A COMPONENT BASED 3D COLLISION AVOIDANCE SYSTEM

BY

PENG WU
B. Eng.

A Component Based
3D Collision Avoidance System

A Component Based 3D Collision Avoidance System

By

PENG WU, B. Eng. (Electrical Engineering)

Wuhan University of Hydraulic and Electric Engineering

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirement

for the Degree

Master of Applied Science

McMaster University

Master of Applied Science (2003)	McMaster University
(Electrical and Computer Engineering)	Hamilton, Ontario

TITLE:	**A Component Based 3D Collision Avoidance System**
AUTHOR:	Peng Wu B.Eng. (Electrical Engineering) Wuhan University of Hydraulic and Electric Engineering, Wuhan, Hubei, China
SUPERVISOR:	Dr. Szabados, Professor Department of Electrical and Computer Engineering McMaster University
CO-SUPERVISOR:	Dr. Allan D. Spence, Associate Professor Department of Mechanical Engineering McMaster University
NUMBER OF PAGES:	ix, 63

Abstract

The Coordinate Measuring Machine (CMM) is a key improving the reliability and quality of production. During operation, however, collision between the CMM, probe system, and parts may result in costly damage. Avoiding potential collisions is always an important issue for CMM inspection.

This thesis reports design and implementation of an online collision detection system for CMMs. Compared to previous offline collision detection implementations, the new online collision detection system can react to potential collision situations in real time.

The collision avoidance system architecture and algorithms are described in this thesis. Each possible CMM translation and probe rotation case is considered. Every component in this system is connected by certain communication protocol and distributed on different computers according to the requirements of applications. This design brings more flexibility to this system.

The system components for inspection path planning, collision detection, and CMM motion control are distributed to separate computers, with a TCP/IP based protocol used for communication. This modular, component based design provides more flexibility.

Acknowledgements

I would like to express my deep appreciation to my supervisors, Dr. Szabados and Dr. Spence for their guidance and encouragement throughout the work of my research during the past two years. I would especially like to thank them for developing my problem solving skills and generous financial support.

To Cheryl Gies, Steve Spencer, Stan Zolinski, Fran Hustak, thank you for your help in administrative and computer technical aspects of my work. I am also grateful to my fellow graduate students David Chang and Harley Chan who provided me valuable suggestions and assistance in my research. I would also like to thank other graduate students Fang Liu, Hongjuan Wang, Hongyan Liu, Antanio Naddour, John Harris, John (Philip) Mitchell for their friendship. I really enjoy the unforgettable time spent with you guys.

Finally I would like to thank my family for your love and support. Without your inspiration, I cannot realize my dream originated twelve years ago.

Contents

ABSTRACT .... **III**

ACKNOWLEDGEMENTS .... **IV**

LIST OF FIGURES .... **VII**

LIST OF TABLES .... **IX**

CHAPTER 1: INTROUDUCTION **1**

CHAPTER 2: LITERATURE REVIEW **4**

2.1 Collision Detection Problems in Manufacturing 4

2.2 Interference Detection Methods 6

2.2.1 Space - time Volume Intersection 6

2.2.2 Swept Volume Interference 7

2.2.3 Multiple Interference Detection 7

2.2.4 Trajectory Parameterization 8

2.3 Communications 8

2.3.1 Layers of TCP/IP 8

2.3.2 Internet Protocol 10

2.3.3 Transmission Control Protocol 11

2.3.4 Windows Sockets 13

2.4 CMM Components and Operations 14

2.4.1 CMM Components 16

2.4.2 CMM operation 18

CHAPTER 3: ONLINE CMM COLLISION DETECTION SYSTEM 20

3.1 Architecture 20

3.1.1 CheckMate 22

3.1.2 GEOMeasure 23

3.1.3 OTC Controller 24

3.1.4 Collision Detection System 24

3.2 Implementation 28

3.2.1 CMM Geometric Representation 28

3.2.2 Swept Volume Generation 30

3.2.3 Communication Protocols Connections 41

3.2.4 ACIS Network Extension 43

CHAPTER 4: EXPERIMENTS AND PERFORMANCE ANALYSIS 46

4.1 Experiment Configuration 46

4.2 Experiment Procedure 48

4.3 Performance analysis 53

CHAPTER 5: SUMMARY AND FUTURE WORK 56

5.1 Achievements 56

5.2 Limitations 57

5.3 Future Work 58

REFERENCES 59

List of Figures

Figure 2.1: A dual-arm workcell 5

Figure 2.2: Sweeping of a geometric entity 7

Figure 2.3: Model of TCP/IP layers compared with OSI model 9

Figure 2.4: The relationship between an IP datagram and a TCP segment 11

Figure 2.5: TCP three - way handshake 12

Figure 2.6: Windows sockets model and OSI model 13

Figure 2.7: CMM Configurations 15

Figure 2.8: DEA IOTA - P moving bridge CMM 16

Figure 2.9: Renishaw probe head system [16] 17

Figure 2.10: Renishaw RGH22 optical encoder displacement transducer 18

Figure 3.1: CAD-CMM system without online collision detection systems 20

Figure 3.2: Working flow of online collision detection system 22

Figure 3.3: CAD-CMM system with online collision detection system 24

Figure 3.4: The Procedure of collision check 27

Figure 3.5: A moving bridge CMM model 29

Figure 3.6: Plane surface creation 30

Figure 3.7: CMM move case 1 31

Figure 3.8: CMM move case 2 32

Figure 3.9: CMM move case 3 33

Figure 3.10: CMM move case 4 34

Figure 3.11: Probe in Cartesian coordinate system 35

Figure 3.12: Probe rotation case 1..36

Figure 3.13: Probe rotation case 2..37

Figure 3.14: Probe rotation case 3..38

Figure 3.15: Probe rotation case 4..39

Figure 3.16: The possible space by considering two extreme situations.....................40

Figure 3.17: Probe rotation case 5..41

Figure 3.18: Conversion process between serial and TCP/IP protocols.....................42

Figure 3.19: ACIS TCP/IP Startup..44

Figure 3.20: Procedure to process collision check request..................................45

Figure 4.1: Hardware configuration of experiments..47

Figure 4.2: Tested parts..47

Figure 4.3: Combination of CMM and tested part model.....................................49

Figure 4.4: Online measurements of yoke..50-51

Figure 4.5: Online measurements of ball bar..52

List of Tables

Table 4.1 Time of measurements of yoke..54

Table 4.2 Time of measurements of ball bar...55

Chapter 1

Introduction

The Coordinate Measuring Machine (CMM) is an important component in manufacturing systems because it can provide high precision measurements for parts to ensure the quality of products. In recent decades, the CMM has become more and more widely integrated into Computer Aided Design (CAD) / Computer Aided Manufacturing (CAM) to play a key role in dimensional inspection. A typical computer based CMM consists of mechanical components, a probe head with two angle turning capability of the probe, a controller and a computer to send commands to the controller and analyze the data transmitted from the CMM.

Since the CMM has a five axes measuring capability, it may collide with inspected parts, fixtures, or even with itself. Collision can easily damage the probe, part, fixture, or one of the CMM components. Human supervision can then modify the inspection path manually to avoid collisions, but visual check is labour intensive and unreliable. Therefore, off line collision free path planning was proposed in recent years. In such a case, the whole inspection procedure is simulated by software to verify the inspecting path. If the proposed path causes collision, the system may change the inspection path to avoid collision. This method can avoid most collisions but it has its own disadvantages. First, it is not real time and needs computer simulation to verify its validity in off line mode. Second, it may spend a long time to get collision free inspection path in some complicated cases. Therefore a real time collision detection system for CMMs is proposed. It can not only release the workload of a human operator but also realize part inspection in an on line mode and completely avoid collision during the actual inspection procedure.

The remainder of the thesis is organized as follows.

Chapter 2 is the literature review on collision detection problems in manufacturing, approaches to collision detection, communication, CMM components and operation. Four approaches to collision detection: spatio-temporal intersection, swept volume interference, multiple interference detection, trajectory parameterization are included. For the communications part, the TCP/IP protocol and Winsock are discussed. CMM components and how CMM measures parts are also presented in this chapter.

Chapter 3 describes the architecture and implementation of an online collision detection system embedded in Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM) Systems. The components in online collision detection modeller such as CAD bridge, controller bridge, collision detection bridge, collision check system are analyzed. In implementation of online collision detection systems, CMM geometric representation, swept volume generation, bridging communication protocols, and ACIS network extension are presented. The details of modeling CMM by the ACIS 3D solid modeler are discussed in this chapter. Swept volume generation are classified into two categories, CMM move and probe rotation. Each case for the two categories is presented in this chapter. On the issue of bridging different communication protocols, this chapter illustrates how to implement the connection between serial communication protocol and

TCP/IP communication protocol. Since the current version of ACIS does not have a networking capability, a networking function must be added. This chapter includes how to implement ACIS network extension.

Chapter 4 is the analysis of the performance of the online collision detection system proposed. Experiments details are also described in this chapter.

Chapter 5 is the summary of the thesis and some discussions of future work which can expand this research to wider applications.

Chapter 2

Literature Review

2.1 Collision Detection Problems in Manufacturing

Robots are widely applied in modern society. Examples include robotic arms on assembly lines, mobile robots operating in extreme environments such as space, deep ocean, and humanoid robots imitating the behavior of human being. It is possible that collision between the robot and parts, robot and robot, robot and operators, even the self collision of robot, may happen when robot moves from one position to another position. Collision may damage costly components of robots and parts and may cause substantial economic losses. If robots collide with operators, the safety of operators is seriously threatened. Therefore, the research of collision in robotics area is so important.

Early research concentrated on the collision between a single robot and parts. With the growing competition in the market, multiple robots appear more frequently in manufacturing area. Multiple robots working together have a number of advantages: 1) improve the working capability. 2) decrease total cycle time because robots can work on the task in parallel. 3) increase reliability because robot can interchange tasks. 4) decrease space and material handling time. But multiple robots require more complicated control strategies for coordinated motion to avoid potential collision. In a manufacturing environment, even though paths of individual arms can be implemented very accurately for most assembly robots, the synchronization of the different robotic arms may not be guaranteed without the coordination of individual controllers. As a result, online collision free path planning is rather more important than offline path planning. For example, Figure 2.1 [1] shows a dual arm assembly workcell in which both arms have to pass through a gluing station. The offline path planner has determined two collision free paths for the arms. However, one of the arms may be delayed due to actuator overloading, jamming or other abnormal situations during the gluing operation resulting in a collision.

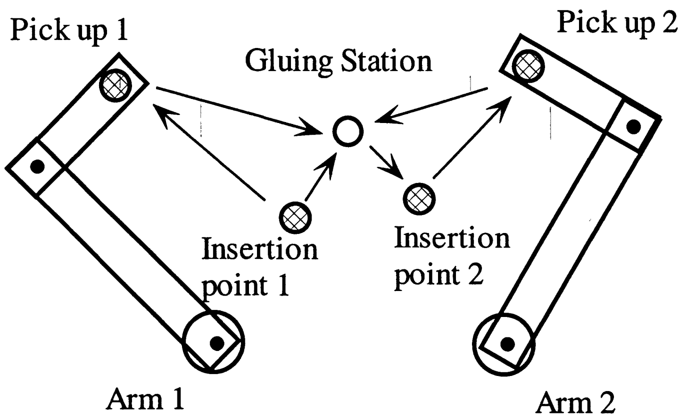


Figure 2.1: A dual – arm workcell

The Coordinate Measuring Machine (CMM) is a special case of robot. Typically it can move along three axes of Cartesian coordinate system and has probes with two degrees of freedom. The CMM probe needs to move closely around the inspected part and touch the part to get the measured data, so it is possible that the probe interferes with the inspected part and fixtures. Monitoring by a human operator can then modify the inspection path manually to avoid collisions but this visual check is time consuming and unreliable. Therefore, CMM collision free inspection path planning has been proposed in recent years. Several solutions [2][3][4][5][6] are presented but all these solutions are based on offline modes. In such a case, the whole inspection procedure is simulated by CAD software prior to actual execution to verify the inspection path. If the planned path causes a collision, the system may change the inspected path. This method can avoid most collisions but it has its own disadvantages. First, It is not real time and needs computer simulation to verify its validity. Second, it may spend a long time to get collision free inspecting path in some complicated cases. Therefore a real time collision detection system is proposed. It not only releases the workload of a human operator but also completely avoids collisions during the inspection procedure.

2.2 Interference Detection Methods

The problem of interference detection can be stated as follows: Given a set of objects and their appropriate motions, decide whether the objects will collide each other over a given time span. Solutions to this problem mainly come from four methods: space-time intersection, swept volume interference, multiple interference detection and trajectory parameterization[7].

2.2.1 Space - time Volume Intersection

The approach of Space – time volume intersection to collision detection is based on extrusion operations [8] which create four dimensional extruded volume having three dimensions in space and one dimension in time. The extruded volume of an object is a set of points representing the spatial occupancy of the object corresponding to time. Two objects collide with each other if and only if their extrusions intersect.

2.2.2 Swept Volume Interference

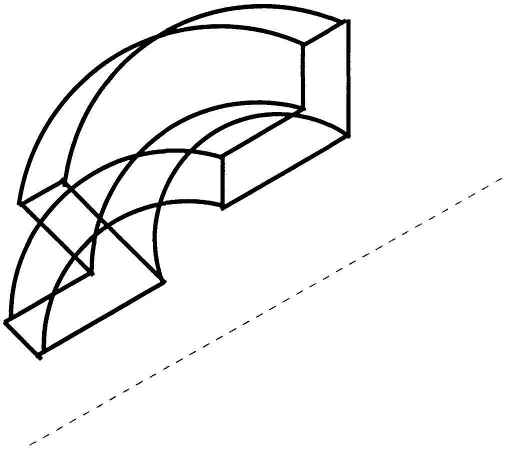


Figure 2.2: Sweeping of a geometric entity

An object moving along an arbitrary path during a given period creates a swept volume (Figure 2.2)[9]. This approach can decide that no collision will occur if the swept volumes generated by the objects have no intersection. However, the intersection of swept volumes does not imply that the collision happens as the objects may pass through the intersection zone at different times. In order to decide whether the collision takes place, one of the objects is fixed and the relative motion of other objects generates swept volumes. If the swept volumes intersect with the fixed objects, collision occurs. This thesis adopted this method to detect potential collisions between CMM and parts or CMM self collision.

2.2.3 Multiple Interference Detection

Multiple interference detection samples object trajectories in a certain interval and repeatedly applies a static interference test. The sampling strategy affects whether this method is effective or not. A too coarse sampling may not detect a collision, while a too frequent sampling will be computationally expensive.

2.2.4 Trajectory Parameterization

Trajectory parameterization describes the object trajectories as functions of time and the collision moment is determined by solving the polynomials. If the degrees of the polynomials are more than five, solutions can not be found analytically, and numerical methods will need to be applied. The computation of the collision instant is very expensive.

2.3 Communications

With the development of Internet, more and more industrial applications use Transmission Control Protocol/Internet Protocol (TCP/IP) to connect various devices in factories.

2.3.1 Layers of TCP/IP

Networks use layering for their transmission protocols and each layer provides certain unique functions that not provided by any of the other layers. TCP/IP also adopts this network architecture. TCP/IP does not follow the Open System Interconnection (OSI) model of network architecture by International Organization for Standardization (ISO). OSI model has seven layers: physical layer, data link layer, network layer, transport layer, session layer, presentation layer. However, TCP/IP has five layers shown in Figure 2.3. Neither presentation layer nor session layer is included in this mode.

OSI network layers	TCP/IP layers
Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Transmission Control Protocol
Network Layer	Internet Protocol
Data Link Layer	Network Interface Protocols
Physical Layer	Physical Networks

Figure 2.3: Model of TCP/IP layers compared with OSI model

The layer of physical networks provides the mechanical and electrical characteristics to establish, maintain and release physical connections. The layer of network interface protocols provides data link service. TCP/IP doesn't define any specific protocol in this layer, but instead interfaces with any available protocols, such as Ethernet, Token Ring, X.25 etc. The layer of Internet protocol provides the service of exchange of information. The layer of transmission control protocol provides services so that one application program on a particular host can communicate with another application program on a remote host. Two protocols are in this layer: User Datagram Protocol (UDP) which cannot guarantee data to reaching receiver, and Transmission Control Protocol (TCP) which can guarantee data reaching receiver. The application layer is responsible for formatting data and performing services which depend on different application needs.

Each layer on the sending host adds information to the message and each layer on the receiving host removes information from the message. For example, the application layer generates data and passes the data to the transport layer. The TCP layer on the sending host will then add information to the front of the data and pass it to the IP layer. For the receiving host, the information added by each layer in the sending host will be analyzed and removed by the corresponding layer before passing the data to the next higher layer.

2.3.2 Internet Protocol

The basic functions provided by Internet Protocol (IP) are addressing and fragmentation. The delivery of data from one computer to another is unreliable because it cannot guarantee the delivery and provides no sequencing to ensure that the data units are received in order. No acknowledgement is required for any computers. The data are packaged in a datagram which is the fundamental unit of Internet communication.

The current standard of Internet protocol (IP) is version 4. This version uses 32 -bit addresses for both the source and destination computers. However, the 32-bit address length of IP is inadequate for the rapid growth of Internet. A new version of protocol, known as IP version 6, has been proposed to replace the current IP version 4. IP v6 uses 128-bit addresses which can increase the address space enormously to meet the demand of addresses.

2.3.3 Transmission Control Protocol

Transmission Control Protocol (TCP) is the principal transport protocol for all Internet applications, including HTTP, FTP etc. A TCP segment fits inside an IP datagram, as shown in Figure 2.4.

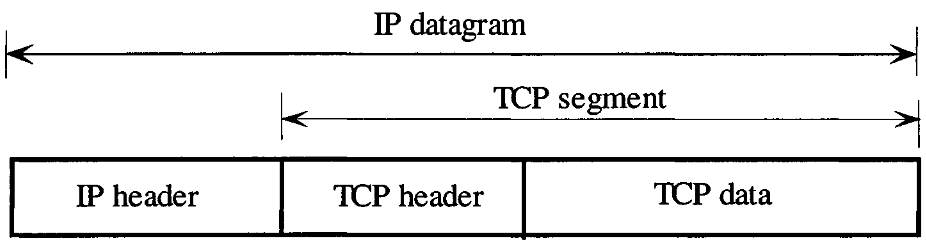


Figure 2.4: The relationship between an IP datagram and a TCP segment.

Compared to unreliable delivery of data, TCP provides reliable end-to-end byte stream over an unreliable Internet link [9]. On the sending host, data streams from higher layer are divided into segments and then transmitted to IP. TCP on the receiving host receives bytes from IP and reassemble them in the exact sequence in which they are transmitted.

A TCP connection between two computers is a full duplex, point to point connection which is established with a three-way handshake [12] as shown in Figure 2.5. The following steps can setup a TCP connection. First, the server must prepare to accept connections from clients. Second, the client sends a segment with the *SYN* flag set. Third, the server sends the client a segment with both SYN and ACK flag set. Fourth, the client sends a segment with the ACK flag set.

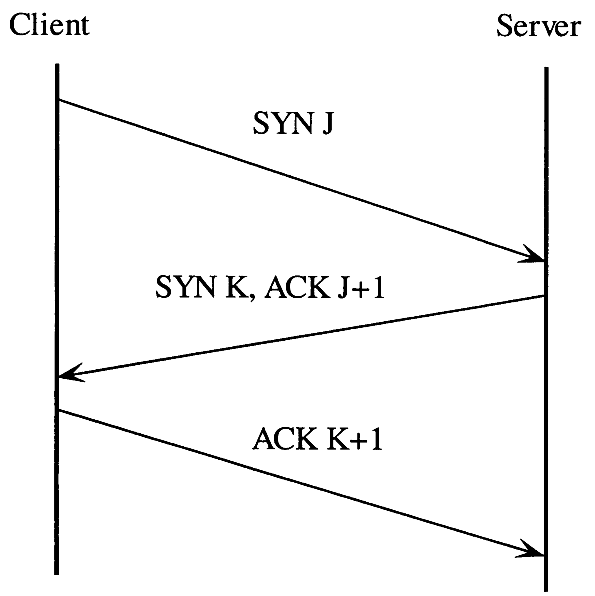


Figure 2.5: TCP three - way handshake

After the connection is established, both the server and the client can send a stream of bytes to each other. TCP uses the sequence number fields together with *ACK* flags to control this flow of bytes. The sending side doesn't wait for each segment to be acknowledged but instead sends a number of segments together and then waits for the first acknowledgment. If the receiving side has data to send back to the sending side, it can piggyback its acknowledgment and outbound data together in the same segments. The sending side's sequence numbers are not segment indexes but rather indexes into the byte stream. The receiving side sends back the sequence numbers to the sending side, thereby ensuring that all bytes are received and assembled in sequence. The sending program resends unacknowledged segments.

Each side can close its end of TCP connection by sending a segment with the *FIN* flag set, which must be acknowledged by the side on the other end. One side can no longer receive bytes on a connection that has been closed by the program on the other end.

2.3.4 Windows Sockets

A socket is a communications endpoint, and is supported on every operating system. Similar to a wall socket for electric power, a socket is a standard network interface for applications. The Microsoft Windows operating system incorporates Windows sockets and TCP/IP to implement network communication. As shown in Figure 2.6 [13], a Windows sockets application lies on the top of TCP/IP and the Windows sockets Application Programming Interface (API) acts as an interface to connect the application and the TCP/IP protocol stack.

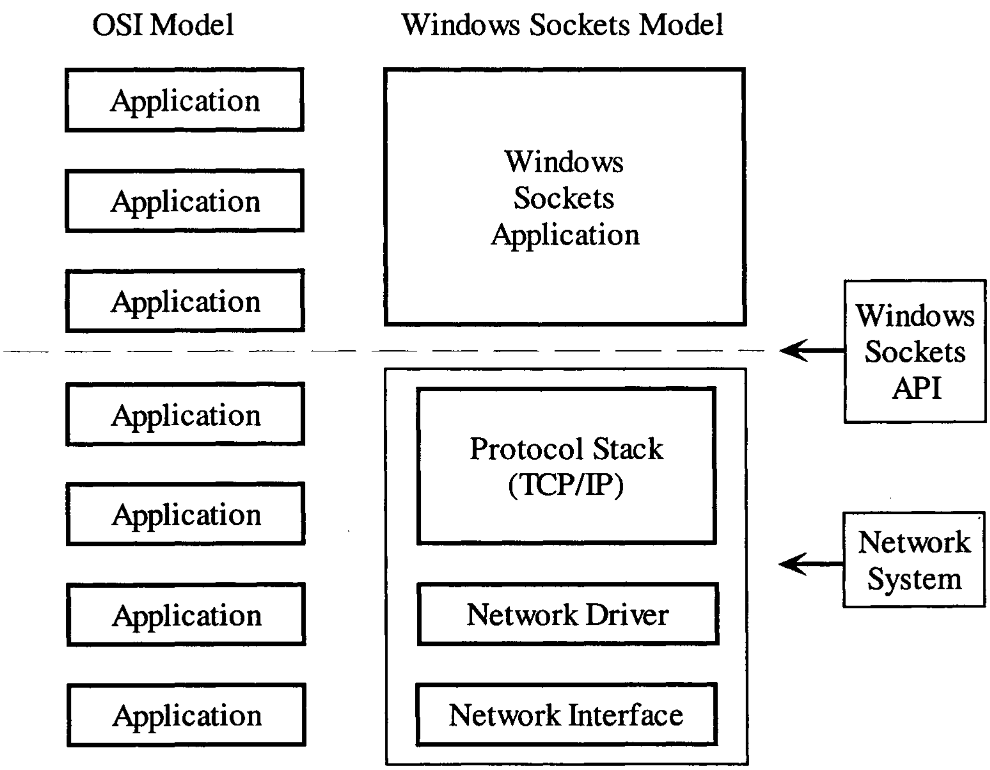


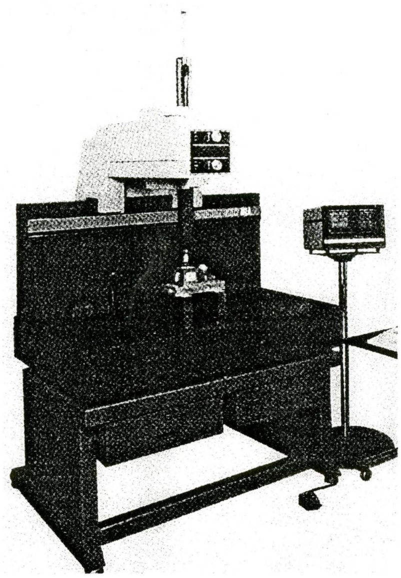
Figure 2.6: Windows sockets model and OSI model

Windows sockets have three operation modes: blocking mode, non-blocking mode and asynchronous mode. A blocking call does not return until the operation is successful. A non-blocking call returns immediately whether the call is successful or not. If the operation fails, the application will wait for a certain interval and request again. A call in asynchronous mode returns immediately and the opposite side will call back. This mode is preferred because it doesn't need to wait like blocking mode or call again like non-blocking mode. However, the implementation of asynchronous mode is the most complicated.

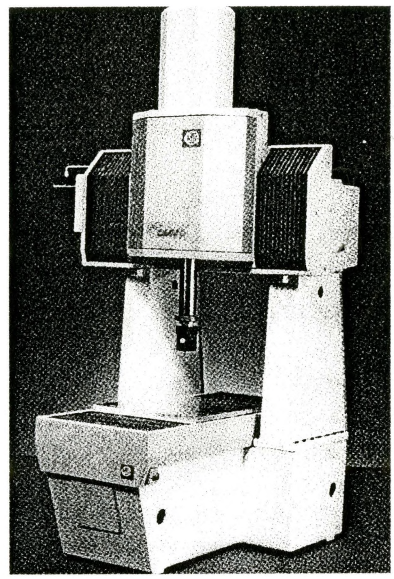
2.4 CMM Components and Operations

A milestone occurred in 1956, when Ferranti Ltd. in Scotland manufactured the first CMM with moving X and Y axes, a fixed Z axis, and a solid probe. According to the ANSI/ASME B89.4 standard [14], CMMs can be classified into eleven categories: fixed table cantilever, moving bridge, fixed bridge, column, moving ram, horizontal arm, duplex mode machine, moving table horizontal arm, gantry, L-shaped bridge, fixed table horizontal arm, moving table cantilever arm. The eleven categories can be condensed into four primary configurations [15]: cantilever (Figure 2.7 (a)) , bridge (Figure 2.7 (b)), horizontal arm (Figure 2.7 (c))and gantry (Figure 2.7 (d)).

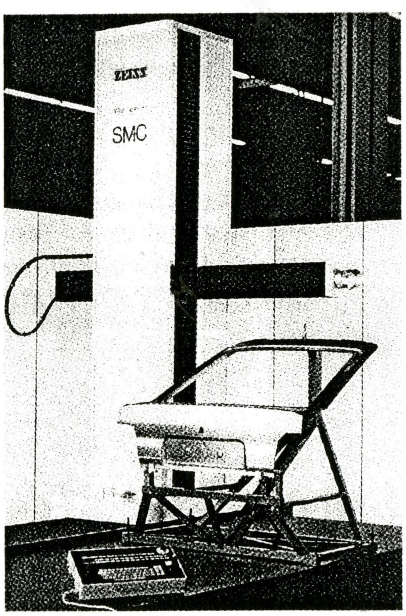
Cantilever CMMs provide good accessibility but are only suitable for general measurements. Bridge CMMs are suitable for measuring parts with medium size and height. Horizontal arm CMMs are suitable for measuring larger parts with high speed. Gantry CMMs are designed to measure very large parts such as aerospace structures, large vehicles. The moving bridge CMM is the most widely used CMM, and is considered in this thesis. (Figure 2.8)



(a)



(b)



(c)

(d)

Figure 2.7: CMM Configurations

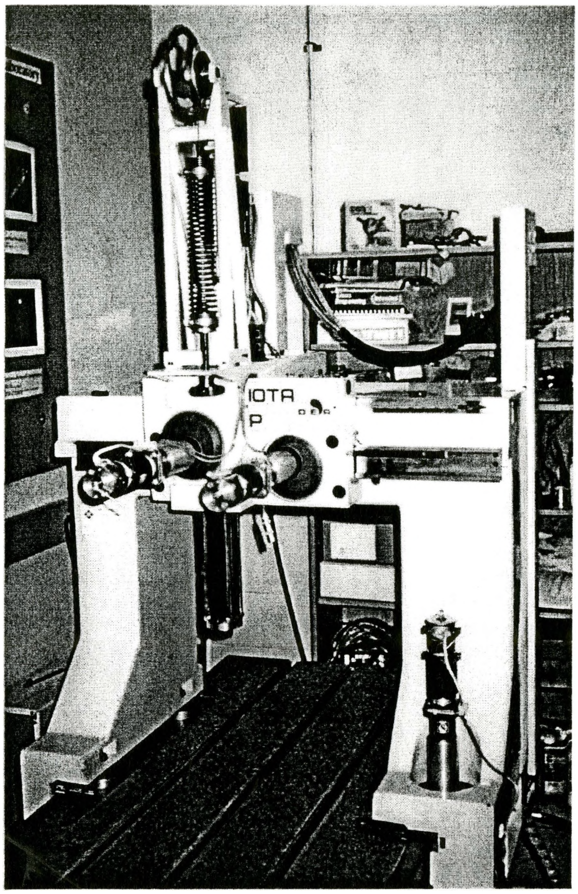


Figure 2.8: DEA IOTA – P moving bridge CMM

2.4.1 CMM Components

The main components of the research DEA IOTA-P moving bridge CMM (Figure 2.8) include the mechanical structure, probe head system, displacement transducers, and driving motors.

The mechanical structure consists of a vertical column moving along the Z axis, a horizontal column moving along a horizontal track parallel with the Y axis, a bridge moving along the X axis and a granite table supporting the whole mass of the bridge, vertical and horizontal columns.

The probe head system in this thesis is Renishaw PH10M system (Figure 2.9) [16] which consists of a PH10M motorized probe head, a PHC10 controller and a PHD10 probe head drive unit. The Renishaw probe used in this thesis is a common touch trigger probe (TTP). When the tip of the probe touches the surface of the part, contact broken within the probe generates an electrical signal for the PHC10 controller which cooperates with the CMM motion controller to read the coordinates of the touch point. The PHD10 probe head drive unit provides manual control of probe head.

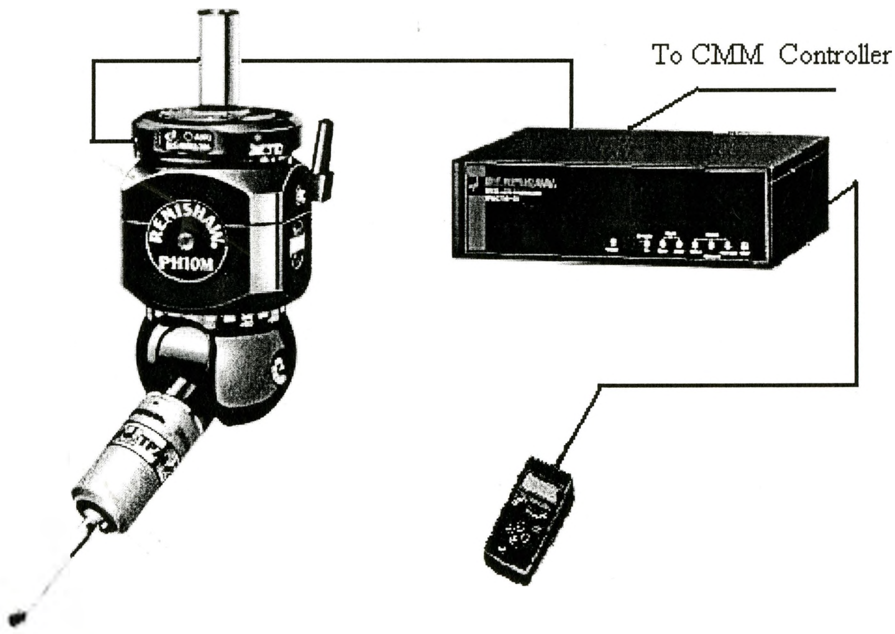


Figure 2.9: Renishaw probe head system [16]

Movement of the CMM along each axis direction is driven by rotary motors, gear boxes, and rack and pinions. Positional information is provided by the optical encoder displacement transducer. (Figure 2.10)

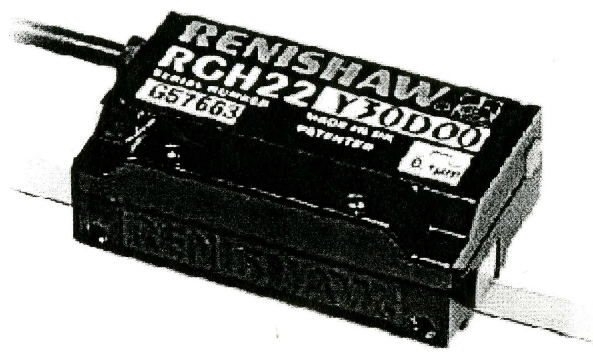


Figure 2.10: Renishaw RGH22 optical encoder displacement transducer

2.4.2 CMM operation

A measurement carried out by CMM normally follows five steps.

First, before a measurement is performed, calibration of probe tip is necessary. This procedure determines the length and orientation of probe styli by using a sphere or cube with known dimensions.

After finishing calibration of the probe tip, the CMM should determine the part coordinate system with respect to the machine coordinate system. This ability of computer controlled CMMs eliminates the alignment of parts to the machine coordinate system. When a part is placed on the table, the part coordinate system can be setup by the following steps. The human operator normally touches three points on the surface of the part to define a plane to fix two rotations and one translation, then takes two points to generate a line to fix one rotation and one translation, and last, takes one point to fix one translation. This procedure can determine the part coordinate system and transformation relationship with machine coordinate system.

The third step is the measurements of points on surface of parts. To accomplish the measurements, the CMM moves the probe to a position remaining a certain distance from the part surface. The probe then moves to the part along the direction normal to the measured surface of the part. When the probe touches the surface of the part, the movement of the touch trigger probe stops and a signal is triggered and transmitted to the CMM controller. The current position is recorded and then the probe starts to move back to a specified distance from the part.

The fourth step is to calculate the metrology of the part by the CMM software and the last step is to display the measurement results and report the related analysis based on the measurement results.

Chapter 3

Online CMM Collision Detection System

3.1 Architecture

The general architecture of CAD directed CMM systems without online collision detection is shown in Figure 3.1. First, the inspection program is generated in the vendor neutral Dimensional Measurement Interface Specification (DMIS) language. Next, CAD software in the control computer translates the DMIS program into a vendor specific command language. The real time CMM controller executes the commands sent from the control computer to measure parts and sends inspection data back. The results are translated back into DMIS and sent to the CAD system for analysis.

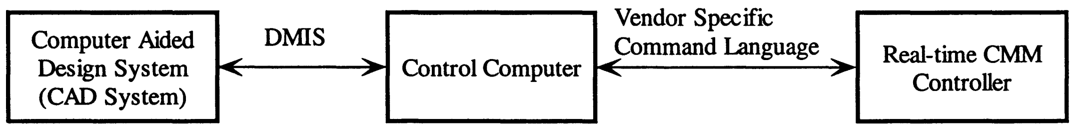


Figure 3.1: CAD-CMM system without online collision detection systems

During measurements, the CMM may collide with inspected parts or itself. Collision may destroy costly probes or inspected parts. Collision may be avoided under the supervision of an operator, but visual checking is unreliable because of the tedious monitoring attention required. Offline collision free path planning has been implemented in recent years. This can simulate the part inspection in an offline mode and find most potential collisions, but drawbacks remain. In particular, this method cannot ensure that collisions due to the environmental changes between the offline simulation and the online inspection period are avoided. Therefore, an online collision detection system is proposed to overcome the disadvantages of operator monitoring and offline collision free path planning.

This system has four components: offline inspection program generation software (CheckMate), online inspection program interpreting software (GEOMeasure), the collision detection system, which is the topic of this thesis, and the real-time CMM controller (OmniTech 5000).

The working flow is shown in Figure 3.2. CheckMate generates a DMIS format inspection program based on the part model in offline mode. After importing the inspection program, GEOMeasure converts it into vendor specific commands and sends them to the collision detection bridge. When the collision detection bridge receives OTC commands, it decides whether this command will change CMM position. If there is no position change, the command is sent directly to the OTC controller. If this command causes movement of the CMM or probe rotation, then this command is sent to the collision check modeller to determine whether any potential collision exists or not. If a collision is detected, the collision detection bridge sends an error message to GEOMeasure and this OTC command is blocked from reaching the OTC controller. If there is no collision, the OTC command is repeated to the OTC controller. The OTC controller obtains a response from the CMM and transmits the feedback to the collision detection bridge. The collision detection bridge then repeats the feedback to GEOMeasure which can create inspection results. CheckMate imports the file, which contains inspection results, and executes offline analysis to generate an inspection report. More detailed descriptions of the components follow.

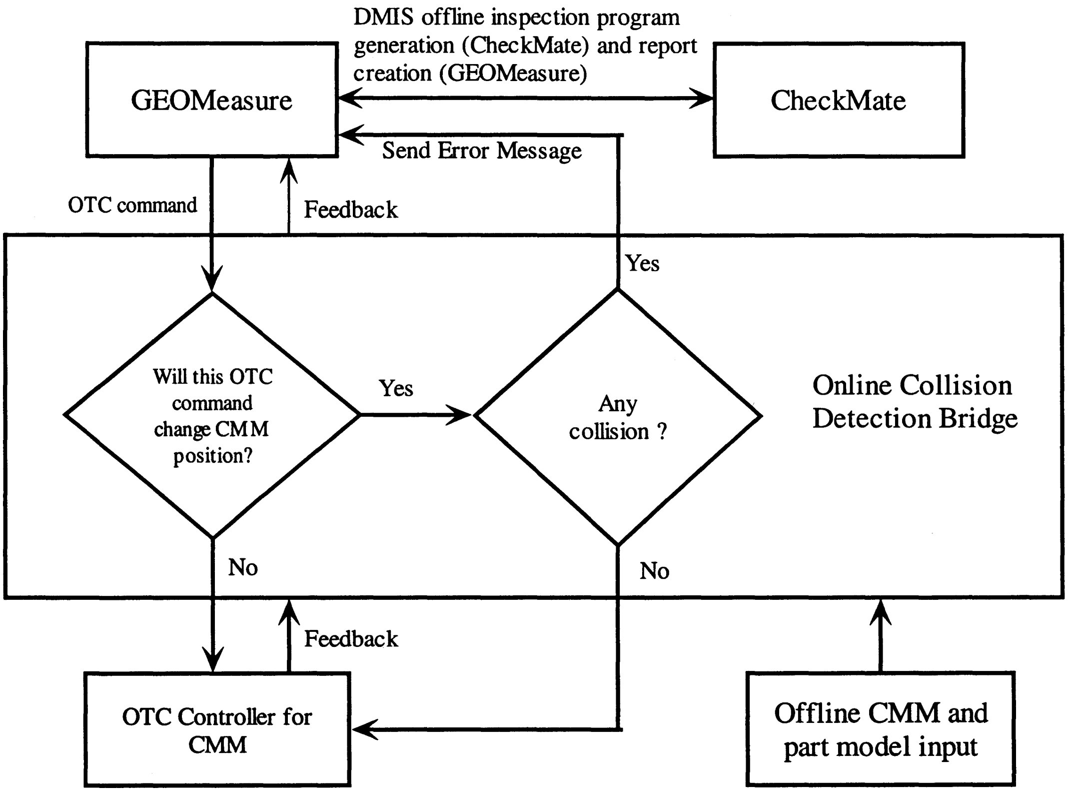


Figure 3.2: Working flow of online collision detection system

3.1.1 CheckMate

CheckMate is an offline programming tool compatible with all modern CMMs [18]. CMM operators import CAD files containing CAD model of parts and manipulate the virtual probe of the CMM in a graphical user interface (GUI) environment to measure parts. After finishing virtual measurements in this environment, CheckMate simulates the procedure of measurements and verifies the inspection program created by CheckMate in DMIS format for GEOMeasure. CheckMate also provides reporting and analysis based on the measurement results from GEOMeasure. Compared to online programming, offline programming reduces the time of using the CMM. The offline characteristics of CheckMate produce significant economic benefit, particularly for the high cost CMMs with large measuring volumes or high precision.

3.1.2 GEOMeasure

GEOMeasure is a Windows NT based online inspection program interpreting software developed by Mutitoyo America Corporation [19]. It can convert a DMIS script into OTC commands.

The DMIS standard [20] was proposed by Consortium for Advanced Manufacturing – International. It provides a common format to exchange inspection data. The same DMIS program can be run in batch mode on different CMMs from different vendors, so DMIS allows a broad choice of solutions to customers.

When GEOMeasure starts to run, it will check the connection with CMM controller. In the initializing procedure, GEOMeasure also selects the right type of probe which is Renishaw touch trigger probe in this thesis. After startup, GEOMeasure imports the inspection program created by CheckMate and translates it into OTC commands to control CMM to carry out measurements. It also produces inspection results from the feedback of CMM controller. CheckMate can use the results to perform further analysis.

3.1.3 OTC Controller

The OTC controller manufactured by Omni-Tech Corporation is an Intel Pentium based CMM motion controller incorporating support for Renishaw probing systems [21]. It adopts an open architecture design to suit a wide range of CMMs. This controller provides an RS-232 serial communication port and 100 BaseTX ethernet for communication with remote computers.

The OTC5000 serial interface protocol used by OTC controller is openly published [22] and has been widely adopted. All commands in the OTC5000 protocol are ASCII strings followed by the mnemonic CK and a character count. The end of the command is carriage return and line feed characters. For example, the command to request CMM status sent from remote computer is: SRSCK3. OTC commands can cause movement of the CMM by a position move command, and probe rotation by probe angle set command. The CMM status can be retrieved by a machine status request command.

3.1.4 Collision Detection System

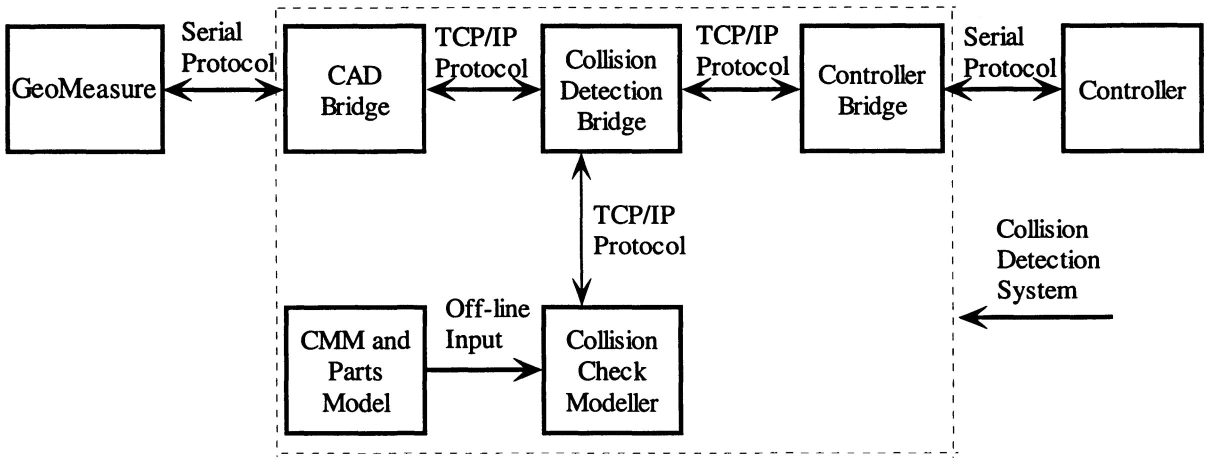


Figure 3.3: CAD-CMM System with online collision detection system

An online collision detection system acts as a guard between the CAD system and CMM controller. It monitors the movement of CMM in real time and prevents collision. Figure 3.3 shows the architecture of an online collision detection system.

The collision detection system consists of a CAD bridge, controller bridge, collision detection bridge, and a collision check modeller (Figure 3.3). The CAD bridge provides intermediate connections between the serial based GEOMeasure and the TCP/IP collision detection system in different communication protocol environments. Likewise, the controller bridge is responsible for exchanging messages between the serial communication protocol used by OTC controller and the TCP/IP protocol used by the collision detection bridge. The collision detection bridge is the exchange center of the online collision detection system that distributes messages to different destinations in different situations. The distribution of its components depends on the requirements of different implementations. The four components can be distributed onto different computers or all reside on the same computer. Every component is connected by the TCP/IP protocol. Because the communication protocol for most systems is still serial, CAD and controller bridges are used for protocol conversion. Newer systems operate using TCP/IP directly.

3.1.4.1 CAD Bridge and Controller Bridge

The CAD bridge and controller bridge are implemented using Microsoft Visual Basic. They are bi-directional converters to connect serial communication protocol and TCP/IP. If bridges receive data by serial communication protocol, they send data by TCP/IP communication protocol. If bridges receive data by TCP/IP, they send data by serial protocol.

3.1.4.2 Collision Detection Bridge

The collision detection bridge processes all commands from the CAD system and feedback from the CMM. When the collision detection bridge starts, its initialing procedure establishes TCP/IP connections with CAD bridge, controller bridge and collision check modeller. It receives commands from CAD system and sends feedback of the CMM to the CAD system. If commands from the CAD system cause a change of the coordinates of the CMM, then the collision detection bridge sends a function call with parameters including current and target coordinate of the CMM and probe to the collision check modeller and requests that it checks whether a collision will occur. If the collision check result sent from collision check modeller indicates that a collision may occur, then the collision detection bridge blocks the command from being sent to the CMM. If a collision will not occur, it allows commands from CAD to reach the CMM. The collision detection bridge always refreshes the current coordinates of the CMM sent from the controller while it is running.

3.1.4.3 Collision Check Modeller

The collision check modeller provides the collision check service for the collision detection bridge. It combines the offline CMM solid model, part solid model, and the real time coordinate of the CMM sent from the collision detection bridge to determine whether a collision may happen or not. The collision check system is implemented using the Scheme ACIS Interface Driver Extension (AIDE) [23] which allows ACIS APIs to be called through extensions of the Scheme language. TCP/IP communication ability is added to ACIS by modifying its C++ source code. When ACIS starts, the TCP/IP service is ready for connecting with the collision detection bridge.

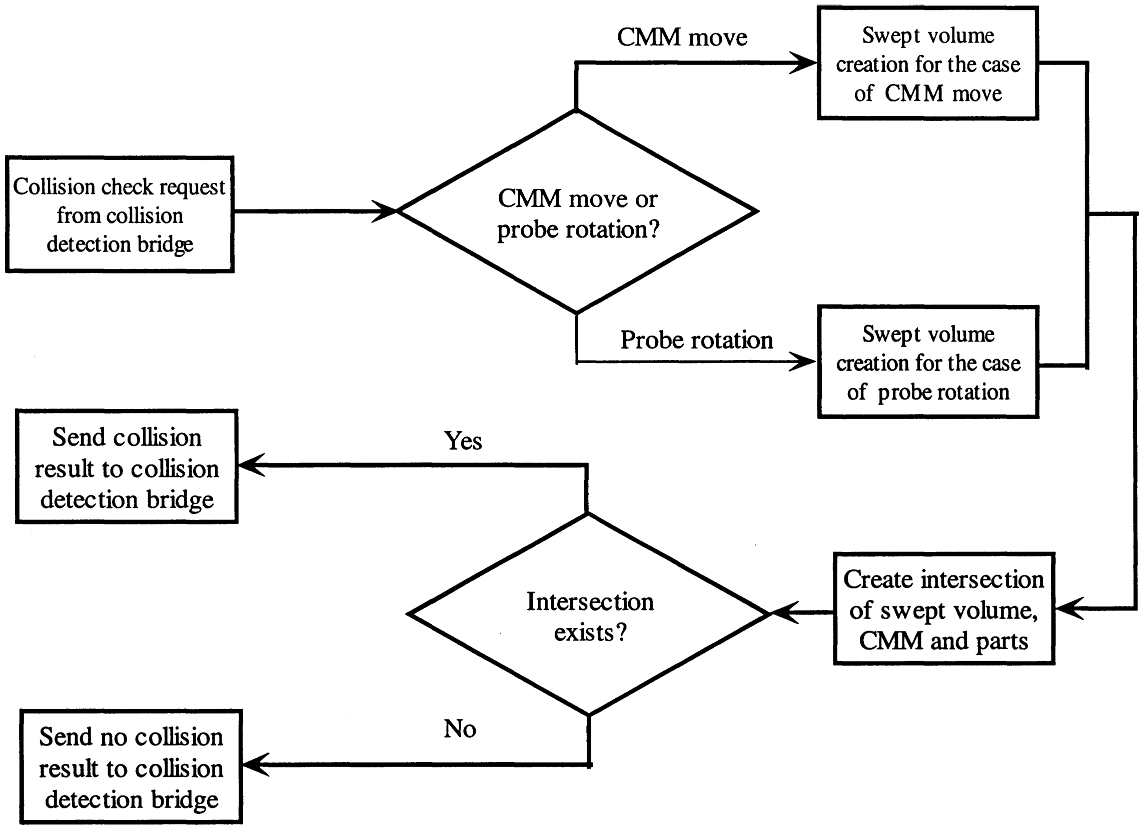


Figure 3.4: The Procedure of collision check

Collision checking is necessary only for the change of CMM position. If the CMM position does not change, collision will not happen. The change of CMM position can be divided into two cases: 1) CMM move; 2) probe rotation. The collision check algorithm is based on the observation that: if the CMM is moving, the probe will not rotate at the same time, and vice versa. The procedure of collision check modeller performs collision check according to the following procedure shown in Figure 3.4. First, the program decides the change of CMM position is the case of CMM move or the case of probe rotation. Second, the program creates swept volumes for different cases. Third, the program applies ACIS to create the intersection of swept volumes, CMM and parts. Fourth, program determines the result. If an intersection exists, then collision is likely. If the intersection is empty, there is no collision. The collision check modeller sends the result to the collision detection bridge by TCP/IP.

3.2 Implementation

3.2.1 CMM Geometric Representation

The CMM used in this thesis is DEA IOTA – P moving bridge CMM. This type of CMM has three orthogonal X, Y, Z axes driven by motors and a rack and pinion. The probe head and probe stylus are installed on the vertical Z axis. A moving bridge provides the Y axis structure, and moves along the table in the X axis direction.

In the solid model of the moving bridge CMM, the bridge and the table are created according to their approximate geometrical shapes. The probe is modeled as a cylinder. The probe head and Z axis component are combined to create another enclosing cylinder with a larger diameter than the probe. When the probe is very close to parts, actual collision may not occur, but the collision check result will indicate a collision. Moving with a small distance between probe and parts is not encouraged in CMM measurements, so the check result keeps the probe from approaching parts too close to avoid potential collision. The horizontally moving component is modelled as a block. It is not necessary to adopt the exact dimension because it never collides with parts or CMM components. Cylindrical shaped motors which drive CMM moving along X, Y, Z axes never collide with parts or other CMM components, so motor models are not included. The above CMM geometric representation keeps the CMM model as simple as possible and speeds the collision detection calculation.

To implement this CMM model, Autodesk Inventor [24] and ACIS are employed. First, CMM components including the granite table, the bridge with four air bearings, and vertical and horizontal movement components are sketched by Inventor according to their dimensions. Second, Inventor exports the components of the CMM model and converts it into a SAT file format which is used in the ACIS 3D Geometric Modeller. Third, ACIS assembles components of CMM model to an integrated CMM model shown in Figure 3.5 before the check collision modeller starts to run.

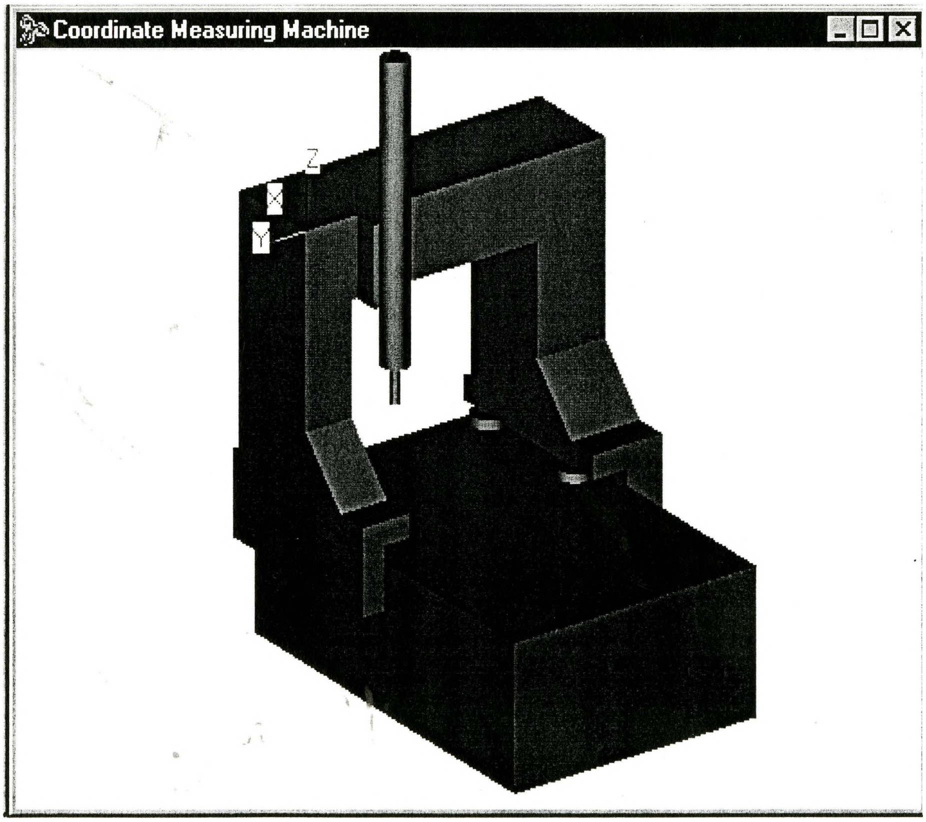


Figure 3.5: A moving bridge CMM model

3.2.2 Swept Volume Generation

As mentioned previously, two cases exist for CMM position change: CMM move and probe rotation. When the probe rotates, the CMM cannot move, and when CMM moves, the probe cannot rotate. The algorithm of swept volume generation is based on the two cases.

3.2.2.1 Swept Volume Created by CMM Move

For a moving bridge CMM, the movement of the bridge and horizontally moving component never collides with parts or other CMM components, so the swept volumes created by the bridge and horizontally moving component are excluded in this algorithm. Since the table is fixed, it has no swept volume. Only the swept volumes created by the vertically moving component and the probe are considered.

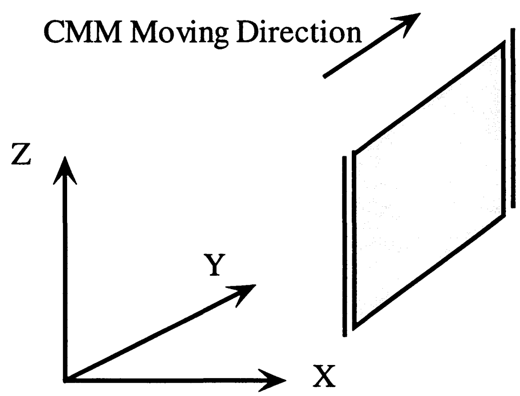


Figure 3.6: Plane surface creation

The vertically moving CMM component and probe are represented as cylinders in the CMM solid model. For the simplicity, the representation of the vertically moving CMM component and probe can be abstracted to their central axes. Swept volume generation of the CMM move follows two steps. First, plane surfaces (See Figure 3.6) of the vertically moving CMM component and probe are created by sweeping their central axes along the CMM moving direction. Second, offsets normal to plane surfaces are generated according to the radius of their represented cylinders. Third, block shaped offsets are combined with solid models of vertically moving CMM component and probe at the target position.

The swept volumes created by vertically moving component and probe while CMM moving have four cases.

Case 1: CMM moving direction is parallel with the central axis of vertically moving component and the probe. In such case, CMM bridge and horizontal moving component do not move, only vertically moving component moves. The swept volumes created are still cylinders shown in Figure 3.7.

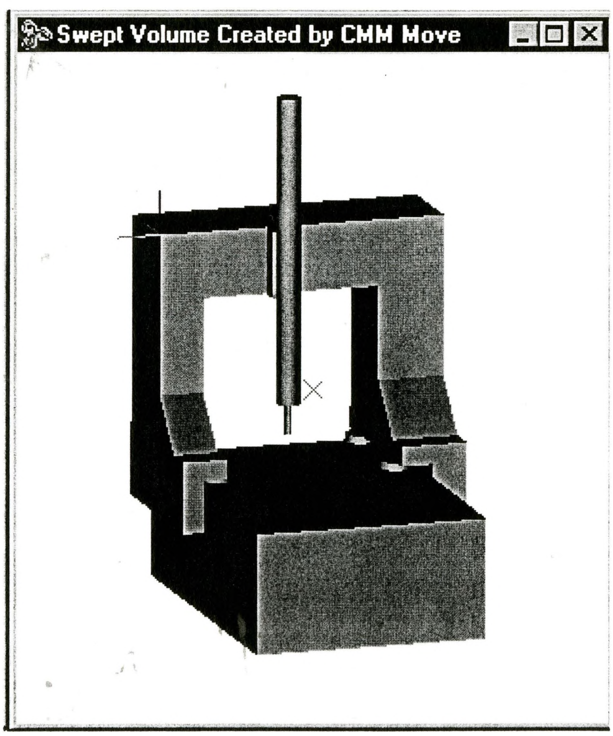


Figure 3.7: CMM move case 1

Case 2: CMM moving direction is parallel with the central axis of vertically moving component but it is not parallel with the central axis of the probe. CMM move is similar to case 1 except that the probe position is different from case 1 in which the probe is parallel with CMM moving direction. The swept volume created by the probe is the union of a block and the probe at the target position. The swept volume created by vertically moving component is still a cylinder (Figure 3.8).

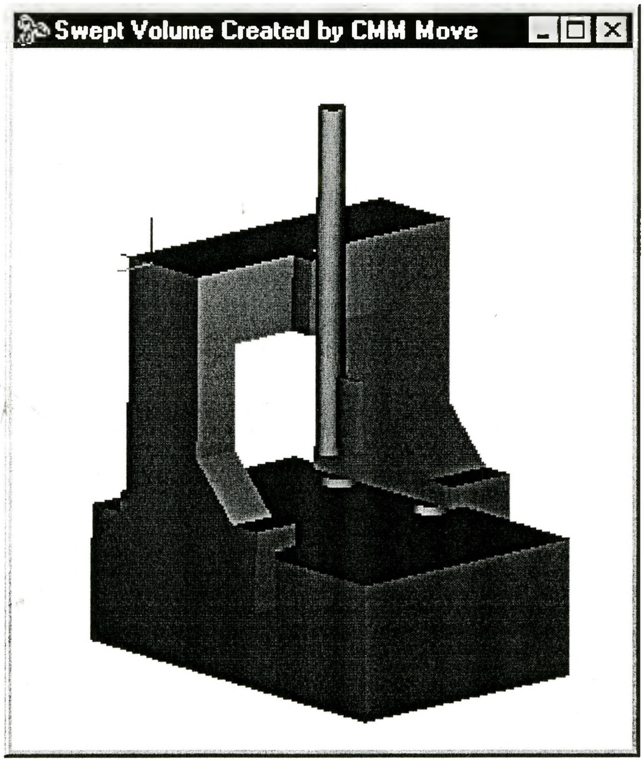


Figure 3.8: CMM move case 2

Case 3: CMM moving direction is parallel with the central axis of the probe but it is not parallel with the central axis of vertically moving component. In this case, the swept volume created by the probe is a cylinder. The swept volume created by vertically moving component is the union of a block and vertically moving component at the target position (Figure 3.9).

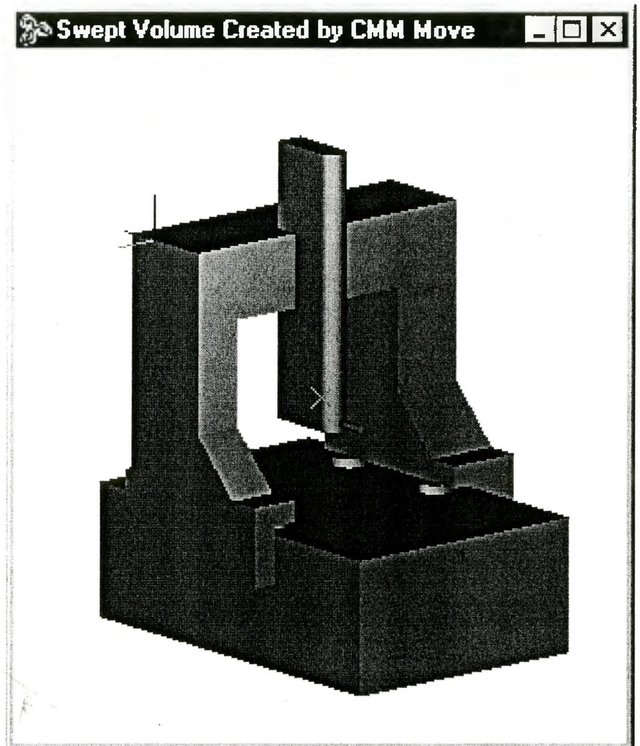


Figure 3.9: CMM move case 3

Case 4: CMM moving direction is not parallel with the central axis of vertically moving component and the probe. Swept volumes created by the probe and vertically moving component are the union of blocks, vertically moving component and probe at the target position. (Figure 3.10)

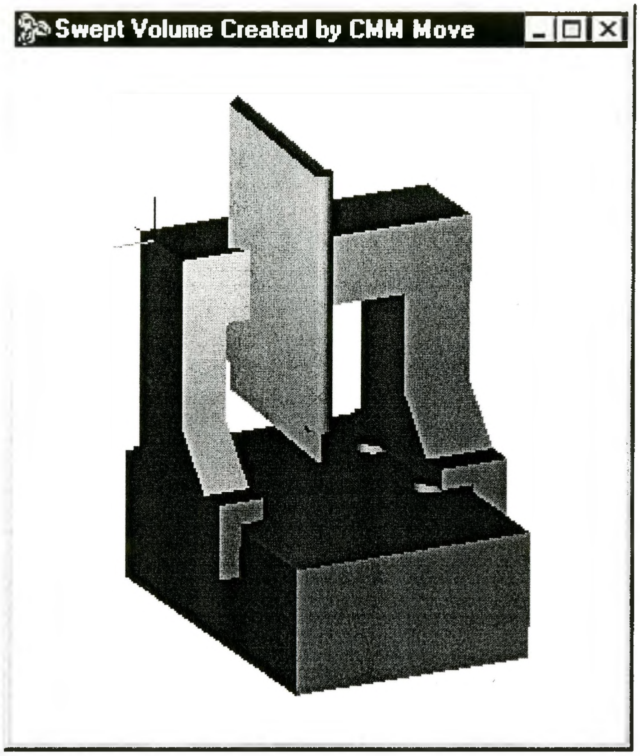


Figure 3.10: CMM move case 4

3.2.2.2 Swept Volume Created by Probe Rotation

The CMM position is determined by the X, Y, Z axes and the A and B angles of the probe as shown in Figure 3.11. The probe rotating point is denoted as (X_0, Y_0, Z_0). Angle A is the angle between Z axis and the probe central axis and angle B is the angle between X axis and the projection of the probe central axis on XY plane. The range of A angle for Renishaw PH10 probe is from 0 to 105 degrees and B angle is from –180 degrees to 180 degrees.

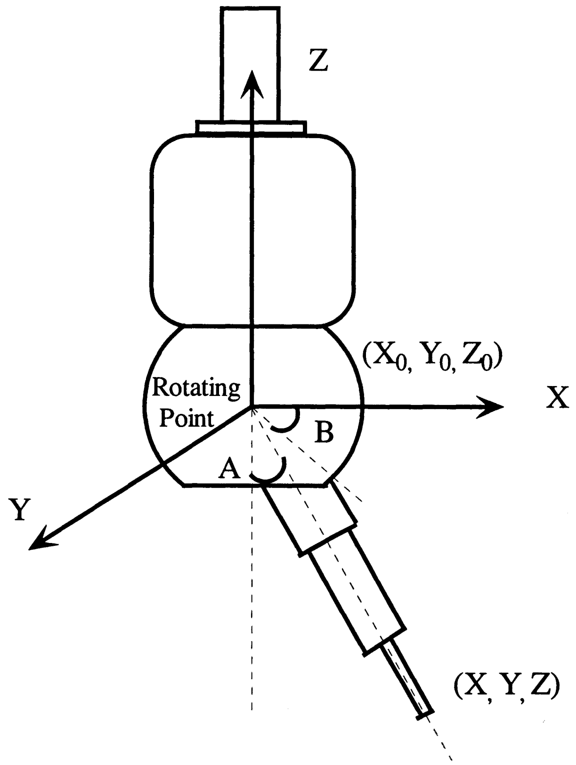


Figure 3.11: Probe in Cartesian coordinate system

From this coordinate system, the coordinate of the tip of the probe (X, Y, Z) can be obtained by the following matrix if the coordinate of rotating point (X_0, Y_0, Z_0) and A, B angles are known.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + l \begin{bmatrix} \sin A \cos B \\ \sin A \sin B \\ \cos A \end{bmatrix}$$

Where l is the length of the probe.

When the probe rotates, its rotation angle in 3D space can be divided into A angle and B angle rotation. Let A and B be denoted as A, B angles before probe rotation, ΔA be A angle variation and ΔB angle be B angle variation. The coordinate of the tip of the probe (X, Y, Z) after probe rotation can be calculated by following matrix.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + l \begin{bmatrix} \sin(A+\Delta A)\cos(B+\Delta B) \\ \sin(A+\Delta A)\sin(B+\Delta B) \\ \cos(A+\Delta A) \end{bmatrix}$$

Swept volumes created by the probe rotation have five cases.

Case 1 is that A angle is zero and only B angle changes. In this case, the probe rotates around its central axis, so the swept volume created by the probe is the probe itself (Figure 3.12).

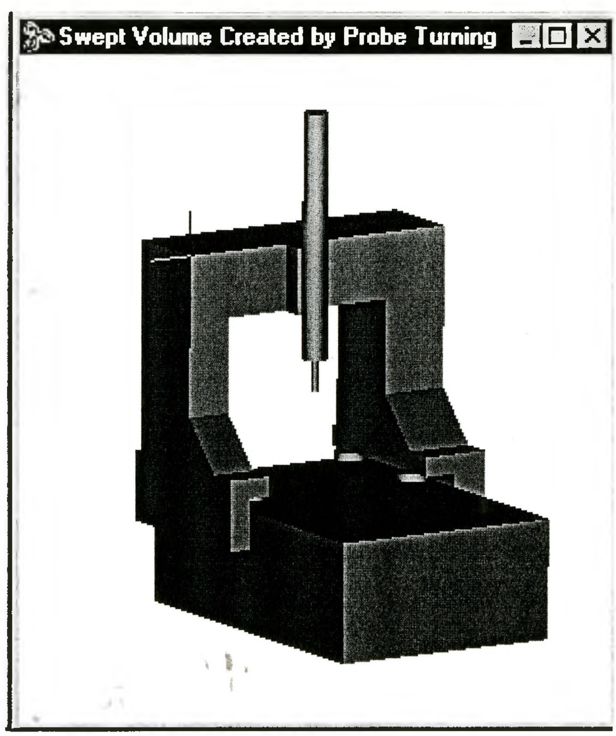


Figure 3.12: Probe rotation case 1

Case 2 is that only the A angle changes. The swept volume created by the probe is a sector (Figure 3.13). The procedure of creating the swept volume follows two step. First, A sectorial shaped surface is created by sweeping the central axis of the probe with ΔA variation around probe rotating point. Second, offsets normal to the sectorial shaped surface are generated according to the radius of the cylinder which represents the probe.

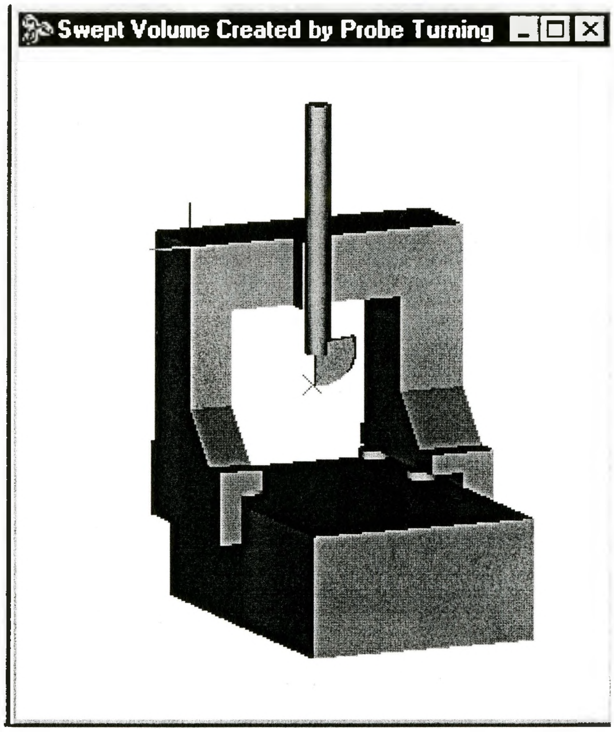


Figure 3.13: Probe rotation case 2

Case 3 is that only the B angle changes and the A angle is 90 degrees. The procedure of creating swept volume is similar to case 2. The swept volume is a sector (Figure 3.14). If the change of B is 360 degrees, the swept volume is a cylinder.

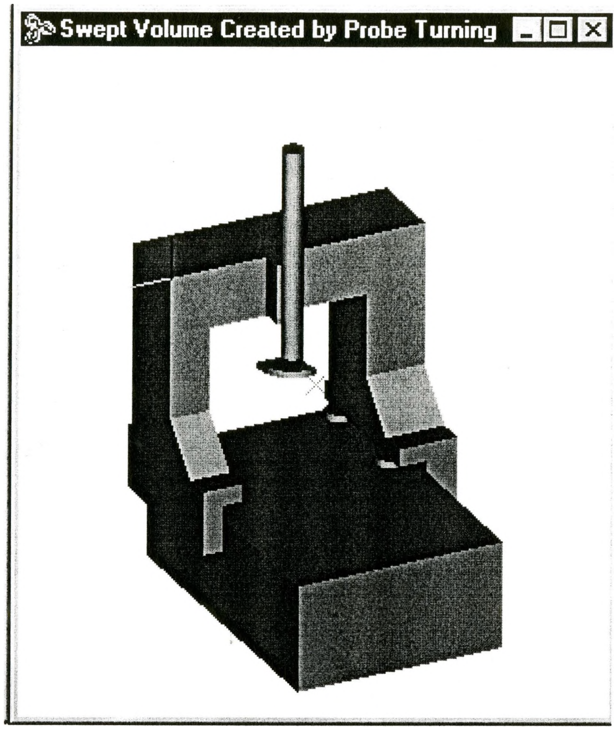


Figure 3.14: Probe rotation case 3

Case 4 is that only B changes but angle A is not zero and B angle is not 90 degrees. The procedure of creating the swept volume also follows two step. First, A conic shaped surface is created by sweeping central axis of probe with ΔB variation around Z axis. Second, offsets normal to the conic surface are generated according to the radius of the cylinder which represents the probe. The swept volume created in this case is a cone if the change of B is 360 degrees or part of cone (Figure 3.15) if the change of B is less than 360 degrees.

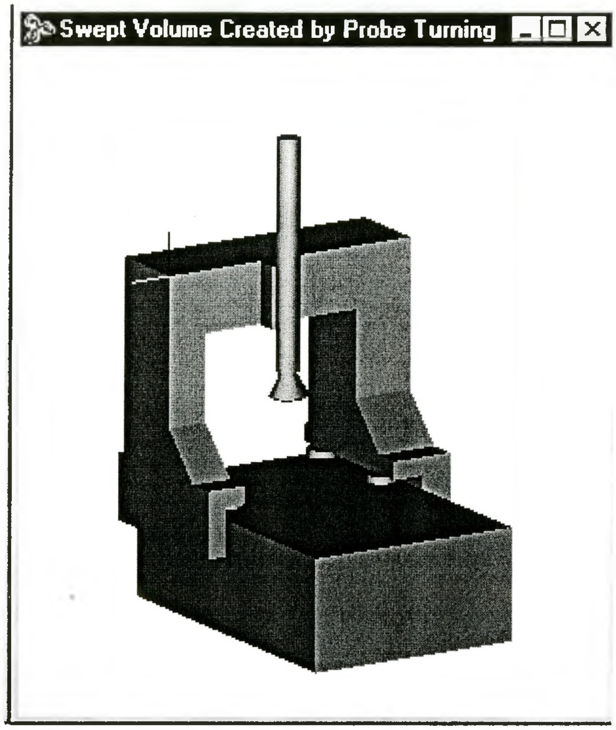


Figure 3.15: Probe rotation case 4

Case 5 is the case other than case 1, 2, 3 and 4. Both the A angle and B angle change in this case. Since motors which drive A angle and B angle changes may not start at the same time and the speed of both motors may not be same, the exact trajectory of the probe rotation is uncertain. The swept volume created by probe rotation in this case is the space including all possible probe rotation trajectories. The space is created by two extreme situations as shown in Figure 3.16. The first extreme situation is that the motor drives the probe to change the A angle, and then the motor drives the probe to change the B angle. The second extreme situation is the motor drives the probe to change the B angle, and then the motor drives the probe to change the A angle. The two extreme situations create an enclosed space which includes every possible probe moving trajectory.

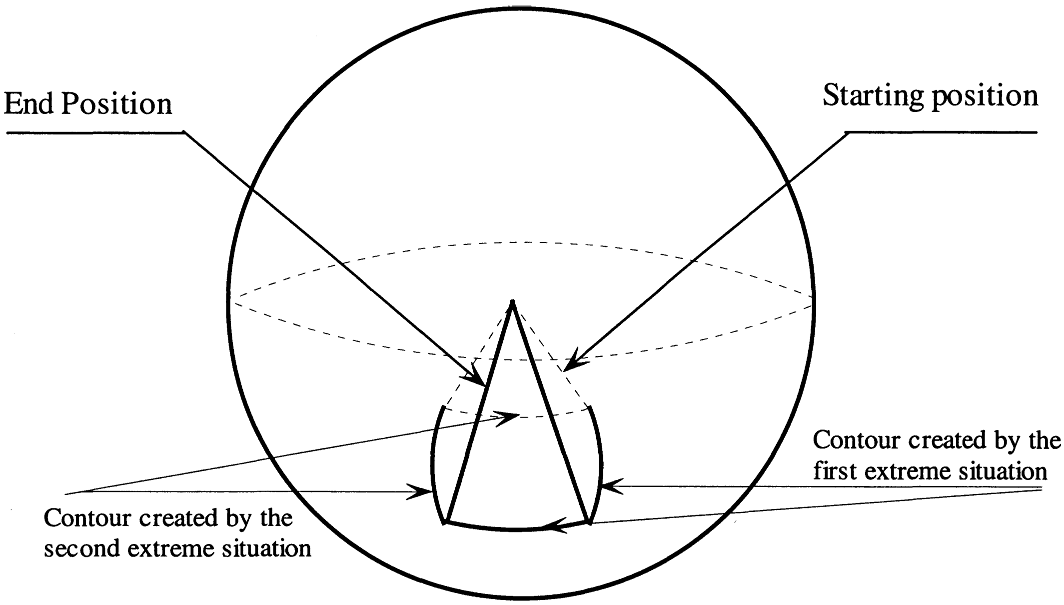


Figure 3.16: The possible space by considering two extreme situations

The procedure of creating the swept volume in ACIS follows three steps. First, a sectorial surface with ΔA angle is created. Second, a swept volume is generated by sweeping this sectorial surface around the Z axis. Third, an offset with the radius of the cylinder representing the probe is applied to the swept volume. After the three steps, the swept volume is created (Figure 3.17)

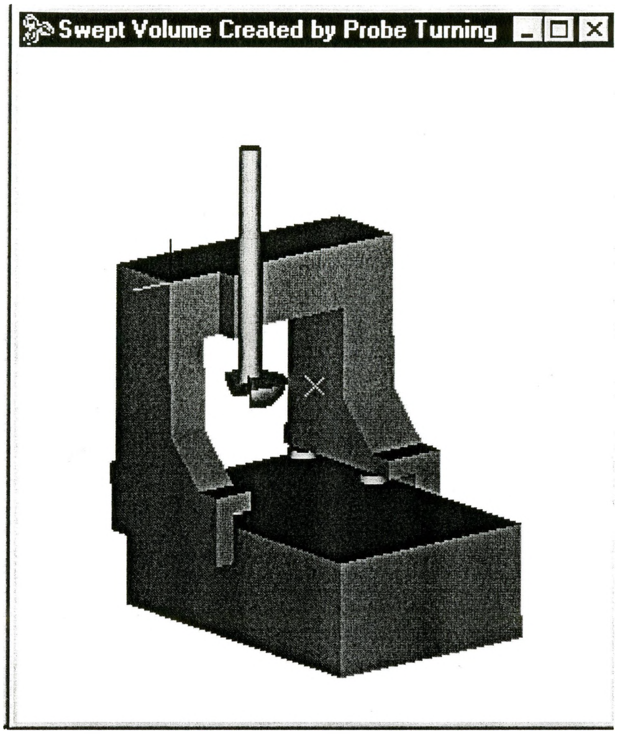


Figure 3.17: Probe rotation case 5

3.2.3 Communication Protocols Connections

Two communication protocols, which are serial communication protocol and TCP/IP are used in online collision detection system. GEOMeasure and OTC controller use serial communication protocol. Collision detection switch and collision check system use TCP/IP. CAD bridge and controller bridge implemented by Microsoft Visual Basic undertake the work of bridging serial and TCP/IP.

In CAD bridge and controller bridge, Microsoft communications control and Winsock control are used. Microsoft communications control is responsible for serial communication and Winsock control is responsible for TCP/IP connection. Figure 3.18 shows the conversion between serial protocol and TCP/IP.

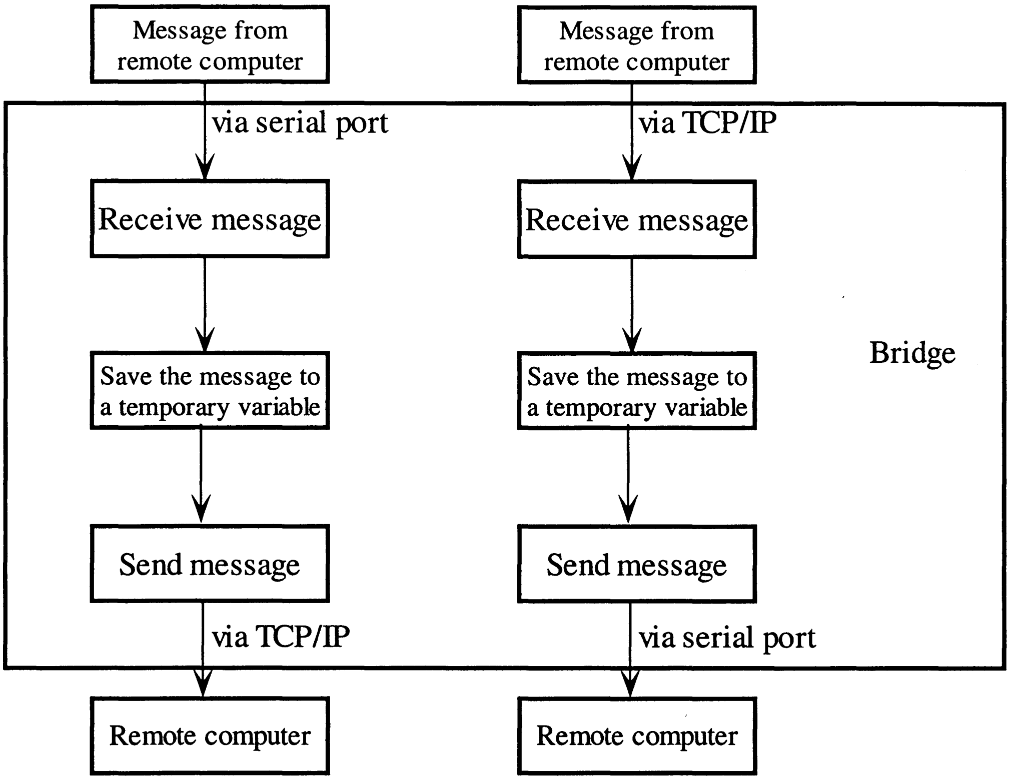


Figure 3.18: Conversion process between serial and TCP/IP protocols

When the bridge program starts to run, it performs an initializing procedure in which settings for communication are loaded from initialization file. The initializing procedure sets the local port number and is ready to accept the connection request from the remote computer for TCP/IP connection. If a remote computer sends a connection request to the bridge, it accepts its request and a TCP/IP connection is established. The initializing procedure also chooses the serial port number, serial communication format and opens the local serial port ready to accept data from the remote serial port. After initialization, the bridge program can process the conversion of serial protocol and TCP/IP. When a message sent from the remote serial port reaches the local serial port, an event is triggered. The Microsoft communications control reads the message from the buffer and stores it in a temporary variable. The Winsock control then sends this message to the remote computer via TCP/IP. Similar to the conversion from serial to TCP/IP, when a message is sent from the remote computer via TCP/IP, an event is triggered, Winsock control acquires the message and puts it in a temporary variable. The Microsoft communications control then sends it to the remote serial port.

3.2.4 ACIS Network Extension

The online collision detection system requires that TCP/IP be available in ACIS to receive the collision checking request and send feedback out. ACIS version 8.0 does not provide a TCP/IP function, so ACIS must be extended to include TCP/IP.

The TCP/IP function is implemented by Winsock programming in the ACIS network extension. When ACIS starts to run, it also loads the TCP/IP segment. To start TCP/IP service (Figure 3.19), the Winsock version is checked at first. The current Winsock version is 2.0 in this implementation. Then, the program creates a listening socket. After the listening socket is set, the operation mode is selected. In this program, polling mode is used because it can process moderate TCP/IP flow in current configuration and simplify program structure. Next, a local port, TCP version, and IP address are established. After acquiring these parameters, the program binds the IP address to the listening socket. The next step is that the program sets the listening socket into listening mode and is ready to accept a TCP/IP connection. When a remote computer requests a connection, it accepts its request, and a TCP/IP connection is setup.

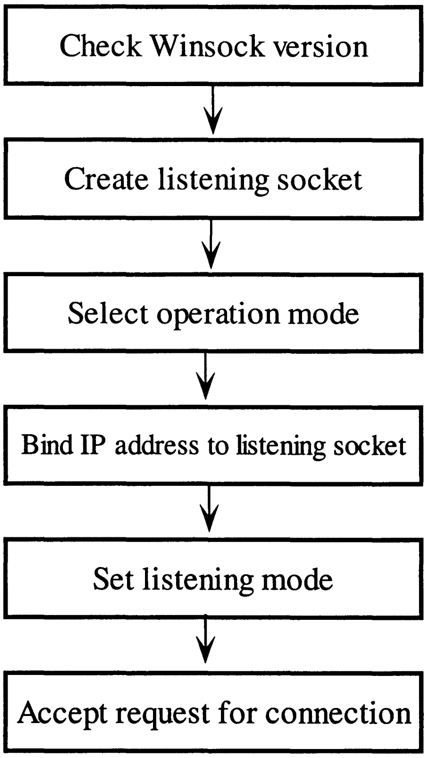


Figure 3.19: ACIS TCP/IP Startup

After a TCP/IP connection is established, the program is ready to receive and send data to the remote computer. The process of receiving and sending data is shown in Figure 3.20. Two buffers, which are input buffer and output buffer, are defined in the program. When a collision check request from the remote computer arrives, it will be placed into the input buffer and then be sent to the collision check modeller. When the collision check modeller sends the feedback to ACIS, the ACIS program puts the feedback into the output buffer and then sends it to the remote computer via TCP/IP.

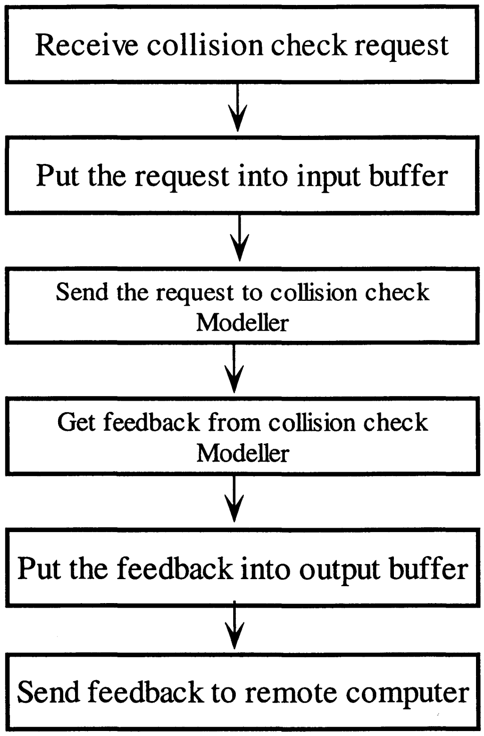


Figure 3.20: Procedure to process collision check request

Chapter 4

Experiments and Performance Analysis

4.1 Experiment Configuration

The hardware configuration (Figure 4.1) of experiments is: DEA IOTA – P moving bridge coordinate measuring machine (Figure 2.9), Renishaw PH10 probe head system (Figure 2.10), OTC5000 CMM motion controller, personal computer A with a 1 GHZ AMD CPU, 256M memory, 100 BaseTX Ethernet card, personal computer B with a 600 MHZ Intel CPU, 256M memory, 100 BaseTX Ethernet card, tested parts which are a yoke (Figure 4.2a) and a ball bar with fixture (Figure 4.2b). The two computers are wired to a switch in a local area network. A serial cable connects COM port1 and COM port2 of computer A because GEOMeasure and CAD bridge run on the same computer.

Mitutoyo GEOMeasure 6000 that translates DMIS commands into OTC commands, CAD bridge and collision detection bridge implemented in Visual Basic 6.0, collision check modeller run on the computer A. The controller bridge runs on the computer B. OTC5000 serial Interface Protocol is the language used by OTC5000 controller.

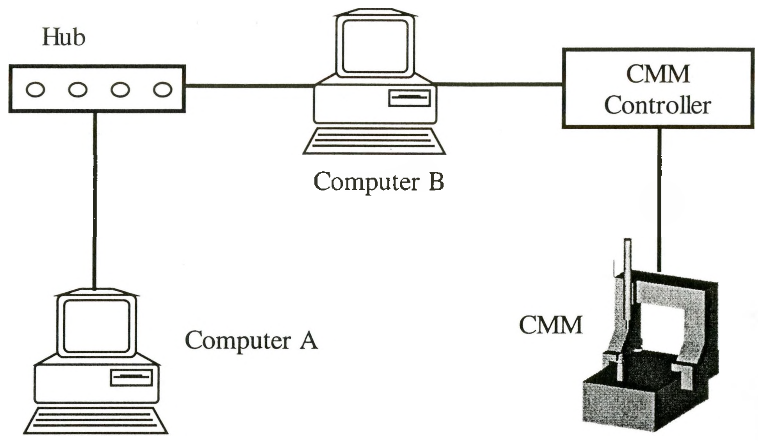


Figure 4.1: Hardware configuration of experiments

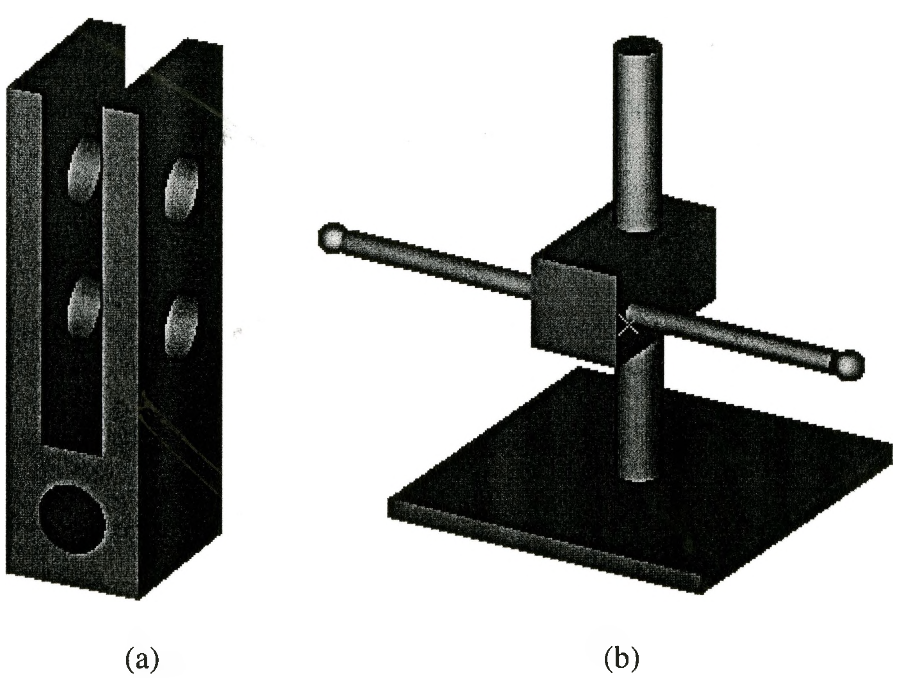


Figure 4.2: Tested parts

4.2 Experiment Procedure

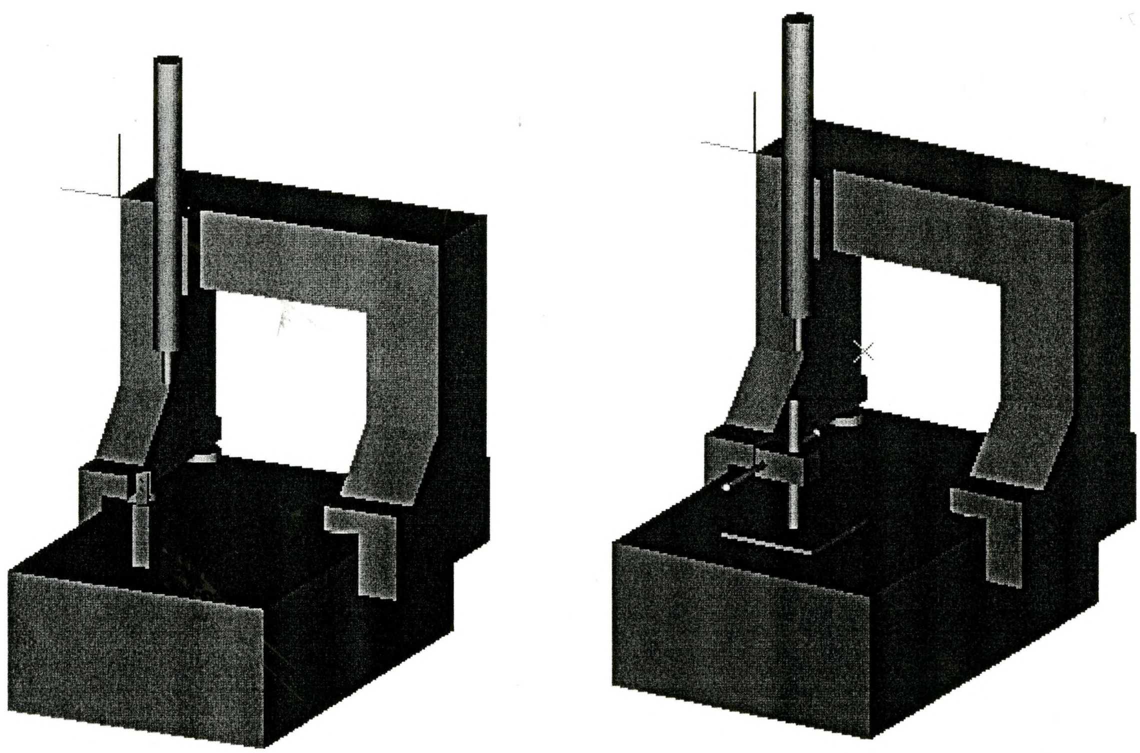
Experimental testing of the collision avoidance system consisted of the following steps: offline DMIS program generation, manual measurement to establish part coordinate system, offline combination of part solid model with CMM solid model, and online measurements.

In the offline DMIS program generation step, CheckMate [18] imports a CAD model of the tested part. The CMM inspection path is created in CheckMate using a graphical user interface. CheckMate then outputs a DMIS program containing the desired inspection path. This DMIS program is separated into two sections. The first section defines the part coordinate system with respect to the CMM coordinate system by manually measuring part datum features. The second section drives CMM automatically to measure the tested part.

In order to combine the part solid model with the CMM solid model, the part coordinate system must be defined. The tested part must be clamped on the fixture and supported on the CMM granite table. The program to manually locate the part coordinate system in generated DMIS program is then executed by GEOMeasure. First, the operator uses a joystick to operate the probe to approach the selected feature on the tested part. Second, three points on the selected surface are touched by the probe tip. GEOMeasure can determine a plane by the measurement results of the three points. Third, two points on another surface of the tested part are selected. GEOMeasure can determine a line by the two points. Fourth, one point is selected on the surface other than the previous two surfaces. After finishing the four steps, a part coordinate system is defined.

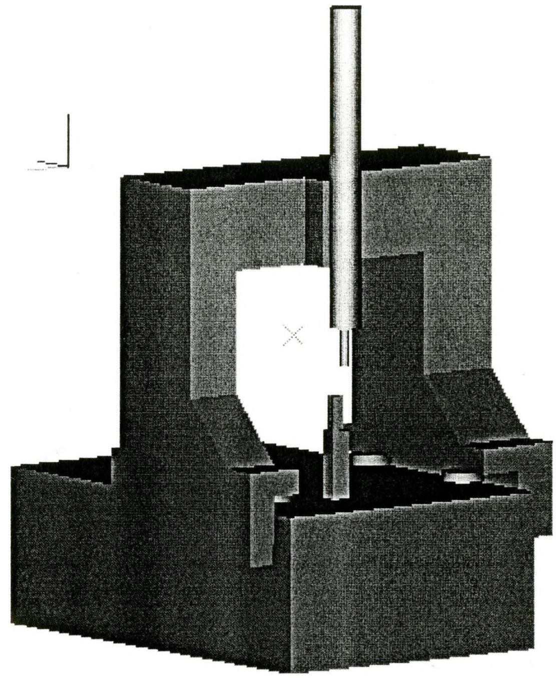
Once the part coordinate system has been defined, the part solid model can be combined with the CMM solid model. Figure 4.3 shows the combination of the CMM model and the tested part model.

The last step of the experiment is online measurements. When the automatic measurement program section in the generated DMIS program is executed, commands to move CMM or rotate probe are sent to the collision check system.

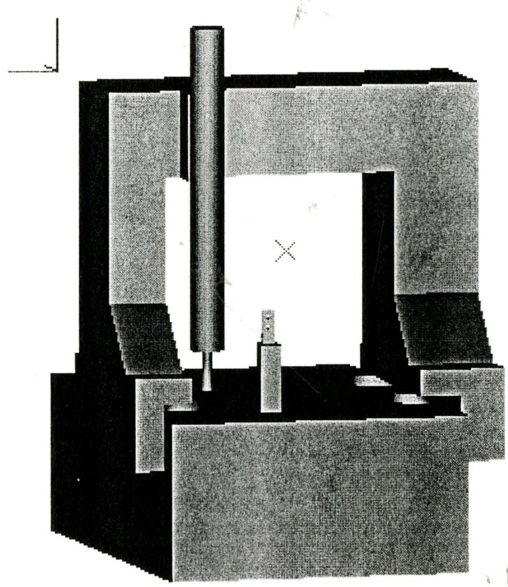


(a) (b)

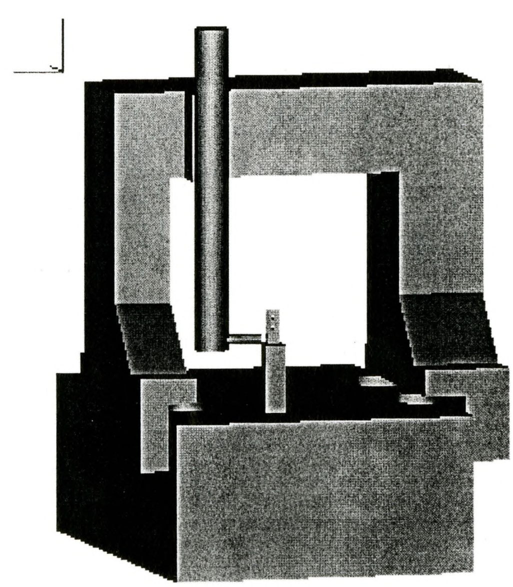
Figure 4.3: Combination of CMM and tested part model



(a)

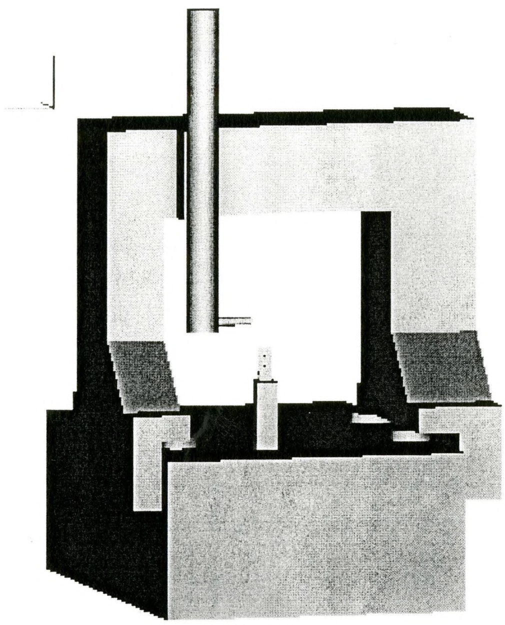


(b)

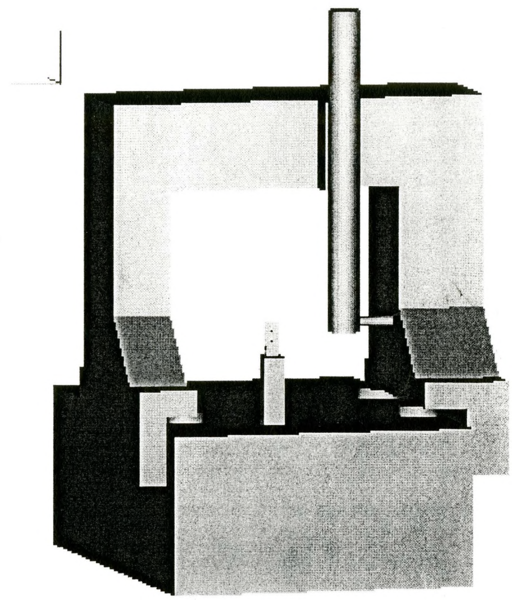


(c)

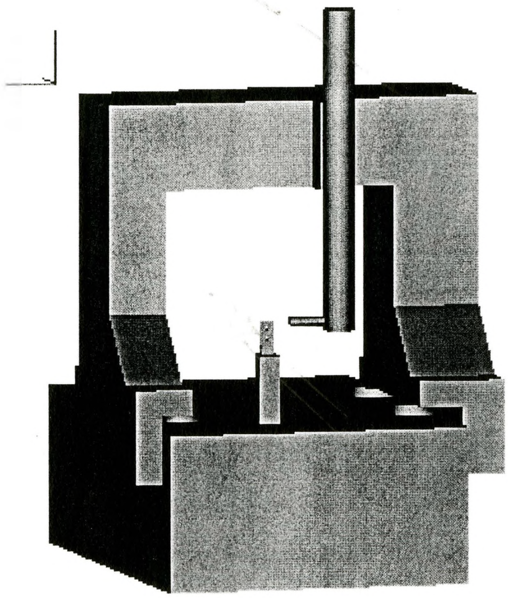
Figure 4.4: Online measurements of yoke



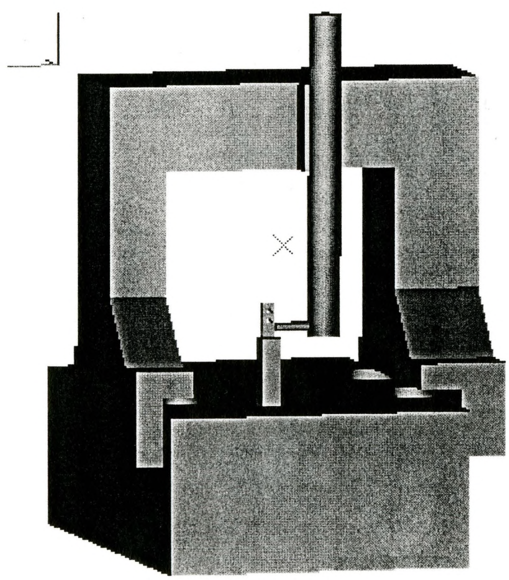
(d)



(e)



(f)



(g)

Figure 4.4: Online measurements of yoke

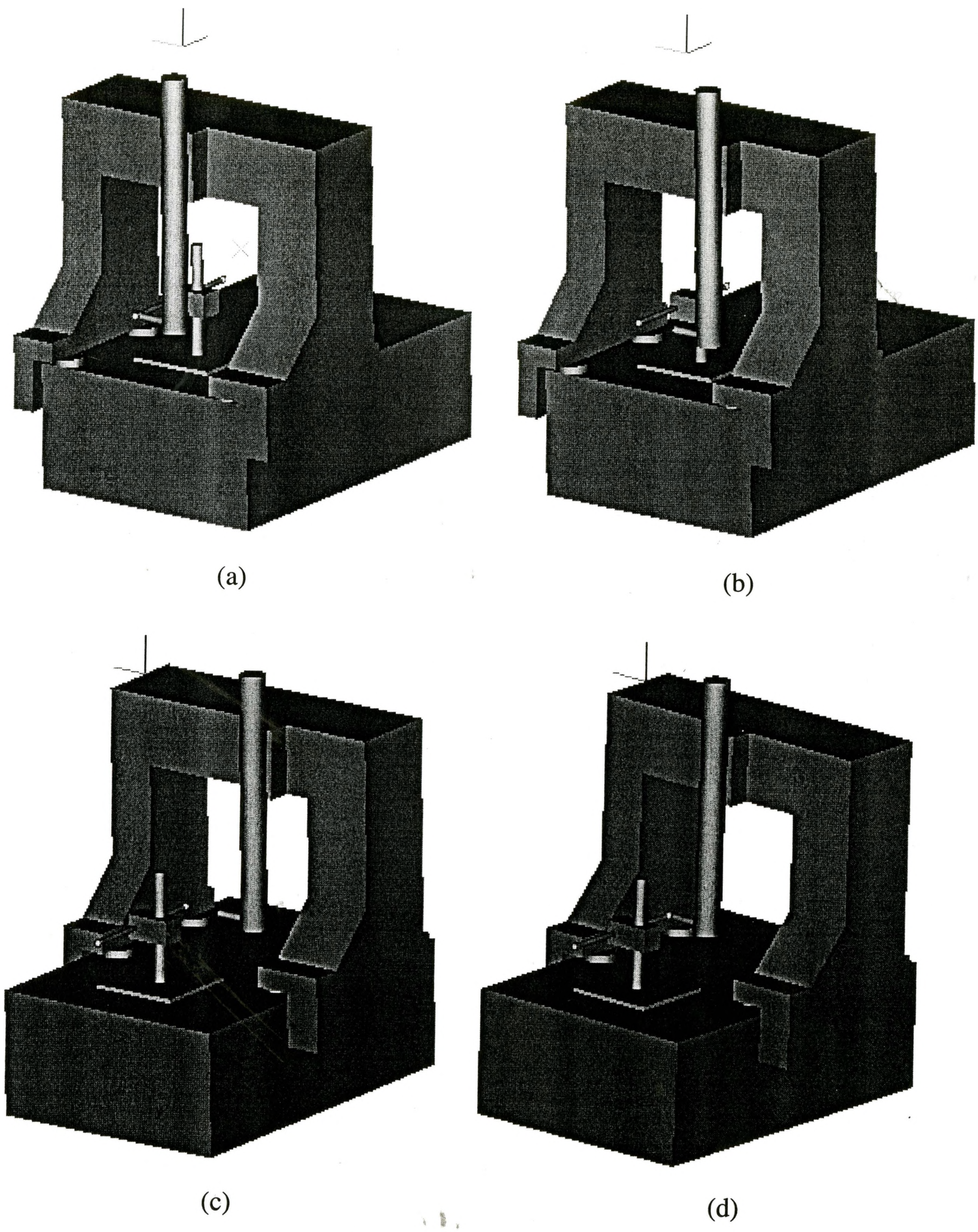


Figure 4.5: Online measurements of ball bar

Figure 4.4 and Figure 4.5 show the process of online measurements of the yoke and ball bar tested parts. For the yoke part, Figure 4.4a shows the initial CMM position with angle A and B equal to zero degrees. The CMM then moves to the position shown in Figure 4.4b. At this position, the probe rotates to angle(A = 90, B = 90) (See Figure 4.4c). The probe then measures several points. After the measurements, CMM moves to the position shown in Figure 4.4d and then to the position in Figure 4.4e. At this position, the probe rotates to angle (A = 90, B = –90) (See Figure 4.4f). The CMM then moves to the position shown in Figure 4.4g to measure points on the surface of the tested part. If the CMM directly moves from the position in Figure 4.4c to the position in Figure 4.4e, an interference between the probe and the tested part exists. The collision detection system will block this command.

For the ball bar, Figure 4.5 demonstrates how online collision detection system avoids potential collision. Two DMIS programs are run to make the comparison. One program that has no collision directs CMM from the position in Figure 4.5a to Figure 4.5b, and then to Figure 4.5c, at last to Figure 4.5d. The other program directly moves CMM from the position in Figure 4.5a to Figure 4.5d. Obviously, this movement causes a collision between the probe and the ball bar fixture. The online collision detection system blocks the command to move the CMM and avoids potential collision.

4.3 Performance analysis

In order to evaluate the performance of the online collision detection system, instructions to measure timing were added into the collision check system. The timing instructions in collision check modeller record the time to check the collision and the timing instructions in collision detection bridge to record the time span from sending collision check request to receiving the feedback, which consists of the back and forth communication time and collision check time. The resolution of the timing is one millisecond. This resolution is enough for evaluating the performance of the online collision detection system as CMM measurements normally last several minutes. The CMM spends the most time in measurements, so if online collision detection consumes a small amount of the overall elapsed time, it is usable in practical applications.

CMM measurement steps	Communication time (ms)	Collision check time (ms)	Total time (ms)
Figure 4.4a – Figure 4.4b	11	380	391
Figure 4.4b – Figure 4.4c	31	390	421
Figure 4.4c – Figure 4.4d	10	410	420
Figure 4.4d – Figure 4.4e	20	381	401
Figure 4.4e – Figure 4.4f	30	391	421
Figure 4.4f – Figure 4.4g	19	451	470
Average	21	391	412

Table 4.1: Time of measurements of yoke

The time recorded in the measurements of yoke is shown in Table 4.1 and the time of measurements of ball bar is shown in Table 4.2. The average communication time is 21.375 and 23.33 milliseconds, collision check time is 398.5 and 347 milliseconds and total average time is 411.875 and 370.33 milliseconds. Table 4.1 and Table 4.2 also show that communication time and collision check time in each measurement step are similar. This is because, in the communication process, the online detection system sends and receives the same number of bytes in each step. The ACIS part solid model and CMM model are constant, implying that the geometric complexity represented in ACIS is fixed. Therefore, the time for collision checking is constant.

CMM measurement steps	Communication time (ms)	Collision check time (ms)	Total time (ms)
Figure 4.5a – Figure 4.5b	30	311	341
Figure 4.5b – Figure 4.5c	20	350	370
Figure 4.5c – Figure 4.5d	20	380	400
Average	23	347	370

Table 4.2: Time of measurements of ball bar

Table 4.1 and Table 4.2 show that the online collision detection system can meet the requirement of CMM measurements because it does not significantly delay the CMM inspection process. If the part model is more complex, the representation of the part model in ACIS is more complex. More complex representation implies that more calculations in ACIS to get results. Hence, a more complex part model requires more time to check for collision.

Chapter 5

Summary and Future Work

5.1 Achievements

The online collision detection system described in this thesis realizes collision detection in real time mode. Following is the list of the achievements of the online collision detection system.

- The online collision detection system can avoid interference between the CMM and parts during inspection. It can also avoid the collision between the probe and other components of CMM.
- The online collision detection system is a flexible system in which each component can be distributed on different computers according to different requirements of applications.
- Connection between serial communication protocol and TCP/IP communication protocol is implemented in this system. This feature brings seamless integration of the online collision detection system and legacy systems, such as GEOMeasure, and the OTC controller software.
- The ACIS network extension enables data exchange across a network. This feature extends ACIS applications from a single computer to computers in a network environment.

- Processing time is a small fraction of the CMM total probing time and hence does not interfere with the primary function of CMM.

5.2 Limitations

The implementation of the online collision detection system is for moving bridge coordinate measuring machines with Renishaw probe heads. This system needs off line input of the CMM model and part model to properly realize collision detection. The accuracy of collision detection depends on the resolution of CMM model and the part model.

Different types of CMMs have different kinetic characteristics, but the collision check algorithm for each of these CMMs is common. However, if two CMM machines operate in a common workspace, the collision check algorithm developed in this system must be extended.

The implementation of collision detection bridge in this system is suitable for OTC5000 serial interface protocol. If other controller languages, such as I++ [25], are used in this system, the system should be modified to adapt to the different language format.

Most current communication protocols in industrial applications of CAD directed CMM systems are serial or TCP/IP. If other communication protocols are used in CAD directed CMM systems, the corresponding protocols need to be included in the CAD bridge or the controller bridge.

5.3 Future Work

The online collision detection system in this thesis is a preliminary collision detection system. It needs to be improved to adapt to broader applications. Following is suggested future work of the online collision detection system.

- The CMM may move along a series of position points in a measurement, multi-point move support needs to be included in this system.
- Automatic registration of the part model and CMM model coordinate system in the ACIS environment is required to integrate with this system to increase the efficiency.
- A more precise CMM model is necessary to implement collision detection during inspection of small features such as slots and holes.
- As the CMM is more and more integrated with manufacturing robots to automate the entire production and inspection process, the CMM and robots may operate in a common workspace. Interference between the CMM and the robots may occur. Collision detection systems applied to the CMM and robots are therefore necessary. While developing this system, the reusable components in current online collision detection system are: CAD bridge, controller bridge, collision detection bridge, TCP/IP extension in ACIS, the portion in collision check modeller related to deciding whether a collision occurs or not. The reusable components can reduce the system development time and provide more flexibility while deploying applications.

References

[1] Sukhan Lee and Hadi Moradi, "A real-time dual arm collision avoidance algorithm for assembly." Journal of Robotic System, 2001, volume 18, No. 8, pp. 477 – 486.

[2] Lin, Yueh-Jaw and Murugappan Prabakar. "New algorithm for determining a collision-free path for a CMM probe." International Journal of Machine Tools and Manufacture, v 39, n 9, 1999, pp. 1397-1408.

[3] C.G. Lu, D. Morton, M.H. Wu, P. Myler. "Genetic algorithm modelling and solution of inspection path planning on a coordinate measuring machine (CMM)." International Journal of Advanced Manufacturing Technology, v 15, n 6, 1999, pp. 409-416.

[4] E. Lu, J. Ni, S.M. Wu. "Algorithm for the generation of an optimum CMM inspection path." Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, v 116, n 3, Sept, 1994, pp. 396-404.

[5] Hong-Tzong Yau and Chia-Hsiang Meng. "Path planning for automated dimensional inspection using coordinate measuring machines." Proceedings - IEEE International Conference on Robotics and Automation, v 3, 1991, Proceedings of the 1991 IEEE.

[6] Zone-Ching Lin, Chein-Chung Chen. "Collision- free path planning for coordinate measurement machine probe." International Journal of Production Research, vol. 39, No. 9, 2001, pp. 1969-1992.

[7] P. Jimenez, F. Thomas, C. Torras. "3D collision detection: a survey." Computers & Graphics, vol. 25, 2001, pp. 269-285.

[8] S. Cameron. "Collision detection by four-dimensional intersection testing." IEEE Transactions on Robotics Automation, volume 6, No. 3, 1990, pp. 291-302.

[9] K. Abdel – Malek, D. Blackmore, K. Joy, "Swept volumes: foundations, perspectives, and applications." International Journal of Shape Modeling.

[10] Ed. Taylor, TCP Complete, McGraw-Hill Professional, New York, 1998, page 7.

[11] Andrew S. Tanenbaum, Computer Networks, 3rd Edition, Prentice Hall, New Jersey, 1996, page 521.

[12] W. Richard Stevens, TCP/IP illustrated, Addison-Wesley Publishing, Massachusetts 1996, page 30.

[13] Bob Quinn, Dave Shute, Windows Sockets Network Programming, Addison-Wesley Publishing, Massachusetts, 1996, page 15.

[14] ANSI/ASME B89.4.1-1997. Methods for performance evaluation of coordinate measuring machines, pp. 13-17.

[15] Bosch, Coordinate Measuring Machines and Systems, Marcel Dekker Inc., 1995, page 45.

[16] Renishaw plc, PH10M automated inspection system user's guide, New Mills, Wotton-under-Edge, Gloucestershire, UK.

[17] Renishaw plc, RGH22 Series Digital and Analogue Readhead data sheet, Doc. No. L-9517-0182, New Mills, Wotton-under-Edge, Gloucestershire, UK.

[18] Origin International Inc., CheckMate Training Manual, Markham, Ontario, Canada.

[19] Mitutoyo America Coporation, GEOMeasure 3000/4000/6000 Getting Started, Operation, Language, Manual. Aurora, IL, US.

[20] Consortium for Advanced Manufacturing – International (CAM – I), Dimensional Measuring Interface Standard v4.0, 2001, http://www.can-i.org.

[21] David W. Chang, Host Signal Processing Based CMM Motion Control, Master thesis, McMaster University, July 2002, page 76.

[22] Omni-Tech Corporation, OTC5000 Serial Interface Protocol, Rev: Q. hppt://www.omnicmm.com, Fenton, Mi, US, January, 2000.

[23] Jonathan Corney, Theodore Lim, 3D Modeling with ACIS, Saxe-Coburg publications, UK, 2001.

[24] Autodesk, Inc., Autodesk Inventor 5.3, http://www.autodesk.com , 2001.

[25] Biedenbach, H.-M., Brunner, J., Gläsner, K., Moritz, G., Pfeifle, J., Resch, J., I++ Dimensional Measuring Equipment Specification Version 1.30, International Association of Coordinate Measuring Machines Manufacturers (ia.cmm) c/o Ernst & Young – ATAG Association Management Ltd, PO Box 8022 Zurich – Switzerland..

[26] T. Lozano-Perez, "Spatial planning: A configuration space approach," IEEE Transactions of Computers, vol. C-32, no. 2, pp. 102–120, 1983.

[27] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots." International Journal of Robotic Research, pp. 90–98, 1986.

[28] K. Kant and S. W. Zucker, "Planning collision-free trajectories in time-varying environments. A two-level hierarchy." Proceedings of IEEE International Conference on Robotics and Automation, 1988, pp. 1644-1649.

[29] Patrick G. Xavier, "Fast swept – volume distance for robust collision detection." Proceedings of the 1997 IEEE International Conference on Robotics and Automation, April 1997, pp. 1162-1169.

[30] R.G. Beaumont, R.M. Crowder, "Real – time collision avoidance in two-armed robotic systems." Journal of Computer-Aided Engineering, December 1991, pp. 233-240.

[31] Ching- Shiow Tseng, Chen-Chun Wu, "Collision detection for multiple robot manipulators by using orthogonal neural networks." Journal of Robotic Systems, 12(7), 1995, pp. 479-490.

[32] Ming C. Lin, Stefan Gottschalk, "Collision detection between geometric models: a survey." IMA Conference on Mathematics of Surfaces, San Diego, CA, vol. 1, May 1998, pp. 602-608.

[33] Ming C. Lin, Efficient Collision Detection for Animation and Robotics, Ph.D. dissertation, University of California at Berkelay, 1993.

[34] Ming C. Lin, Dinesh Manocha, Jon Cohen, Stefan Gottschalk, "Collision detection: Algorithms and Applications." Proceedings of Algorithms for Robotics Motion and Manipulation, pp. 129-142.

[35] A.D. Spence, F. Abrari, M.A. Elbestawi, "Integrated solid modeller based solutions for machining." Computer-Aided Design, 32 (2000), pp. 553-568.

[36] Intelligent Machines and Manufacturing Research Centre (IMMRC), Scheme References, McMaster University, Faculty of Engineering.

[37] R. Kent Dybvig, The Scheme Programming Language, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[38] William Stallings, "IPv6: the new internet protocol." IEEE Communications Magazine, July 1996, pp. 96-108.

[39] C. Barakat, E. Altman, W. Dabbous, INRIA, "On TCP performance in a heterogeneous network: A survey." IEEE Communications Magazine, January 2000, pp. 40-46.

[40] Gary Cornell, Visual Basic 6 from the Ground Up, Osborne/Mcgraw-Hill, CA, USA, 1998.

[41] Anthoy Jones, Jim Ohlund, Networking Programming for Microsoft Windows, Microsoft Press, Redmond, Washington, 1999.

[42] Mickey Williams, David Bennett, et al., Visual C++ 6 unleashed, Indianapolis, IN, Sams, 2000.

[43] Anonymous, "Smart and flexible automated inspection." Quality in Manufacturing, http://www.manufacturingcenter.com/qm/archives/0102/0102automated.asp, 2001.