UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: **REMOTE MACHINING FOR LASER CUTTING APPLICATIONS**

SESI PENGAJIAN: 2007/2008

Saya,   **RAJA MOHD SALLEH BIN RAJA BAHTIAR(861004-23-5945)**

mengaku membenarkan tesis (Sarjana Muda / ~~Sarjana / Doktor Falsafah~~)* ini disimpan di perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis ini adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

☐	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
√	**TIDAK TERHAD**	

Disahkan oleh:

_____   _____
(TANDATANGAN PENULIS)   (TANDATANGAN PENYELIA)

Alamat Tetap:

3A-06-05, Desaku 3, Jalan Melati Indah 2,   **PROF. MADYA DR WAN AZHAR**
Kemensah Heights, Taman Melati Indah,   **WAN YUSOFF**
53100 Kuala Lumpur

Tarikh: 06 NOVEMBER 2008   Tarikh: 06 NOVEMBER 2008

CATATAN: *   Potong yang tidak berkenaan.

**   Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

◆   Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

REMOTE MACHINING FOR LASER CUTTING APPLICATIONS

RAJA MOHD SALLEH BIN RAJA BAHTIAR

A report submitted in partial fulfilment of the requirements
for the award of the degree of
Bachelor of Mechanical Engineering

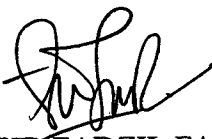Faculty of Mechanical Engineering
UNIVERSITI MALAYSIA PAHANG

NOVEMBER 2008

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project and in our opinion this project is satisfactory in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering.

Signature   :

Name of Supervisor : ASSOC. PROF. DR WAN AZHAR BIN WAN YUSOFF

Position   : DEPUTY DEAN OF FACULTY OF MANUFACTURING
                          ENGINEERING AND TECHNOLOGY MANAGEMENT

Date   : 6 NOVEMBER 2008

Signature   :

Name of Panel   : MOHD FADZIL FAISAE AB. RASHID

Position   : LECTURER

Date   : 6 NOVEMBER 2008

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Name   : RAJA MOHD SALLEH BIN RAJA BAHTIAR
ID Number   : MA05011
Date   : 6 NOVEMBER 2008

I dedicate this project to my loving mother, Aishah Lin bt Abdullah and my late father, Raja Bahtiar bin Raja Safran

"Isn't it astonishing that all these secrets have been preserved for so many years just so we could discover them!"

Orville Wright

ACKNOWLEDGMENTS

ABSTRACT

Today CNC machines are essential for manufacturing companies. CNC machines increases production rate. CNC machines also decreases labour cost and energy consumption. However, CNC machines still have disadvantages. CNC machines today lack the ability of being able to operate whenever, wherever. That disadvantage limits production capabilities. To solve that problem, remote machining application was engineered. Remote machining is a process where a machine operator is able to operate the machine through teleoperation. Remote machine enables CNC machines to operate whenever wherever as long there is a local computer, the remote machine program and internet connection. The remote machine of this project consists of a server and a client program for control and a web camera for remote monitoring. This project is to apply remote machining on FKM's laser cutting machine for machine upgrading and utilizing technology.

ABSTRAK

Mesin 'CNC' merupakan satu kepentingan kepada industri pembuatan pada masa kini. Penggunaan mesin 'CNC' meningkatkan kadar produktiviti. Mesin 'CNC' juga mengurangkan penggunaan tenaga buruh dan juga pembaziran tenaga. Walaubagaimanapun, mesin 'CNC' ada kekurangannya. Mesin 'CNC' tiada kebolehan untuk beroperasi di mana-mana pada bebila masa. Ketiadaan kebolehan itu menghadkan keupayaan produktiviti. Untuk menyelesaikan kekurangan itu, aplikasi 'remote machining' direkakan. 'Remote machining' merupakan suatu aplikasi di mana pengguna mesin boleh mengawal mesin di mana-mana sahaja pada bebila masa melalui 'teleoperation'. 'Remote machine' membolehkan mesin CNC digunakan pada bebila masa di mana-mana dengan adanya komputer di pengguna, komputer server (mesin), dan talian internet. 'Remote machine' projek ini terdiri daripada sebuah program server dan sebuah program pengguna untuk operasi pengawalan dan juga sistem camera internet untuk pengawasan mesin dari jauh. Projek ini adalah untuk mengaplikasikan 'remote machining' ke atas mesin laser FKM bagi tujuan mengoptimumkan mesin dan pengaplikasian teknologi.

TABLE OF CONTENTS

Page

CHAPTER 1   INTRODUCTION

CHAPTER 2   LITERATURE REVIEW

CHAPTER 3   METHODOLOGY

CHAPTER 4   RESULTS AND DISCUSSIONS

CHAPTER 5 CONCLUSION

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT MOTIVATION

The definition of manufacturing is the application of tools to the transformation of raw materials into finished products (Kalpakjian & Schmid, 2003). Another definition is to make or process (a product), especially with the use of industrial machines (Kalpakjian & Schmid, 2003).

Manufacturing exists to make use of the existing raw materials (Kalpakjian & Schmid, 2003). There are methods and machines used to manufacture. The methods and machines are to be the solution for effective manufacturing. Even though the solution has already been found, the manufacturing arena still faces problems.

The first main problem that occurs is the problem of manufacturing time. Companies look for methods and machines to decrease manufacturing time. The increase of manufacturing time increases the rate of profit loss. Manufacturing time too becomes a liability for satisfying customer needs. For production companies, the lesser the manufacturing time the better.

The second main problem is the number of products produced. Companies want big production amounts to be profitable. Producing small amounts at a time consumes time. Small production rate too increases profit loss. Machines or methods that produce small amounts increases labour cost and energy consumption.

The two main problems cannot be eliminated but it can be decreased. That is what engineers today are doing, searching for solutions to decrease or to eliminate problems that exist in the engineering world.

1.2   PROJECT BACKGROUND

Today CNC machines are essential for manufacturing companies. CNC machines increases production rate. CNC machines also decreases labour cost and energy consumption. However, CNC machines still have disadvantages.

CNC machines are only capable of being operated at its location (H.Chen et al. 2005). Operated at location means production can only be done during working hours. The disadvantage of that is time consuming. Time consumption results in increase of profit loss.

The other disadvantage of being operated at its location is that it could result in causing injuries to the operator. If there's a sudden defect to the machine, the operator might end up injured. This causes losses in labour and machine loss.

Another disadvantage of CNC machines is that it requires an operator for each machine (H.Chen et al. 2005). That requirement requires the employment of several operators. Employing more operators result in increase of labour cost.

The current CNC machines are not able to be integrated with engineering programs. The disadvantage is that quality products with fewer defects could not be produced. Another disadvantage is that quality check for products needed to be done. This results in more time consumption.

1.3   PROJECT PROBLEM STATEMENT

The main disadvantage of the CNC machine is it can only be operated at its location (H.Chen et al. 2005). This disadvantage will increase the number of disadvantages for manufacturing companies.

The machine operator can only operate the machine at operation hours (H.Chen et al. 2005). This limits the production amount of products in a day. Another disadvantage is that if there is a need to operate outside operational hours, traveling claims by operators will arise. There are two disadvantages raised here, limited production and increase in labour cost.

Operating machines at its location too may cause injury to the operator which is a liability. This looses the operator and may lose the machine too. For example, the operator slips while the machine is operating. The injured operator accidentally presses a button. The machine malfunctions. This results in labour loss and equipment loss.

This project is to create a remote machine addition for the CNC machine. The remote machine will eliminate the disadvantages of the CNC machine.

1.4   PROJECT OBJECTIVES

The objectives of this project are as follows:

- Apply control transfer protocol between remote computers with local computer.
- Build a user interface program.
- Apply IP camera system
- Assemble and test of the remote machine program on FKM's laser cutting machine

1.5   PROJECT SCOPE

The scopes of this project are:

- Prototype remote machine (will not be commercialized)
- To build simple interface between remote machine and local computer
- Will only be applied to FKM laser cutting machine
- Laser cutting parameters optimization will not be done
- Other modifications of laser cutting machine will not be done
  (Example: material placement on machine, material sorting, ECT)

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, a general discussion on remote machining and issues concerning remote machining is provided. Next, the chapter will focus on the advantages of remote machining. Five case studies are discussed later in this chapter. These provide a more concrete investigation of the above issues in the context of practical examples. The case study involves a milling machine and a turning machine.

2.2 REMOTE MACHINING

Remote machining is a process where a machine operator is able to operate the machine through teleoperation (M. Wang et al. 2005). Teleoperation is defined as a system of machine operating through a communication link, a direct control for a remote machine (A.J. A´lvares et al. 2006). This system provides a great ability of interaction between man and machine (A.J. A´lvares et al. 2006).

Teleoperation can be classified as follows (A.J. A´lvares et al. 2006):

- Manual control without computation aid
- Manual control with significant computational aid
- Supervisory control using the control accomplished by the computer
- Complete automatic control

Remote machine operates by a system whereby the operator controls and supervises a system behind a console and a man-machine interface (A.J. A´lvares et al. 2006). The operator controls the console by analyzing the visual feedback given by the console. The main element of this man-machine interface is the integration of all available information for the system operation. This includes video images, virtual models and graphic interface controls.

The objective of remote machining is to provide a realistic and efficient supervisory control (A.J. A´lvares et al. 2006).This control allows:

- Operation monitoring through video displays or virtual environment.
- Application control intervention and objective modification.

2.3   ADVANTAGES OF REMOTE MACHINING

Quality product manufacturing requires dynamic control and real time monitoring capabilities (L. Wang et. al. 2004). This too is required to fit the rapid changes in production capability and functionality.

The existence of remote machine improves the quality of a manufacturing system (S. Kumar et. al. 2007). Operators of the remote machine are able to supervise the quality of the manufactured product during operations by integrating the remote machine program with quality control software (L. Wang et. al. 2004). With this quality of products are well assured are at the highest level available to be produced.

Remote machine is a essential addition to the rapid rise the need of CNC controllers and "plug-n-play" smart controllers for machining operations (L. Wang et. al. 2004). Remote machining gives an advantage of communication within operating location(.J. A´lvares et al. 2006, L. Wang et al. 2004). Engineers are able share production data while supervising operations. This is an advantage for reducing time consumption and increasing production rate.

Remote machining system can be applied through the internet (G. Nikolakopoulos et al. 2008, A.J. A´lvares et al. 2006). This gives an advantage of saving time, cost and production rate due to the remote machine's low cost requirement, low data requirement and fast data transmission.

2.4   METHODS OF APPLYING REMOTE MACHINING

Remote machining is a control system applied through the internet (G. Nikolakopoulos et al. 2008, A.J. A´lvares et al. 2006). The utilization of the internet gives an advantage of managing a low cost operation. The internet too enables graphic environment creation (A.J. A´lvares et al. 2006).

The services available through the net are as follows (A.J. A´lvares et al. 2006):

- File Transfer Protocol (FTP)

    - This is a system where files or data is uploaded and downloaded through the net (H.M Deitel et al. 2003).
    - Files and data needed for machine operation can be sent through the internet at desired time and location without hassle (A.J. A´lvares et al. 2006).
    - Example of existing FTP software are Skype™ and Yahoo! Messenger
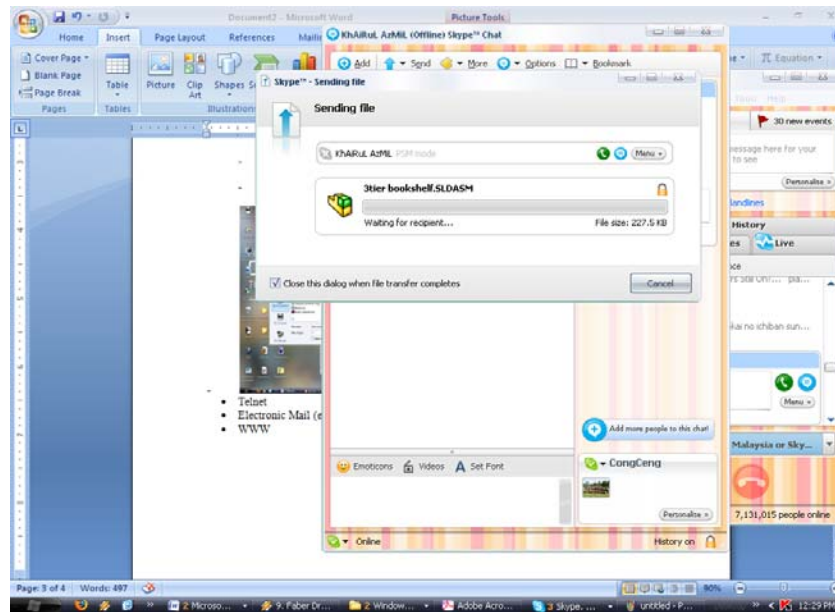
Figure 2.1: File Transfer Protocol system applied by using Skype™

- Internet Telephone (Telnet)

  - Telnet is a system whereby operators use to call through the internet (H.M Deitel et. al. 2003)
  - Operators of a machine are able to communicate through net calls which save cost and time.

- Electronic Mail (e-mail)

  - E-mail is defined as sending and receiving files and messages method through the internet (H.M Deitel et. al. 2003).
  - Files can be transferred by attaching the file in the mail intended to send.
  - Remote machine or the teleoperation process requires file transferring aids and e-mail can be used to transfer files.

- World Wide Web (WWW)

  - The teleoperation system can be transmitted through the WWW system (A.J. A´lvares et al. 2006).
  - Clients or operators can operate by typing a command in the Common Gateway Interface (CGI).
  - It does not necessarily need a webpage or server to operate, a simple operation program is already enough for operation.

For teleoperation, a basic program or server is already enough for an efficient operation. The machine can already be operated by operating the machine through a program and files needed to for processing sent through the internet.

2.5   PREVIOUS PROJECTS OR RESEARCHES

2.5.1   REMOTE MACHINING APPLIED TO MILLING MACHINE
(L. Wang et. al., 2004)

This research was to develop an appropriate methodology with open architecture for real-time monitoring and networked CNC machining. The remote machining system was applied by using a framework called Wise-ShopFloor (Web-based-integrated sensor driven e-ShopFloor) and designed based on VCM (View-Control-Model).The remote machining system was experimented on a milling machine.

The Wise-ShopFloor was designed to provide users with a web-based and sensor-driven manufacturing environment. Real-time monitoring and remote control are the cores of this framework. The framework utilizes Java 3D and Java servlets which enables system implementation. The images appear on screen is a 3D model due to the need of quick data transfer. The Wise-ShopFloor framework too provides the web-ready camera incase if there's any need for real scenes analysis.
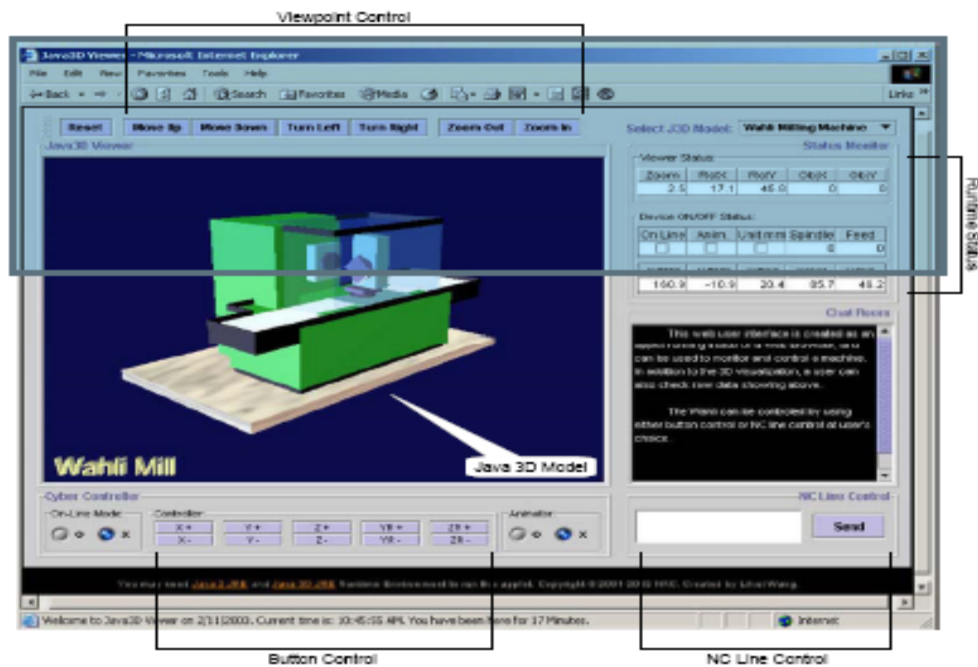
Figure 2.3: 3D model featured by Wise-ShopFloor

Source: L. Wang et al. (2004)

The Wise-ShopFloor framework 3D model is not that efficient as in the 3D model captured from the real scene does not really feature quality images. The disadvantage of the framework is that the operator could not really analyze defects that occur during manufacturing process. It requires better 3D visualization aids in order to gain higher quality images.

The advantage of the framework is that by using the 3D model, test experiments using mechanical properties test software can be applied according to 3D model received from the framework.

2.5.2   TELEOPERATION BY USING CNC TURNING MACHINE
(A.J. A´LVARES ET AL. 2006)

This research was done to experiment the application of WebDNC server teleoperating system on a CNC turning machine. The system runs on a system a server that can be accessed through http://WebDNC.graco.unb.br based on a client-user server. The system is composed based on two modules:

- o   There are three servers. One server captures images by using a Web-camera during real-time operation. The other server runs a personal program installed to a personal computer to store and upload data for the turning machine to operate. This server is linked to another server that is run on a Linux Platform which is connected to the CNC machine.
- o   The client is able to operate through Java Applets and HTML pages.

The server implements a simple FTP system for data uploading and downloading. The slow data transmission and inherent delays creates problems for smooth teleoperation. The system requires a program that is able to transmit data smoothly and at a fast rate. The program too must have a security system whereby the data transmitted is transmitted safely without information being leaked out. Figure 2.4 shows the sample of WebDNC server teleoperation system.
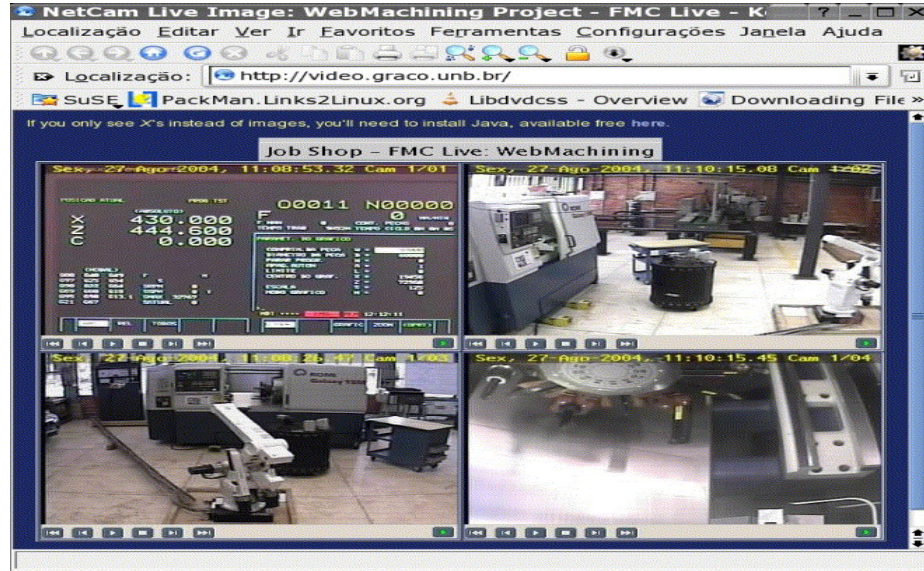
Figure 2.4: WebDNC Server

Source: A.J. A´lvares et al. (2006)
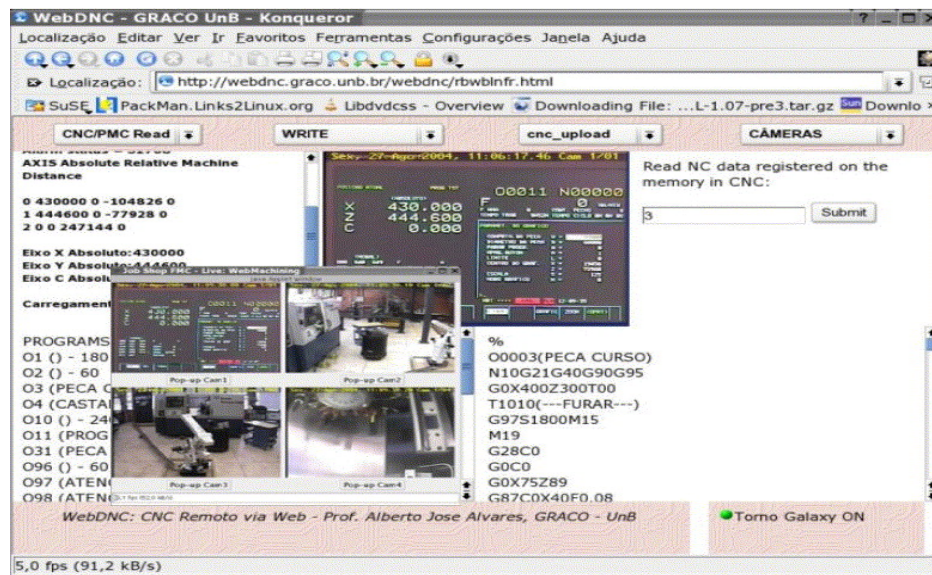
Figure 2.4: Continue

2.5.3   TELEOPERATION FOR MANURE SPREADER (N. MURAKAMI, 2008)

Remote machining was applied to a manure spreader in Japan. The manure spreader is controlled using remote machine software that utilizes Google® Earth for machine navigation. The operation location and operating route is determined through the map provided by Google® Earth. After obtaining the location and operating route, the information is sent to a simple GPS system for the machine to operate on. This is similar to the auto drive ability of certain modern cars that utilizes the GPS system.

The viewing of the operator was done by applying a hyper-omni camera to the machine. The camera is able to zoom four times but is only one dimensional. The camera applied is not able to rotate on other angles. This results in the disability to capture or view images on other angles which may cause operating problems.
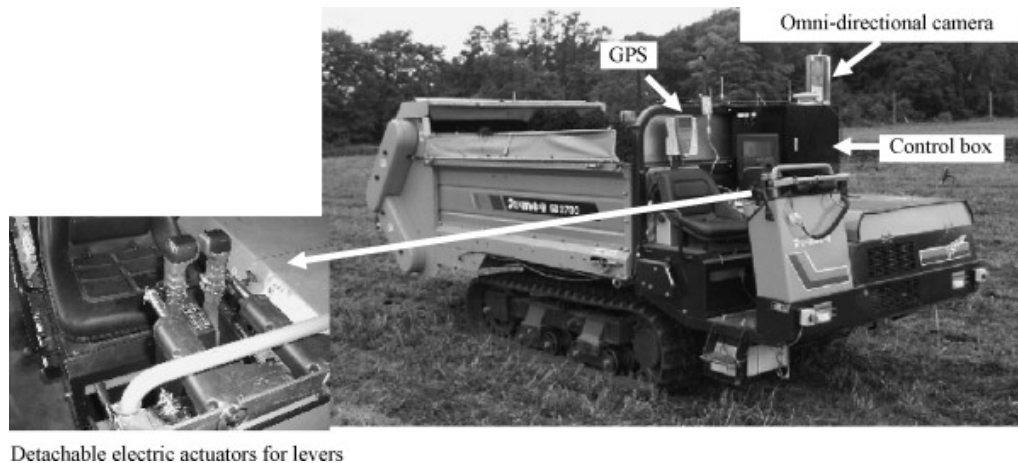
Figure 2.5: Manure Spreading Machine

Source: N. Murakami (2008)
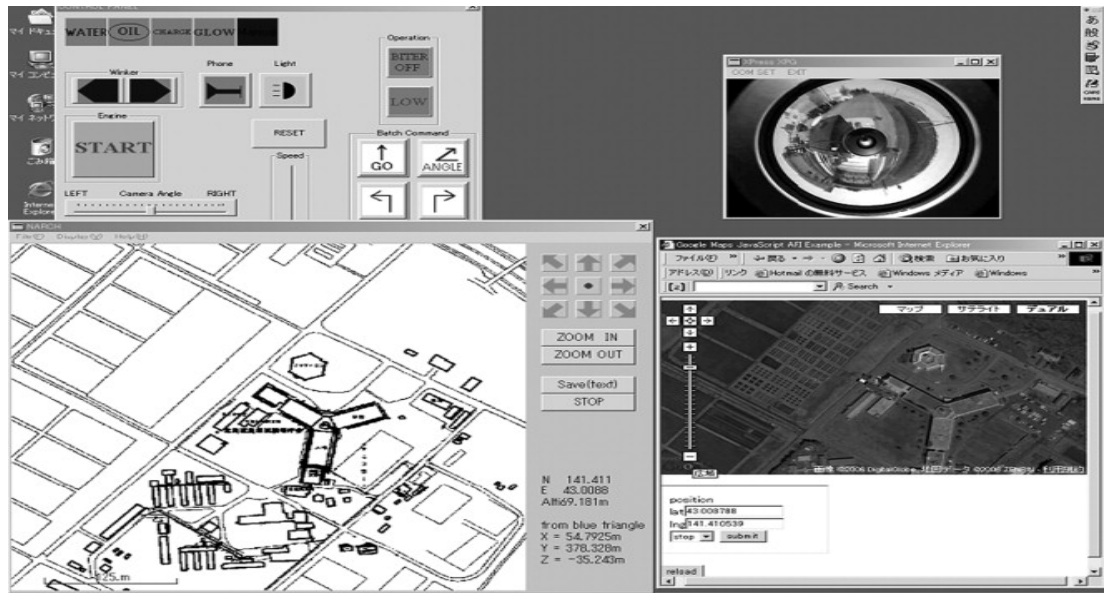
Figure 2.6: Remote machine software and GPS navigation system

N. Murakami (2008)

2.5.4   REMOTE MACHINE FOR FINITE ELEMENT ANALYSIS (H.CHEN ET AL. 2005)

This remote machine software is applied using the www system. Clients are able to run finite element analysis without installing analysis software in a computer. A client would only require logging in to the finite element website to run an analysis.

The framework of the finite element analysis server implements C++ and Java. Java was applied to enable 3D images construction. C++ was applied to enable analysis work. The creation of a finite element analysis server enables fast analysis work. It too eliminates the requirement of upgrading an average local computer to a high performance computer in order to cope with requirements to run the analysis software.
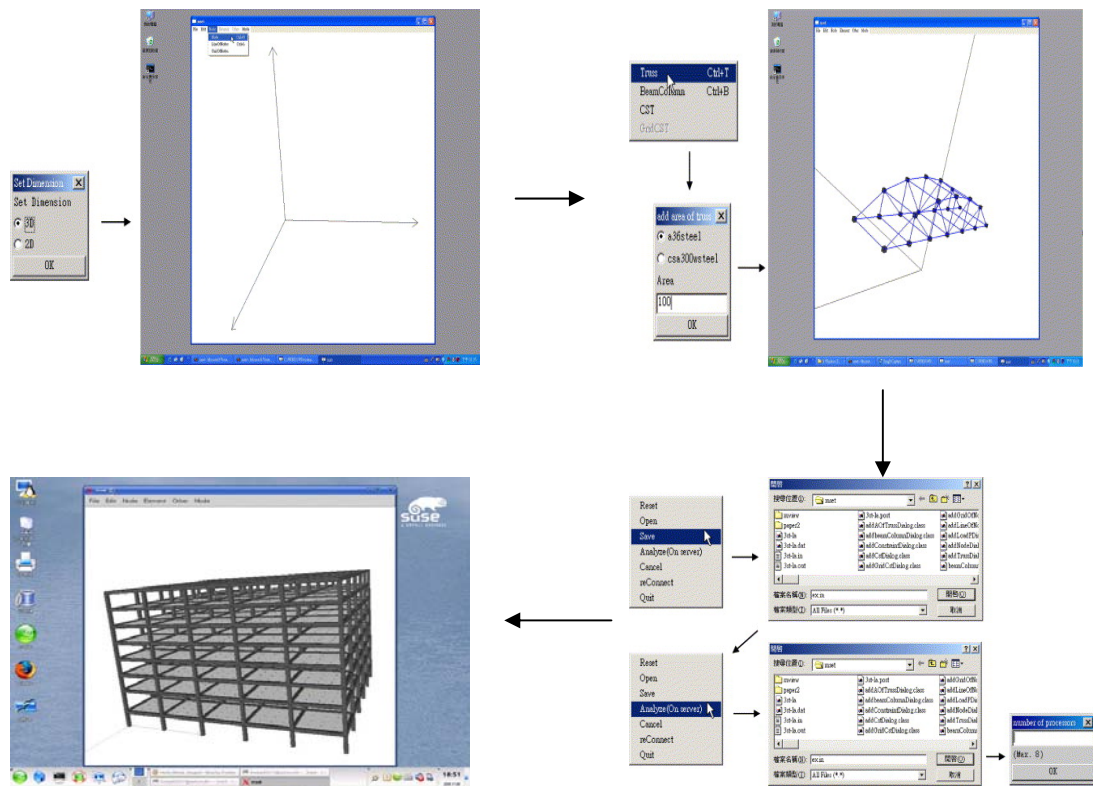
Figure 2.7: Finite Element Analysis done through a Website

Source: H. Chen et al. (2005)

2.5.5 REMOTE MACHINE ONLINE MONITORING SYSTEM (M-CH. PAN ET AL. 2004)

An online monitoring system including data acquisition, signal-processing, feature extraction, pattern recognition through the artificial neural networks, and online video surveillance was created. The monitoring system was created for machine condition monitoring.

This remote system is implemented using the Socket of the TCP/IP as Internet communication interface, and coded with the use of BCB programming language. A particular feature of the system is the in-house coded analysis-tool modules at the client end, which offers agile signal-processing schemes for feature extraction, and a BP algorithm of ANNs for further decision making. A complete monitoring cycle is demonstrated as an example to evaluate the effectiveness of the developed system.

The remote machine system gives an advantage of employing one operator to monitor the whole machine system. This advantage decreases labor cost.
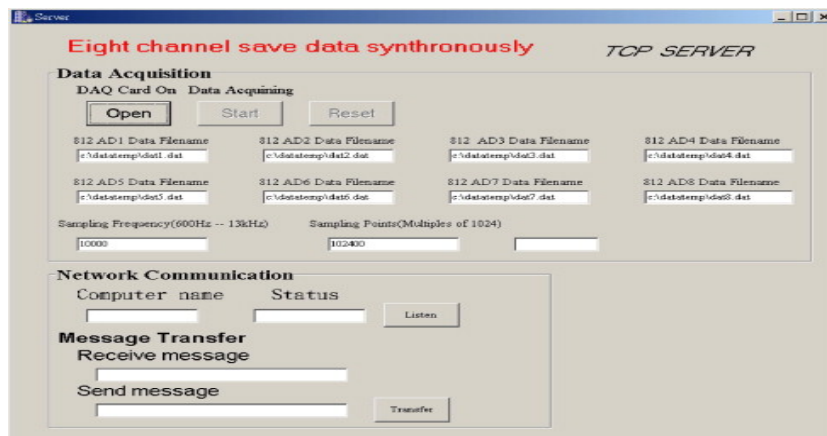
Figure 2.8: Sample monitoring system

Source: M-Ch. Pan et al. 2004
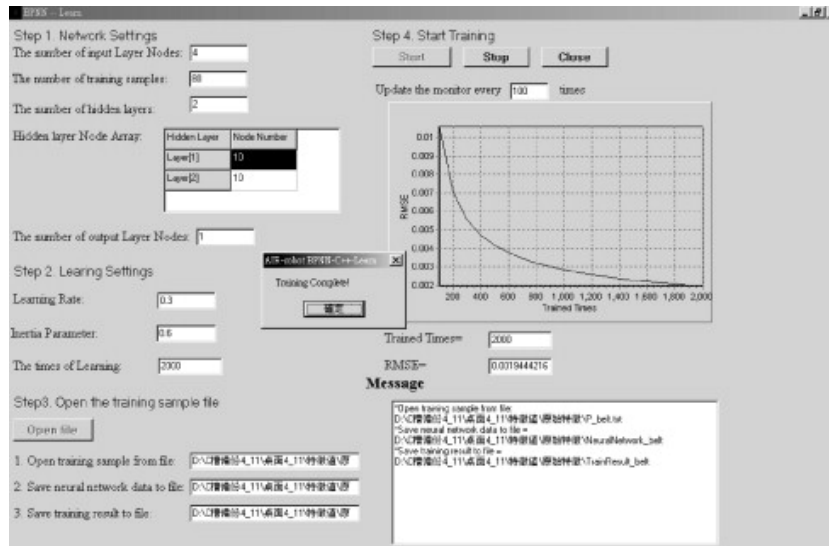
Figure 8: Continue

2.6   SUMMARY

This chapter gives a general introduction to remote machining and the advantages of having a remote machining system in a CNC machine. The internet is a utility tool for remote machining as there are various ways of applying the remote machine system. Real time images are best to be used for quality control analysis.

CHAPTER 3

METHODOLOGY

3.1   INTRODUCTION

This chapter focuses on the methods that were used to realize this project. Methods of applying the remote machine and design of remote machine were discussed. The design applied for the project is specified according to the project objectives. Issues that have been discussed in the literature review are observed to select the most suitable method for project application.

3.2   CONCEPTUAL STUDY

Conceptual study is important for understanding of remote machining concept. Concept study is crucial before initiating a project. The understanding of how remote machine works is important for the designing of new remote machine system.

3.3   CONCEPTUAL DESIGN

Conceptual design was done to create a solid foundation to build a successful user interface. A simple interface design is represented in the interface in order for the user to be able to learn and use the program. The design is to determine how the system of remote machine works. The remote machine was designed based on three key concepts:

1. control transfer protocol between remote computers with local computer
2. user interface program
3. IP camera system

3.3.1   CONTROL TRANSFER PROTOCOL

The control transfer protocol is used to transfer data from computer to computer. Commands are sent through the internet using strings which will be converted into integers or characters once when received by either server or client.

3.3.2   USER INTERFACE PROGRAM

The interface program must be designed based on 2 criteria. The criteria are:

I.   Applicable between operator and machine
II. Methods of operating must be easy to be learned by operator

3.3.3 IP CAMERA SYSTEM

The web camera is used for observation of the machining process from the operator's computer screen. By using a web camera, the operator is able to navigate the machine to a desired position.

3.4 OVERALL METHODOLOGY PROCESS
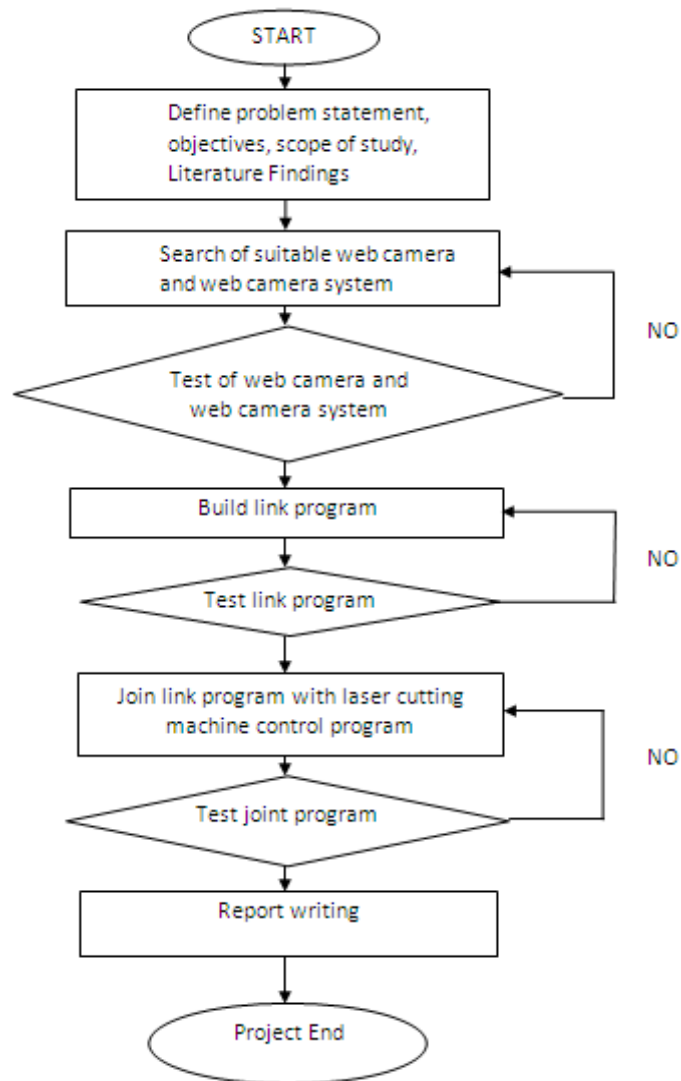
Figure 3.1: Overall project flowchart

The project started by defining the disadvantage of the CNC machine being not able to be operated anywhere, anytime. After defining the problem statement, objectives of the project were determined. Once objectives were determined, literature findings from available journals were carried out. The findings consisted of the definition of remote machining, advantages of remote machining, and methods of remote machine application.

A search of a suitable web camera and a web viewing camera system was done after the facts and figures of related projects were certified in the literature findings process.

3.5 WEB CAMERA AND CAMERA SYSTEM

The web camera used for the viewing of the remote machine is D-Link DCS – 5220. The camera's system allows a live video feed on a compatible 3G mobile phone or PDA.

Figure 3.2: DCS-5220

Source: www.dlink.com

3.5.1   WEB CAMERA ABILITY

This full-featured camera boasts motorized pan and tilt capabilities that allow a greater coverage area with a single camera. Using the pan and tilt feature along with 4x digital zoom, users can zero-in on a subject much more effectively.

3.5.2   REMOTE MONITORING

The remote monitoring function can viewed at pcnc.ump.edu.my. From the site, the user is able to view the machining process using the pan, tilt and zoom functions.

3.6   REMOTE MACHINING PROGRAM

The programming language used for this program is C++. The application was created into a windows console. The compilation of the codes was done using a GNU Compiler through CodeBlocks code compiler. The program consists of 3 parts:

a) Remote server
b) Remote client
c) Machine control program – jogging

Remote machine functions by linking a server (the computer that controls the machine) to a client program. The remote controls the movements of the laser machine's head. The linking is done by using linker function, Winsock 2.2. A library file, libws2_32.a must be added to the compiler's library in order for a successful code compilation. **Figure 3.3** describes the flow of the program designed.

```
                          ┌─────────────┐
                          │    START    │
                          └─────────────┘
                                 │
                                 ▼
        ┌──────────────────────────────────────────┐
        │              Server start                 │◄──────────┐
        └──────────────────────────────────────────┘           │
                                 │                              │
                                 ▼                          NO  │
                  ╱───────────────────────────╲                │
                 ╱     Server links with client ╲───────────────┤
                 ╲       of same version or     ╱                │
                  ╲───────────────────────────╱                 │
                                 │                          NO  │
                                 ▼                              │
                  ╱───────────────────────────╲                │
                 ╱     Server links with client ╲───────────────┘
                 ╲       of same IP number      ╱
                  ╲───────────────────────────╱
                                 │
                                 ▼
        ┌──────────────────────────────────────────┐
   ┌───►│          Server request for client's      │◄──────────┐
   │    └──────────────────────────────────────────┘           │
   │                             │                              │
   │                             ▼                     Invalid  │
   │              ╱───────────────────────────╲        command  │
   │             ╱        Client states         ╲───────────────┘
   │             ╲           option             ╱
   │              ╲───────────────────────────╱
   │                             │
   │                             ▼
   │    ┌──────────────────────────────────────────┐
   │    │         Server links with jogging         │
   │    └──────────────────────────────────────────┘
   │                             │
   │                             ▼
   │    ┌──────────────────────────────────────────┐
   └────│              Machine moves                 │
        └──────────────────────────────────────────┘
   Loop
```
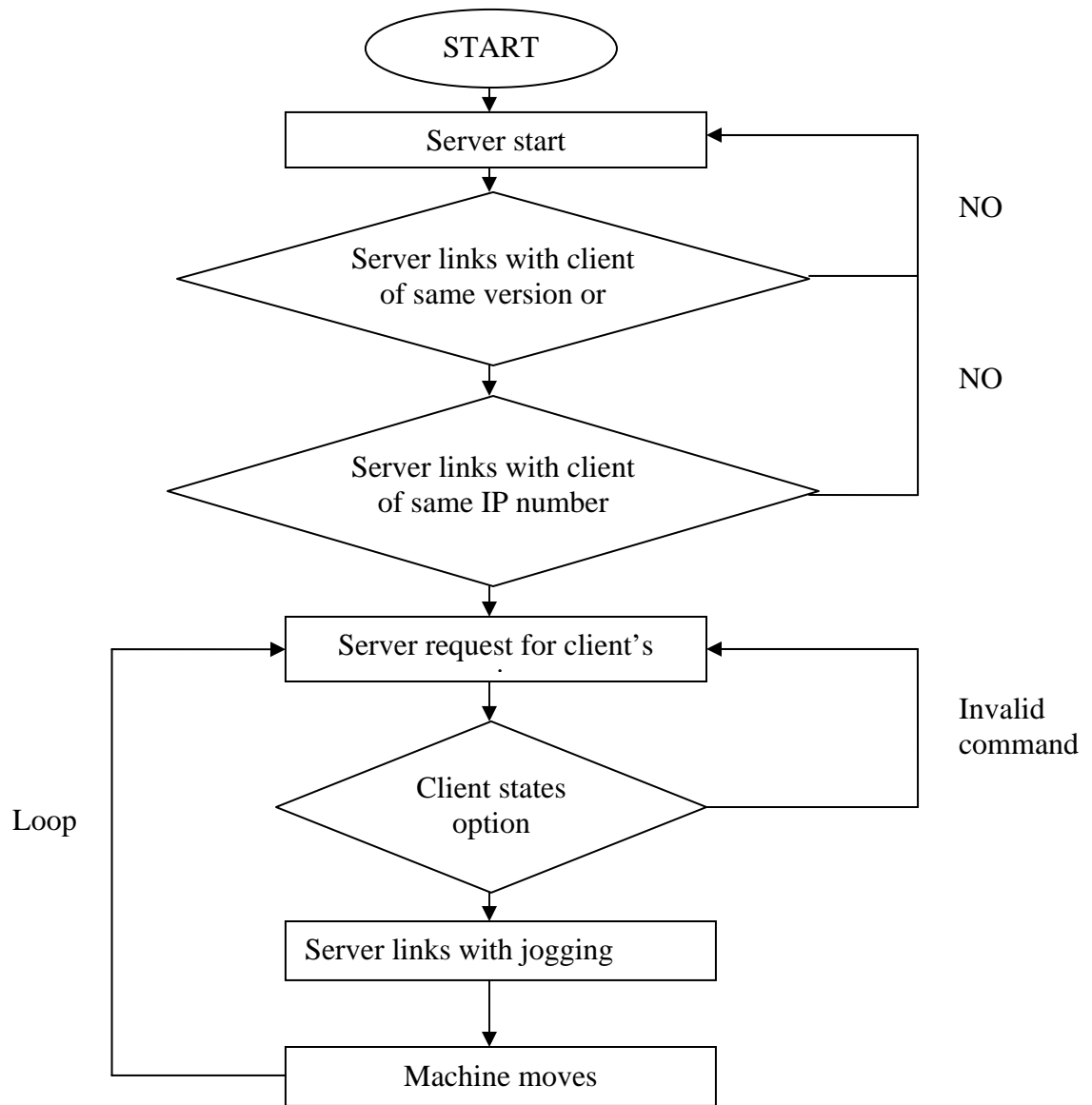
Figure 3.3: Program process flowchart

3.6.1   REMOTE SERVER

The server was created as a service provider for clients. In this case, the service is the laser cutting function of the machine. The steps in creating a server were as shown in Figure 3.3.
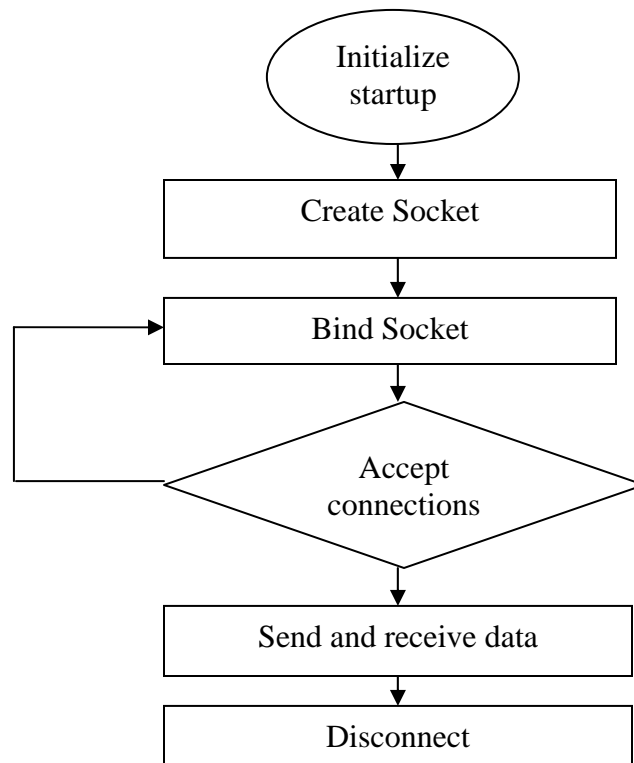
```
        ┌─────────────────┐
        │    Initialize   │
        │     startup     │
        └─────────────────┘
                 │
                 ▼
        ┌─────────────────┐
        │  Create Socket  │
        └─────────────────┘
                 │
                 ▼
        ┌─────────────────┐
     ┌─▶│   Bind Socket   │
     │  └─────────────────┘
     │           │
     │           ▼
     │       ◇ Accept
     └───── connections ◇
                 │
                 ▼
        ┌─────────────────┐
        │Send and receive │
        │      data       │
        └─────────────────┘
                 │
                 ▼
        ┌─────────────────┐
        │   Disconnect    │
        └─────────────────┘
```

Figure 3.4: Creating a server

The integral part of the server is the version of the server and the internet protocol address of the server. Both parts are the main conditions where the client program has to meet in order for a successful connection

3.6.2   REMOTE CLIENT

      The client requests for the server to provide service. Server and client both have different characteristics. Server waits and provides while client connects and requests. The flow of the client program is as shown in **Figure 3.4**
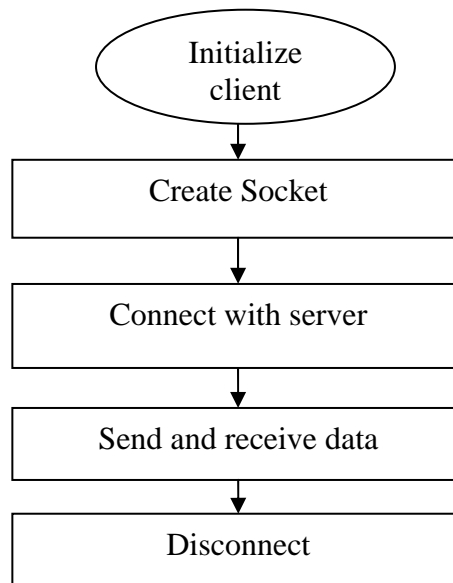
```
        ╭─────────────╮
        │  Initialize │
        │    client   │
        ╰──────┬──────╯
               ▼
   ┌──────────────────────┐
   │     Create Socket    │
   └──────────┬───────────┘
              ▼
   ┌──────────────────────┐
   │   Connect with server│
   └──────────┬───────────┘
              ▼
   ┌──────────────────────┐
   │  Send and receive data│
   └──────────┬───────────┘
              ▼
   ┌──────────────────────┐
   │      Disconnect      │
   └──────────────────────┘
```

Figure 3.5: Flow of client program

The client has to be the same version with the server or a lower version. The IP address of the server must be keyed in to the client's program to enable a connection.

3.6.3 MACHINE CONTROL

The control program applied in this project was the basic machine head control program (Jogging). The program controls the movements of the machine head, where the machine head will be moved and positioned to an operator's desire.

The program was created in a header function to simplify the source code structure. The creation of the header function was also to simplify the analysis process whenever error occurs.

Integers 1~5 were used for the command to control the machine. The integers represented the following commands:

- 1 : Forward
- 2 : Backward
- 3 : Left
- 4 : Right
- 5 : Quit program

3.7   TESTING OF REMOTE MACHINE AND CAMERA SYSTEM

The final stage of the program creating process was to test the remote machine program and the web camera function. First the function ability of the camera was tested out then the remote machine program was tested.
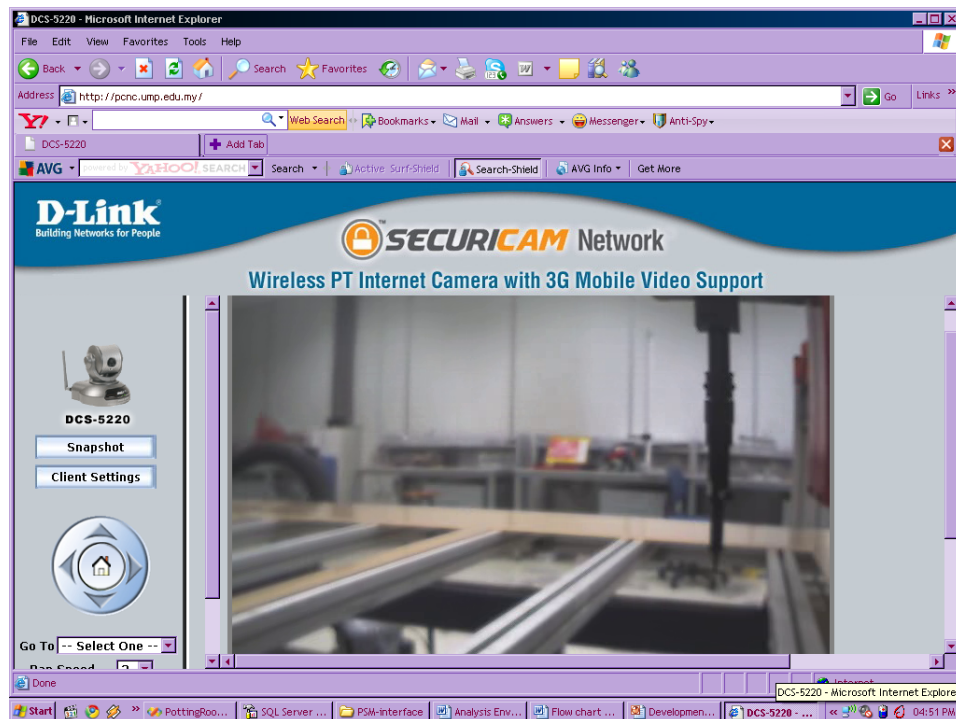

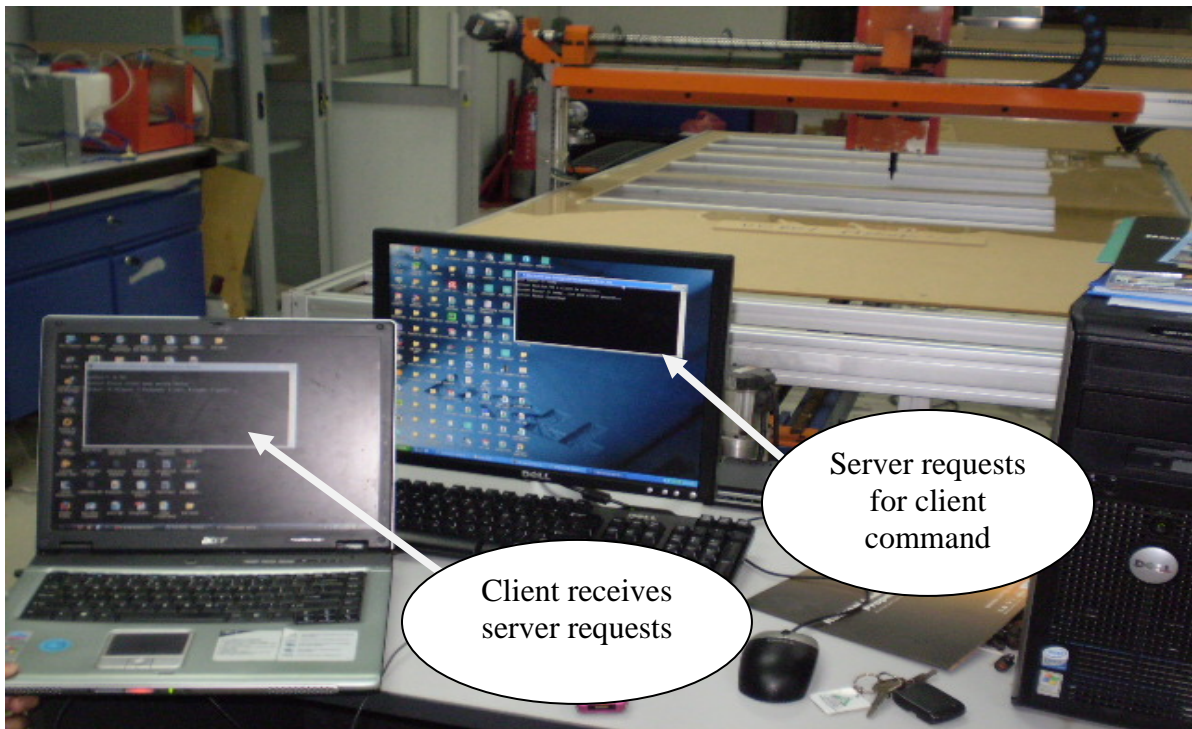**Figure 3.6:** Web camera system

Source: pcnc.ump.edu.my

Server requests for client command

Client receives server requests
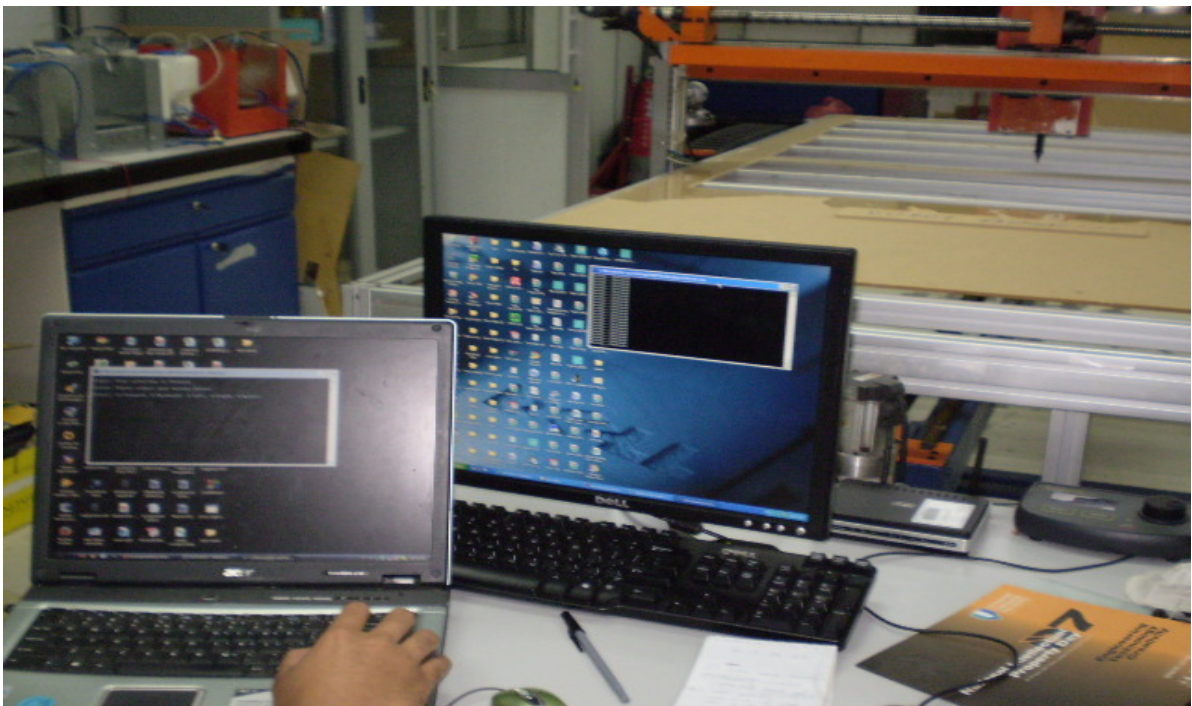
Figure 3.7: Server and Client Connects

Figure 3.8: Jogging program commenced

3.8   SUMMARY

Chapter three elaborated the information on flow of the creation process of the remote machine program. Detailed descriptions were given on each stage of the creation process until the remote machine was finally created successfully. The previous chapter gave a general introduction on remote machine and the advantages of remote machine applied on CNC machines.

CHAPTER 4

RESULT AND DISCUSSION

4.1   INTRODUCTION

This chapter discusses the problems and consideration that involves the creation process of the remote machine program. The program involves the usage of 2 computers and FKM's laser cutting machine. The topics of program compilation, linkage between server and client, data transfer and machine control will be covered in this chapter.

The creation of this program was done step by step. The purpose of step by step creation process was to enable analysis of problems that may rise during the creation process without affecting the program at whole.
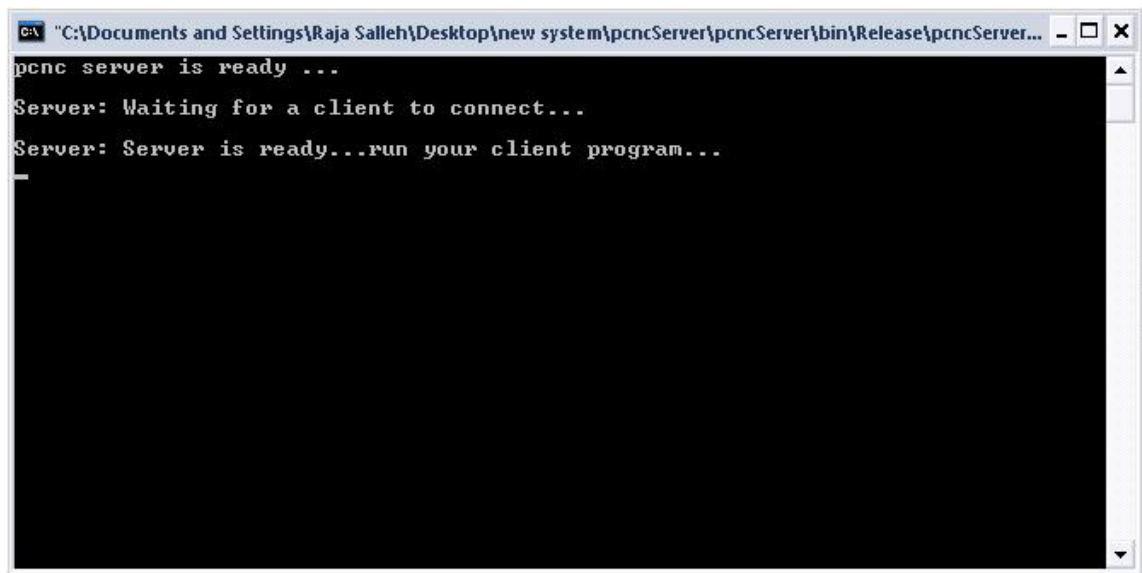
4.2   COMPILATION OF PROGRAM

A socket server system was created for the remote machine program. The system used was called Winsock 2.2. The system consisted of a server and a client program. A library file, libws2_32.a was needed to enable the compilation of both client and server program. All compilations were done using GNU compiler through compiling software called CodeBlocks.

4.3   CONNECTION BETWEEN SERVER AND CLIENT

The remote machine can only activate when the server and client program is linked and connected. Once server and client are connected the functions of the remote machine can be applied to the laser cutting machine. The server was created with a setting of two conditions to be met by the client program for a successful program connection. The two conditions are:

- The client must be the same version or lower of the server
- The client must have the same IP address of the server embedded in the program

The two conditions are to be met consecutively. If the two conditions are not met by the client the server will remain as it is until there is a successful connection with the client as shown in **Figure 4.1**.
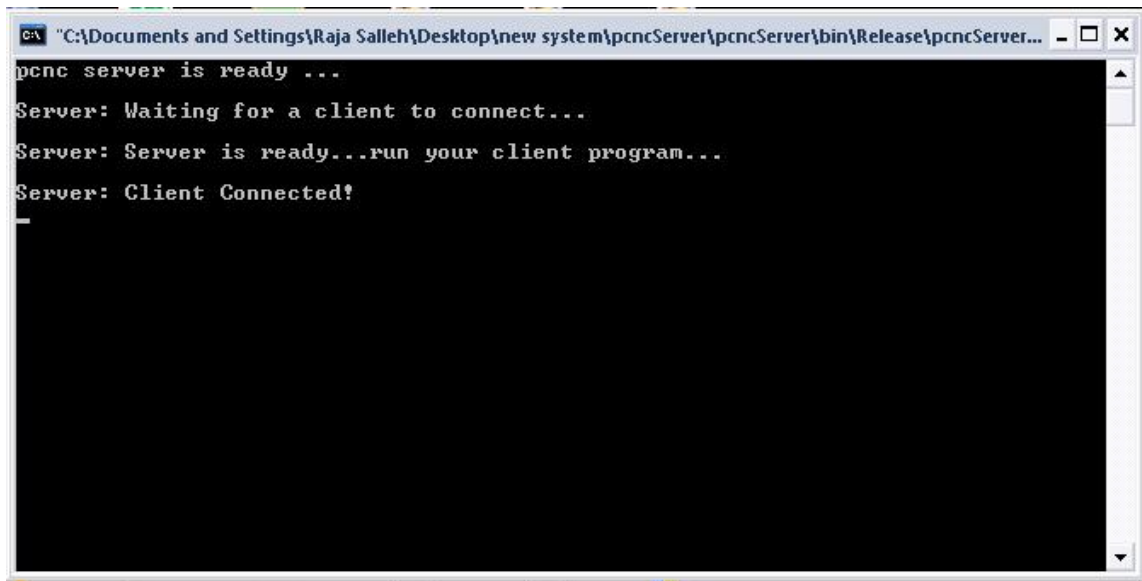
Figure 4.1: Server

4.3.1 CLIENT OF SAME VERSION

A server can connect to a client with the same version or lower. The properties of both server and client can accommodate each other which enable the connection. A higher version of client can not be connected due to different program properties.
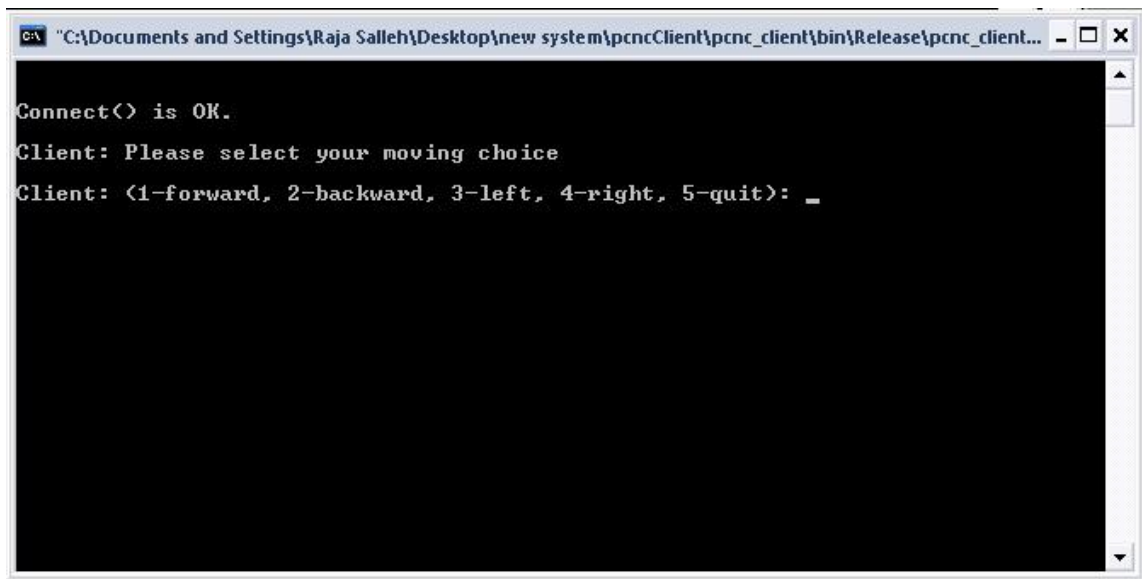
4.3.2 CLIENT OF SAME INTERNET PROTOCOL ADDRESS

After a successful bind between server and client of the same version, a request of the same IP address is issued out by the server for a successful connection. The client with the same IP address as the server only can connect with server. The example of a successful connection is shown in **Figure 4.2** and **4.3**. **Figure 4.4** shows the example of an unsuccessful connection.

Figure 4.2: Successful connection of server

Figure 4.3: Successful connection of client

Figure 4.4: Failed connections

4.4   DATA TRANSFER

The main function of the socket server is to enable data transfer from one computer to another. The data sent through the server is in string form. To enable data transfer, basic declarations of data sent and received are needed to be done.

4.4.1   CONVERTING STRING TO INTEGER

The data sent and received in strings are needed to be converted into integers to enable machine control. The integers are applied in the switch function in the program. The function of 'atoi' was applied to convert string into integer.

4.5   MACHINE CONTROL

A prototype of a remote machine has to be applied to the simplest function of the machine. In the laser cutting machine case, the 'jogging' function is the simplest. 'Jogging' is the function of moving the machine head from one point to another. The function is controlled by a program which is connected to the machine from the computer through a parallel port.

The 'jogging' program is executed by keying number 1~4 which represents forward, backward, left and right respectively. The laser cutting process can be enabled through the jogging program but it is not being activated in this project test for safety reasons.

After successful tests of socket server connection and data transfer, the 'jogging' program was included into the server program to test remote machining. The first stage of the test was carried out without switching on the machine. The intention was to test for any possible errors of the program.

4.5.1   ERROR IN PROGRAM

There was an error on the keyboard hit function of the program. The jogging program has a function whereby the machine can be stopped by a hit of the enter button of the keyboard. That function functioned when the program was run by its own but was not applicable when the program was linked into the server program

.

It was found that the command to stop the program from running did not reach the jogging program. The data of the command was transferred successfully through the net but the command only reached the server program. The command could not reach the jogging program due to the non-existence of a keyboard hit command in the client program. **Figure 4.5** and **4.6** shows the error flow of the program.

Figure 4.5: Jogging program receives command
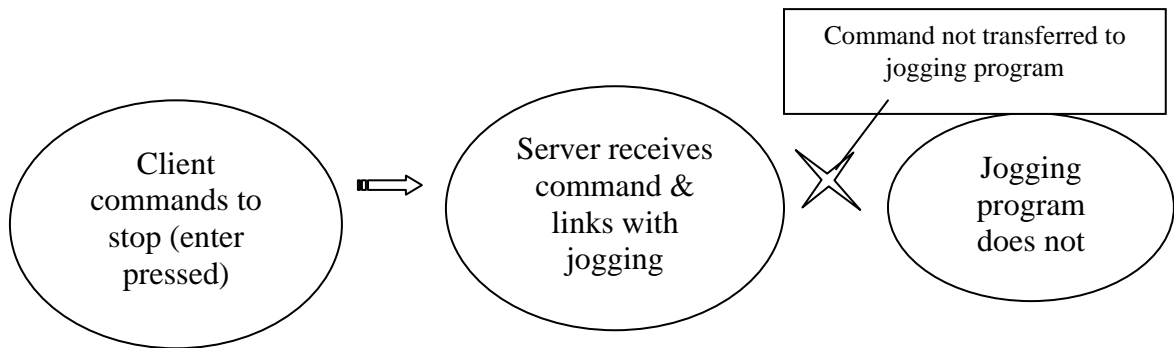
Figure 4.6: Command failed to be reach jogging

4.5.2 SOLUTION
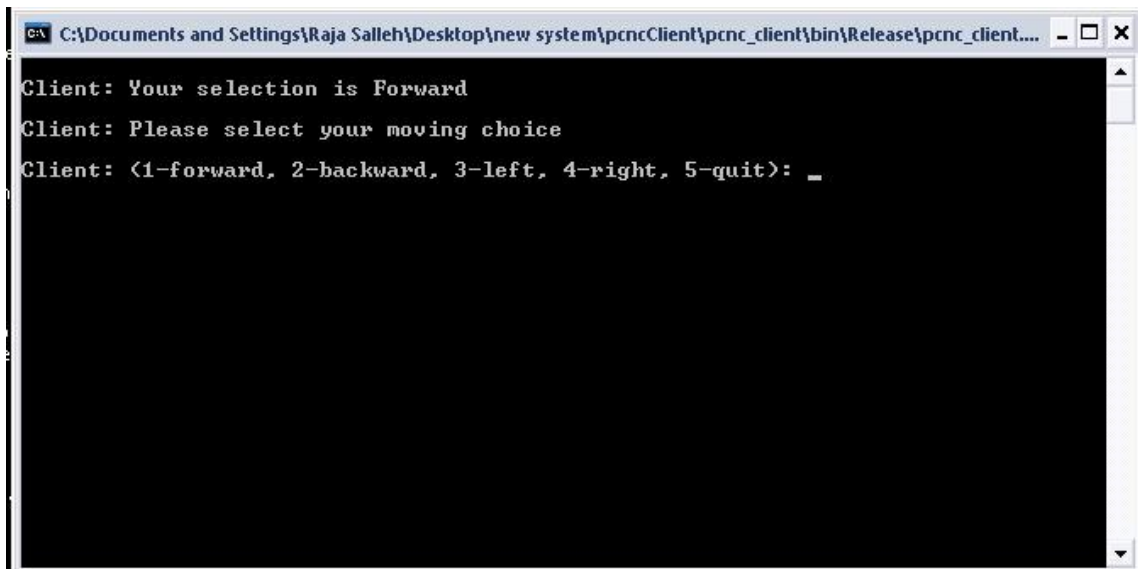
The problem was solved by setting limits to each moving function. The length of travel was limited to five centimeters per command. The setting in the source code was set to 10,000 pulses for each command. The machine was set to a speed of five cm/s. The calculations of the speed set were as follows:

$$1 \text{ pulse} = 100\mu s$$

$$10,000 \text{ pulses} = 5 \text{ cm}$$

$$1 \text{ pulse} = \frac{5 \text{ cm}}{10000} = 0.0005 \text{ cm / pulse}$$

$$\text{Machine Speed} = \frac{0.0005}{100\mu} = 5 \text{ cm/s}$$

Figure 4.7: Client connects and sends command to server

Figure 4.8 Server receives command and runs

4.6 SUMMARY

This chapter generally discusses the results obtained from the test carried out on the remote machine program. The results obtained shows that remote machine is applicable on FKM's laser cutting machine. Although successful, the remote machine has its limitations. The machine can not be stopped by using the keyboard hit function. Travelling limits had to be set on each moving function.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

There were many walls encountered through out the development of the remote machine program. One of them was the server had to be modified several times to accommodate the functions of the jogging program. The other problem was the compilation of the program where the needs of the libws2_32.a library file in the compiler's library.

This project requires knowledge in C++ programming. Lack of knowledge in this field causes lots of mishaps and collision in programming a program. The understanding of functions, declarations and program structure is the key to a successful programming. A perfect example of the need of C++ programming knowledge is the development of the remote machine project.

The main objective of this project was to create a remote machine program to be applied on FKM's laser cutting machine which the machining process can be viewed through the internet. The remote machine program was successfully created with an application of a web camera system which can be viewed through pcnc.ump.edu.my.

5.2 Recommendations

The remote machine created was a prototype. The prototype must be improved before it can be commercialized or applied on other machines. The improvements that can be applied to the remote machine are as follows:

1. Program the remote machine in Visual Basic for a simpler user interface and user friendly.
2. Solve the keyboard hit function problem. The machine must be able to operate at full potential.
3. Improve the program to be able to receive G-code files and machine materials.
4. Apply the remote machine function on other CNC machines.
5. A good study of computer programming and internet knowledge for FKM students.

REFERENCES

A. J. A´lvares, J. C. E. Ferreira, 2006, *Journal of Materials Processing Technology* 179

G. Nikolakopoulos, A. Panousopoulou, A. Tzes, 2008, *Control Engineering practice* 16

H-M. Chen, Y-C. Lin, 2005, Internet based analytical services using a Java based GUI. In: B.H.V. Topping, Editor, *Proceedings of the tenth international conference on civil, structural and environmental engineering computing (CC2005)*, Civil-Comp Press, Stirling, UK (2005) paper 14

H.M Deitel, P.J. Deitel, A.B. Goldberg, 2003, *Internet & World Wide Web How to Program*, Third Edition

L. Wang, P. Orban, A. Cunningham, S. Lang, 2004, *Robotics and Computer-Integrated Manufacturing* 20

M-Ch. Pan, Y.-H. Liu, 2004, Transmission-component monitoring and comparison of two artificial neural network schemes, in: Proceedings of SPIE Volume: 5391 *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, pp. 806–814

M.Wang, J.M.K.Liu, 2005, *Robotics and Autonomous Systems* 32

S. Kumar, A. Nassehi, 2007, S.T. Newman, R.D. Allen, M.K. Tiwari, *Robotics and Computer Integrated Manufacturing* 23

```cpp
#include <iostream.h>

#include <stdlib.h>

#include <string.h>

#include <winsock2.h>

#include "jogging.h"


void pcncGetReady(void){


WORD wVersionRequested;

WSADATA wsaData;

int wsaerr;

wVersionRequested = MAKEWORD(2, 2);

wsaerr = WSAStartup(wVersionRequested, &wsaData);

if (wsaerr != 0)

{

   cout << "Winsock dll tidak dijumpai!" << endl;

   return;

}

if (LOBYTE(wsaData.wVersion) != 2 || HIBYTE(wsaData.wVersion) != 2 ){


   cout << "dll ini tidak menyokong Winsock versi " << LOBYTE(wsaData.wVersion) << "." <<
HIBYTE(wsaData.wVersion) << endl;

   WSACleanup();

   return;

}

SOCKET m_socket;

m_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
```

```cpp
if (m_socket == INVALID_SOCKET)

{

    cout << "Error at socket(): " << WSAGetLastError() << endl;

    WSACleanup();

    return;

}

sockaddr_in service;

service.sin_family = AF_INET;

service.sin_addr.s_addr = inet_addr("127.0.0.1");

service.sin_port = htons(55555);

if (bind(m_socket, (SOCKADDR*)&service, sizeof(service)) == SOCKET_ERROR)

{

    cout << "bind() gagal: " << WSAGetLastError() << endl;

    closesocket(m_socket);

    return;

}

if (listen( m_socket, 1) == SOCKET_ERROR)

    cout << "listen(): Error listening on socket " << WSAGetLastError() << endl;

cout << "pcnc server is ready ..." << endl;


SOCKET AcceptSocket;

printf("\nServer: Waiting for a client to connect...\n" );

printf("\nServer: Server is ready...run your client program...\n");




while (1)

{

    AcceptSocket = SOCKET_ERROR;

    while (AcceptSocket == SOCKET_ERROR)
```

```
        {
            AcceptSocket = accept(m_socket, NULL, NULL);
        }
        printf("\nServer: Client Connected!\n");
        m_socket = AcceptSocket;
        break;
    }

    int arahan = 0;
    int bytesSent;
    int bytesRecv = SOCKET_ERROR;
    char sendbuf[200] = "This string is a test data from server";
    char recvbuf[200] = "";

    while ( strcmp(recvbuf, "5" ) != 0 )

    {
    bytesSent = send(m_socket, sendbuf, strlen(sendbuf), 0);
    if (bytesSent == SOCKET_ERROR)  printf("\nServer: send() error %d.\n", WSAGetLastError());
    bytesRecv = recv(m_socket, recvbuf, 200, 0);
    if (bytesRecv == SOCKET_ERROR)  printf("\nServer: recv() error %d.\n", WSAGetLastError());


    else
    {
        arahan = atoi(recvbuf);
        switch (arahan)
        {
            case 1 : {printf("Forward\n"); jogging(&arahan); break;}
            case 2 : {printf("Backward\n"); jogging(&arahan); break;}
```

```c
        case 3 : {printf("Left\n"); jogging(&arahan); break;}

        case 4 : {printf("Right\n"); jogging(&arahan); break;}

        case 5 : {printf("Quit\n"); break;}

        default : {printf ("\nInvalid command\n");}
    }
}


}


WSACleanup();

return;

}
```

APPENDIX B

(Programming of Client Program)

```c
#include <stdio.h>

#include <stdlib.h>

#include <iostream.h>

#include <winsock2.h>

int main()

{

 WSADATA wsaData;

 int iResult = WSAStartup(MAKEWORD(2,2), &wsaData);

 if (iResult != NO_ERROR)

 {

  printf("Client: Error at WSAStartup(). ");

 }

 SOCKET ConnectSocket;

 ConnectSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

 if (ConnectSocket == INVALID_SOCKET)

 {

  printf("Error at socket(): %d.\n\n", WSAGetLastError());

  WSACleanup();

  return 0;

 }
```

```cpp
sockaddr_in clientService;

clientService.sin_family = AF_INET;

clientService.sin_addr.s_addr = inet_addr("127.0.0.1");

clientService.sin_port = htons(55555);


if      (connect(ConnectSocket,      (SOCKADDR*)&clientService,      sizeof(clientService))      ==
SOCKET_ERROR)

{

printf("\n\nFailed to connect.\n\n");


WSACleanup();


return 0;

}


else


printf("\n\nConnect() is OK.\n\n");

int bytesSent;

int bytesRecv = SOCKET_ERROR;

int selection = 0;

char message[100] = "";
```

```cpp
while (selection != 5)

{

printf("Client: Please select your moving choice\n\n");

printf("Client: (1-forward, 2-backward, 3-left, 4-right, 5-quit): ");

cin >> selection;

if (selection == 1)

{

    system("cls");

                char message[100] = "1";

                bytesSent = send(ConnectSocket, message, strlen(message), 0);

                printf("\nClient: Your selection is ");

                printf("Forward");

                printf("\n\n");

}

else if (selection == 2)

{

    system("cls");

                char message[100] = "2";

                bytesSent = send(ConnectSocket, message, strlen(message), 0);

                printf("\nClient: Your selection is ");

                printf("Backward");

                printf("\n\n");

}
```

```c
else if (selection == 3)

{

    system("cls");

                    char message[100] = "3";

                    bytesSent = send(ConnectSocket, message, strlen(message), 0);

                    printf("\nClient: Your selection is ");

                    printf("Left");

                    printf("\n\n");

}

else if (selection == 4)

{

    system("cls");

                    char message[100] = "4";

                    bytesSent = send(ConnectSocket, message, strlen(message), 0);

                    printf("\nClient: Your selection is ");

                    printf("Right");

                    printf("\n\n");

}
```

```c
    else if (selection == 5)
    {
        system("cls");
                    char message[100] = "5";
                    bytesSent = send(ConnectSocket, message, strlen(message), 0);
                    printf("\nClient: Your selection is ");
                    printf("Quit");
                    printf("\n\n");
                    break;
    }
    else
    {
        system("cls");
                    char message[100] = "Wrong selection!";
                    bytesSent = send(ConnectSocket, message, strlen(message), 0);
                    printf("\nClient: Wrong selection! Please try again!\n\n");
    }


    }
    WSACleanup();
    system("pause");
    return 0;


}
```

APPENDIX C

(Programming of Machine Control)

```
#ifndef JOGGING_H_INCLUDED

#define JOGGING_H_INCLUDED


#include "iocpp.h"

#include <stdio.h>

#include <conio.h>

#include <iostream>


void jogging(int * arahan)

{

int laju =100;

LoadIODLL();

/******Forward***************/

if (*arahan == 1){

for (int i=0; i<10000;i++){

printf("Running... Forward \n",*arahan);

        PortOut(888,3);  // kiri

        Delay(laju);

        PortOut(888,2);

        Delay(laju);

}
```

```c
}
/******backward***************/
if (*arahan == 2){
for (int i=0; i<10000;i++){
printf("Running... Backward \n",*arahan);
        PortOut(888,1);  // kiri
        Delay(laju);
        PortOut(888,0);
        Delay(laju);
}
}
/******Left*************/
if (*arahan == 3){
for (int i=0; i<10000;i++){
printf("Running... Left \n",*arahan);
        PortOut(888,12);  // kiri
         Delay(laju);
        PortOut(888,8);
        Delay(laju);
}
}
/******Right**************/
if (*arahan == 4){
for (int i=0; i<10000;i++){
printf("Running...fRight \n",*arahan);
        PortOut(888,4);  // kiri
        Delay(laju);
        PortOut(888,0);
```

```
            Delay(laju);

        }


    }
    UnloadIODLL();

    return;

}


#endif // JOGGING_H_INCLUDED
```