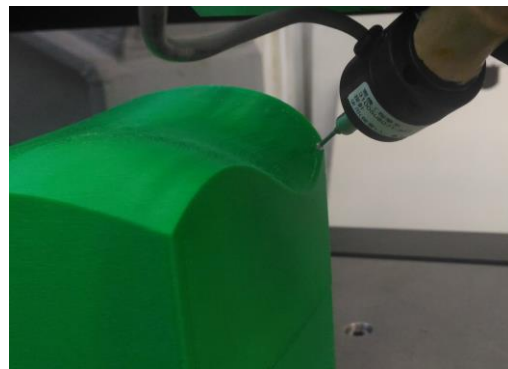
(a) Probe first point



(b) Probe second point



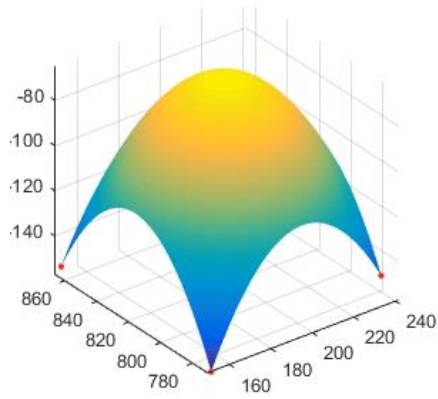
(c) Probe third point



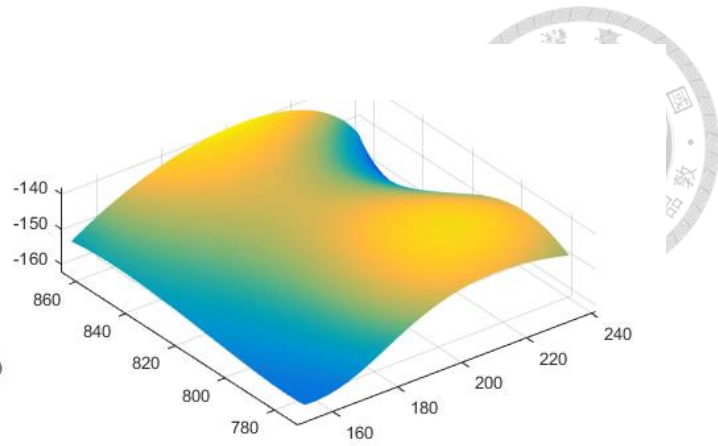
(d) Probe fourth point

Fig. 6-2 Implementation of freeform surface measurement

After obtaining the for position data feedback, the surface reconstruction can be done and the result is shown in Fig. 6-3. We compare it with original CAD surface to obtain the next measured point by the form-error model.



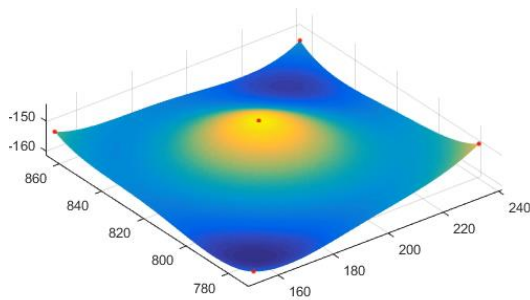
(a) The first reconstructed surface



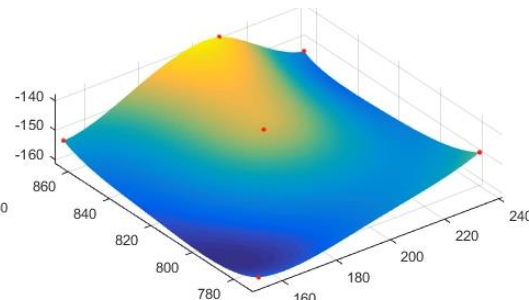
(b) surface from CAD model

Fig. 6-3 The first reconstructed surface compared with CAD model

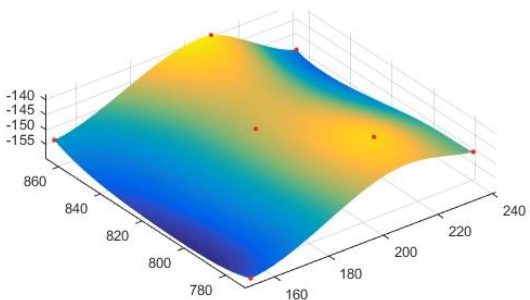
We can use the proposed measurement trajectory generator to plan the next path and execute the measurement, we show some reconstruction results with different measured points in the



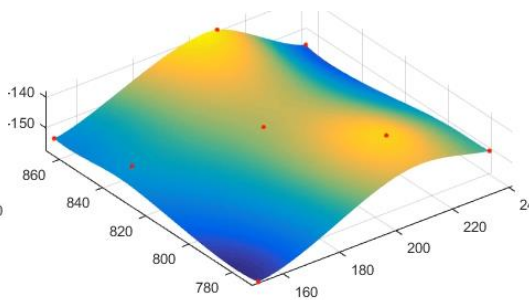
(a) Result with 5 measured points



(b) Result with 6 measured points

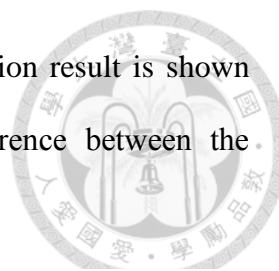


(c) Result with 7 measured points

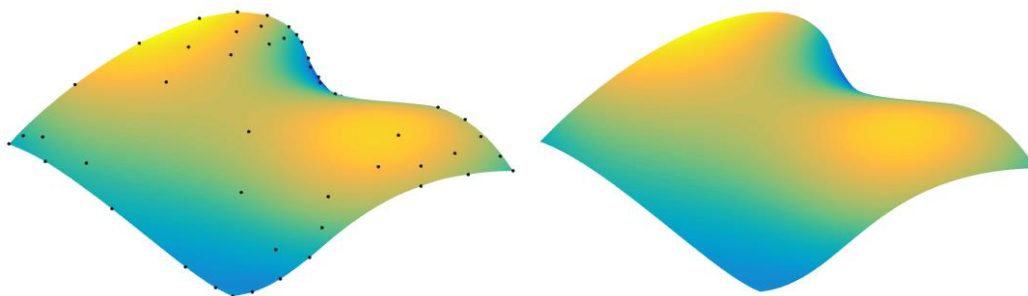


(d) Result with 8 measured points

Fig. 6-4 Surface reconstruction result with different number of measured points



We totally measure 50 points onto the surface, and the reconstruction result is shown Fig. 6-5. We can see that it's very hard to compare the difference between the reconstructed surface and CAD model.



(a) Reconstruction result with 50 measured points (b) surface from CAD model

Fig. 6-5 The final reconstructed surface compared with CAD model

To validate the efficiency of the measurement method, we calculate the sum of error values and obtain the error descent shown in Fig. 6-6.

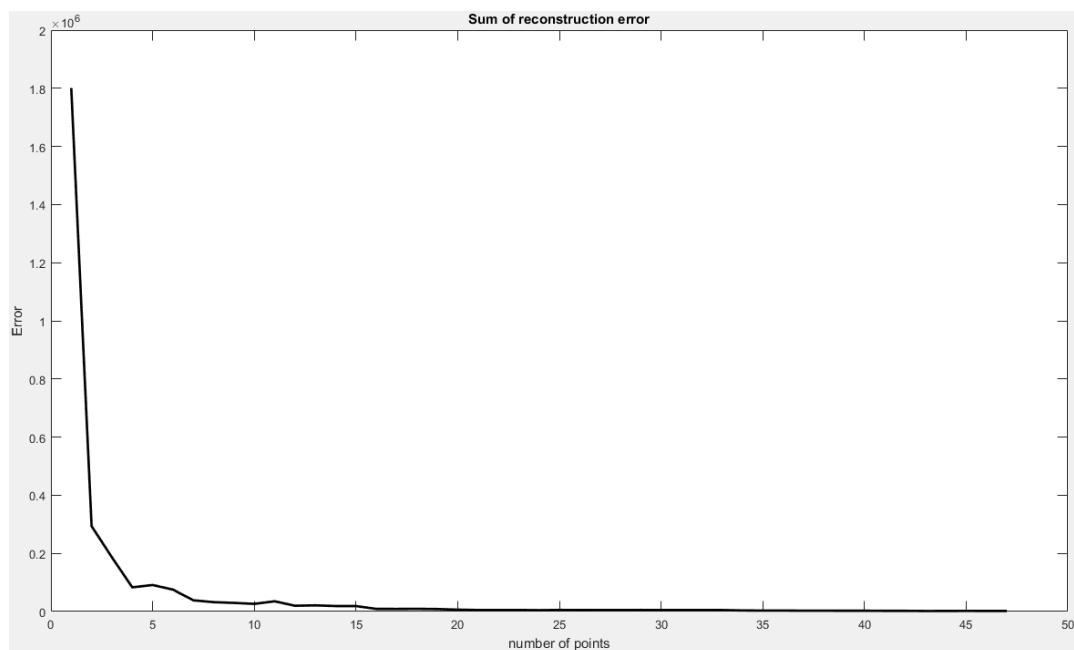


Fig. 6-6 Error descent of reconstruction error



6.3 Data Analysis and Surface Validation

Before starting the validation, we repeat the points data we have now.

- 40000 points from CAD model, denoted as P_C
- 50 points for localization P_L , and there corresponding points in CAD, P_{LC}
- 50 points measured from real surface P_M , and there corresponding points in CAD, P_{MC}

If we only use the $|P_{CM} - P_M|$ to determine the validation error, it seems to be unreasonable because the unilater data cannot be on behalf of total state of the real surface. Therefore we need to implement the roposed validation method in section 5.6. And out target is to find the E_{val} as the final result.

For our final reconstructed surface from 50 measured points P_M , The resultant error E_V^S is 0.1972, which means the error at a random point onto the surface is 0.1972mm in average. The error distribution is shown in Fig. 6-7.

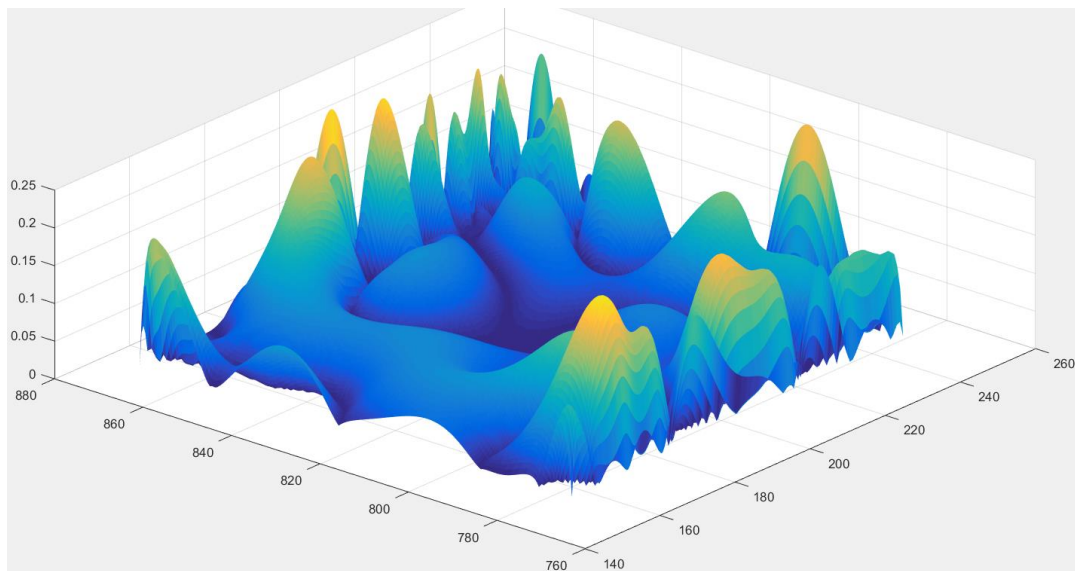
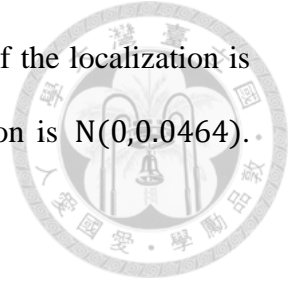


Fig. 6-7 The resultant error distribution

However, as we mention before, it's not a reasonable values because we still not



consider the uncertainties that interfere the result. The uncertainty of the localization is calculated as $N(0,0.0173)$ and the uncertainty of the reconstruction is $N(0,0.0464)$.

Therefore the final validation error can be obtain by

$$\begin{aligned} E_{\text{val}} &= E_{\gamma}^S - N(0, V_{\gamma}^L + V_{\gamma}^R) \\ &= 0.1972 - N(0, 0.0637) \end{aligned} \quad (6-1)$$

We can physically describe the validation result is that the maximum error at a points is 0.1972mm while the minimum error at a points is $0.1972 - 0.0637 = 0.1335\text{mm}$.

Finally, if we consider the accuracy of the TCP of the CMM, 0.02mm, we can furthermore indicate that the variance should be added on the value. Therefore the result should be

$$\begin{aligned} E_{\text{val}} &= E_{\gamma}^S - N(0, V_{\gamma}^L + V_{\gamma}^R) - N(0, 0.02) \\ &= 0.1972 - N(0, 0.0837) \end{aligned}$$

The minimum error of a point on the real surface is $0.1972 - 0.0837 = 0.1135\text{mm}$, and the maximum is not changed.

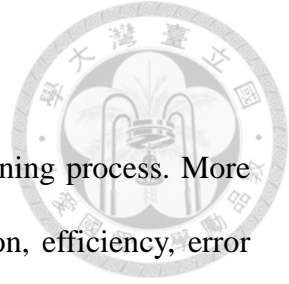
The 3D printer we use is called UPBOX!, which localization accuracy is 0.01mm~0.02mm, and its fabrication error is $\leq 0.1\text{mm}$. Considering the time and some other reasons that may cause the accuracy lower, we think it result is reasonable.



Fig. 6-8 3D printer used to manufacture the workpiece

Chapter 7 Conclusion

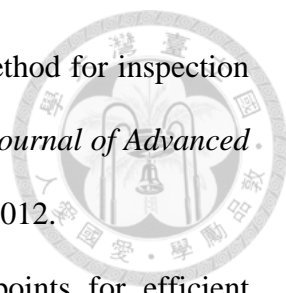
Coordinate measurement is a key issue in the industrial machining process. More and more researches try to improve their topics, including precision, efficiency, error compensation. In this thesis, we propose a freeform surface inspection and validation method to quantify the difference between reconstructed surface and CAD model. The full procedure is introduced in the different chapter, and the content is complete enough for the surface inspection and validation. However, we believe coordinate measurement is a deep technique that many topics are still can be studied. Radius compensation, measurement path planning, surface validation, and other many topics are still can be an independent research thesis. In this thesis, we provide a complete and efficient measurement process which is enough for validating a freeform surface. Although the process is not perfect enough, we believe this thesis create a strong basis for the measurement technique in iCeIRA, which is able to lead to more significant results in the future.

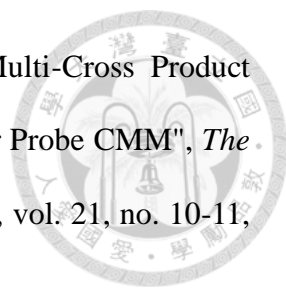


REFERENCE




- [1] I. Ainsworth, M. Ristic and D. Brujic, "CAD-Based Measurement Path Planning for Free-Form Shapes Using Contact Probes", *International Journal of Advanced Manufacturing Technology*, vol. 16, no. 1, pp. 23-31, 2000.
- [2] T. Woo, R. Liang, C. Hsieh and N. Lee, "Efficient sampling for surface measurements", *Journal of Manufacturing Systems*, vol. 14, no. 5, pp. 345-354, 1995.
- [3] S. Obeidat and S. Raman, "An intelligent sampling method for inspecting free-form surfaces", *The International Journal of Advanced Manufacturing Technology*, vol. 40, no. 11-12, pp. 1125-1136, 2008.
- [4] X. He, Y. Xue, M. Ni, C. Hua and C. Li, "Adaptive measuring algorithm for CMM based on 5-th Bezier curve", *IEEE ICCA 2010*, 2010.
- [5] A. Zhou, J. Guo and W. Shao, "Automated inspection planning of freeform surfaces for manufacturing applications", *2011 IEEE International Conference on Mechatronics and Automation*, 2011.
- [6] L. Mu, Z. Yin and Y. Xiong, "Sampling Method for Similar Section Surface Inspection", *2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, 2011.
- [7] X. Lin, S. Jiang, X. Liu and K. Yang, "The CMM Measurement Path Planning for Blade Surface Based on the Contour Measurement", *2011 Second International Conference on Digital Manufacturing & Automation*, 2011.
- [8] Bai Yuewei, Wei Shuangyu, Liu Kai and Wang Xiaogang, "A strategy to automatically planning measuring path with CMM offline", *2010 International Conference on Mechanic Automation and Control Engineering*, 2010.

- 
- [9] M. Yu, Y. Zhang, Y. Li and D. Zhang, "Adaptive sampling method for inspection planning on CMM for free-form surfaces", *The International Journal of Advanced Manufacturing Technology*, vol. 67, no. 9-12, pp. 1967-1975, 2012.
- [10] Y. Li and Z. Liu, "Method for determining the probing points for efficient measurement and reconstruction of freeform surfaces", *Measurement Science and Technology*, vol. 14, no. 8, pp. 1280-1288, 2003.
- [11] Y. Chen, Z. Ma and H. Xu, "Key technologies of 3D surface inspection for complex workpiece using OMP60 probe", *2009 IEEE International Conference on Automation and Logistics*, 2009.
- [12] E. Wang, Y. Gao, C. Su, P. Yu, M. Jiang and Y. Liu, "New algorithm of measuring the cone section with the coordinate machine measurement", *2012 International Conference on Optoelectronics and Microelectronics*, 2012.
- [13] H. Aoyama, M. Kawai, T. Kishinami and N. Taniguchi, "A New Method for Detecting the Contact Point between a Touch Probe and a Surface", *CIRP Annals - Manufacturing Technology*, vol. 38, no. 1, pp. 517-520, 1989.
- [14] C. Menq and F. Chen, "Curve and surface approximation from CMM measurement data", *Computers & Industrial Engineering*, vol. 30, no. 2, pp. 211-225, 1996.
- [15] R. Johnson, Qingping Yang and C. Butler, "Dynamic error characteristics of touch trigger probes fitted to coordinate measuring machines", *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 5, pp. 1168-1172, 1998.
- [16] M. Ristic, I. Ainsworth and D. Brujic, "Contact probe radius compensation using computer aided design models", *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 215, no. 6, pp. 819-834, 2001.

- 
- [17] Y. Lin and W. Sun, "Probe Radius Compensated by the Multi-Cross Product Method in Freeform Surface Measurement with Touch Trigger Probe CMM", *The International Journal of Advanced Manufacturing Technology*, vol. 21, no. 10-11, pp. 902-909, 2003.
- [18] F. Zhang, J. Lu, S. Tang, Y. Yan and H. Sun, "Analytic model of machining errors for thin walled parts based on the deformation of the workpiece-fixture system", *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, 2009.
- [19] A. Nafi, J. Mayer and A. Wozniak, "Novel CMM-based implementation of the multi-step method for the separation of machine and probe errors", *Precision Engineering*, vol. 35, no. 2, pp. 318-328, 2011.
- [20] L. Laaouina, A. Nafi and A. Mouchtachi, "Application of CMM separation method for identifying absolute values of probe errors and machine errors", *2016 International Conference on Engineering & MIS (ICEMIS)*, 2016.
- [21] Z. Mei, "CMM Measurement Error Model Based on High-order Lagrange Interpolation", *Information Technology Journal*, vol. 12, no. 15, pp. 3457-3461, 2013.
- [22] J. Mayer, Y. Mir, F. Trochu, A. Vafaeseefat and M. Balazinski, "Touch probe radius compensation for coordinate measurement using kriging interpolation", *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 211, no. 1, pp. 11-18, 1997.
- [23] Zexiang Li, Jianbo Gou and Yunxian Chu, "Geometric algorithms for workpiece localization", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 864-878, 1998.
- [24] Z. Xiong and Z. Li, "Error compensation of workpiece localization", *Proceedings*

2001 ICRA. *IEEE International Conference on Robotics and Automation* (Cat. No.01CH37164).

- 
- [25] Z. Xiong, M. Wang and Z. Li, "A Near-Optimal Probing Strategy for Workpiece Localization", *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 668-676, 2004.
- [26] K. Morishige, K. Kase and Y. Takeuchi, "Collision-free tool path generation using 2-dimensional C-space for 5-axis control machining", *The International Journal of Advanced Manufacturing Technology*, vol. 13, no. 6, pp. 393-400, 1997.
- [27] Jin Bao, Wang Shuguo and Yili Fu, "Sensor-based motion planning for robot manipulators in unknown environments", *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [28] A. Ademovic and B. Lacevic, "Path planning for robotic manipulators using expanded bubbles of free C-space", *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [29] M. Ngom and F. Nagata, "Detection of workpieces on a CNC machining table for measurement and automatic tool positioning", *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, 2014.
- [30] V. Mehrad, D. Xue and P. Gu, "Inspection of freeform surfaces considering uncertainties in measurement, localization and surface reconstruction", *Measurement Science and Technology*, vol. 24, no. 8, p. 085008, 2013.
- [31] T. Möller, "A Fast Triangle-Triangle Intersection Test", *Journal of Graphics Tools*, vol. 2, no. 2, pp. 25-30, 1997.
- [32] N. Gelfand, L. Ikemoto, S. Rusinkiewicz and M. Levoy, "Geometrically stable sampling for the ICP algorithm", *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*

VITA



姓名：李晟

性別：男

生日：10.10.1993

籍貫：台中市

學歷：

1. 民國 106 年 國立台灣大學電機工程學研究所畢業
2. 民國 104 年 國立台灣科技大學電機工程學系畢業
3. 民國 100 年 國立台中高級工業職業學校控制科畢業

發表著作：

1. Ren C. Luo, Alexander Chang, Cheng Li, "A Novel Peg-in-hole Approach based on Geometrical Analysis for Inclined Uncertainty," *IEEE International Conference on Advanced Intelligent Mechatronics, AIM 2017*.
2. Ren C. Luo, Cheng Li, "Novel Strategy on Efficient Determination of Inspection Path and Sampling Points for Free-Form Surface Measurement," *International Symposium on Robotics, ISR 2017*.

榮譽事績：

2016/09/01 第九屆上銀智慧機器手臂實作競賽 開發組 亞軍

2016/09/01 第九屆上銀智慧機器手臂實作競賽 疊疊樂 冠軍