國立交通大學

電機與控制工程學系

碩士論文

建構以網際網路為基礎之教學實驗室

Construction of Internet-based Education Laboratory

研究生：江培瑄

指導教授：陳永平 教授

中華民國九十三年六月

建構以網際網路為基礎之教學實驗室

Construction of Internet-based Education Laboratory

研究生：江培瑄　　Student：Pey-Chine Jiang

指導教授：陳永平　　Advisor：Yon-Ping Chen

國立交通大學

電機與控制工程學系

碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

建構以網際網路為基礎之教學實驗室

學生：江培瑄　　　　指導教授：陳永平　教授

國立交通大學電機與控制工程學系

摘要

這篇論文提出一個允許來自不同地方的使用者可以執行線上模擬及透過網路操作實驗設備的遠端處理控制實驗室。在這個整合網路系統中，提供了一個容易使用IE(WWW)的介面供給使用者可以操作而不需要在使用者端安裝任何軟體，及提供一個實際的倒立單擺實驗設備可以讓使用者把已設計好的PID參數透過網路來執行及驗證，和影像伺服器及CCD來提供執行的影像給使用者。使用在這篇論文中所提出的架構，不管主要的伺服器及每個E-labs之間的距離多遠這個整合網路系統都可以容易地被擴充及建立。因此，一個無邊界的教學用實驗室就可以被建立而且也可以方便地供人使用。這個遠端控制實驗可以在http://140.113.149.107/labpage/main/main.ht網址下執行。

Construction of Internet-based Education Laboratory

Student: Pey-Chine Jiang Advisor: Professor Yon-Ping Chen

Department of Electrical and Control Engineering

National Chiao Tung University

ABSTRACT

This paper presents the construction of a remote-accessed control experiment laboratory, which allows the users from everywhere to run the simulation online and to perform control experimental devices via Internet. In this integrated network system, the user friendly interface, IE (WWW), is supplied to the user facilely to operate without installing any software in the client user side, the actually experimental device, inverted pendulum, is provided for the user to operate the experiment using the designed PID parameters, and image server and cameras are offered the user to view the execution result. Using the proposed architecture in this paper, the integrated system could be easily expanded and set up no matter how the distance between the main server and E-labs is. Thus, a boundless education laboratory could be established and conveniently used by the users. The remote experiments could be executed from *http://140.113.149.107/labpage/main/main.htm*.

Acknowledgment

在這兩年的研究所生涯中，首先我要感謝指導老師　陳永平教授的細心指導；在老師的指正與教導下，使得我得以順利完成此篇論文。另外，老師求學治學的方法與待人處事的態度也都讓我獲益良多。此外，也感謝克聰學長與建峰學長在我遇到問題時的解惑與建議，讓我得以解決每次的難題。最後感謝口試委員梁耀文副教授與　張浚林副教授供寶貴的意見，使得本篇論文更甄完整。

其次，感諺可變結構控制實驗室的克聰學長、豐裕學長、建峰學長、天德學長、豐洲學長、依娜、翰宏、智淵、倉鴻、世宏以及學弟們，桓展、昌衢、文魁及欣達在我課業上與生活上的幫助。最後特別感謝我的父母親、家人以及男友健弘對於我的支持與鼓勵。謝謝您們！

僅以此篇論文獻給所有關心我、照顧我的朋友們。

江培瑄　2004.6.28

Contents

List of Figures

List of Tables

Chapter 1 Introduction

In engineering sciences, students are often asked to physically do some experiments if they would like to learn knowledge or skills related to engineering theories or technologies. However, the cost in setting up the experiments is undoubtedly high and perhaps unreasonable, and sometimes the experiments are too far away to execute. Fortunately, the Internet-based technologies have been well developed recently and widely applied to various fields, including educational, commercial, and personal applications. Following swift development and widely using of the Internet, distance-learned is also developed and everyone can obtain more information and search for the answers of the problems that they meet via Internet. The purpose of constructing an internet-based laboratory is offering users to design and to implement their controllers at anytime from everywhere, and providing an expandable and integrable system, including of server machine and E-labs. As for a completely actual controller designed, the mathematical models have to derive first, and the simulation of the system will follow to do. The internet-based laboratory here will also support the function of simulation online for the users to simulate their models without simulation software at the users' place. As for simulation finishing, the designed parameters could be written to the interface of the experiment online and submitted to the server machine. After all execution completing, server side will display the resulted data, figures and video on the browser of the client-side. Using this way, the students could verify the learned skills in the actually experiments no matter how the distance is. Nowadays, the established experimental device is inverted pendulum and the used controller is PID controller. Nevertheless, construction of the internet-based laboratory is an expandable and integrable system, so other built experimental devices could easily be added as following the extended rules.

The paper is organized as follows. In chapter 2, the system architecture, hardware and software implementation and the serial communication protocol are described. In chapter 3, the experiment of Inverted Pendulum and its mathematical model, controller design, simulation results, implementation and experiment results are depicted. In chapter 4, experimental devices run via Internet is presented. In chapter 5, the conclusion and feature work will be mentioned.

Chapter 2 System Architecture Description

2.1 Integrated System

The general configuration of the architecture of Internet-based laboratory is shown in Fig-2.1. This architecture is divided into three parts, including Internet, server machine and E-lab. In Internet, the principal purpose is used to connect the client-side and the server-side and to communicate with each other. In server machine, it is an approach to communicate with the outside world, and to manage each E-lab by WinSock program to be a Network laboratory, and to provide a surrounding where the user could run the simulation and execute the experimental devices online. In E-lab, it contains the sub-server, video device, experimental devices and RS232 transmission port. Its main use is to connect the sub-server and experimental devices with RS232 protocol and to transmit the gained results from the experimental devices back to the main server machine. Using this construction, each E-lab is setup individually with several experimental devices and is managed by the server machine with WinSock program whatever each E-lab is close or distance from the main server machine. By this way, the Internet-based laboratory can conveniently expand, and more multiplex experimental devices could be provided no matter what the distance is long or close, and it easy to maintain by the server manager.

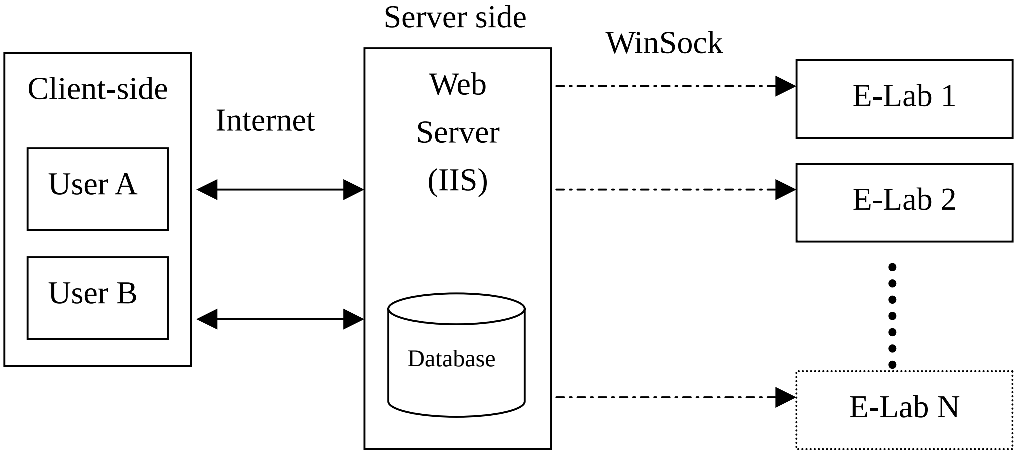


Fig-2.1 Architecture of Internet-based

2.1.1 Internet

In terms of Internet, the user could gain and research the information from everywhere in the world via Internet. Due to the popular WWW (World Wide Web), the users' growth rates are increased, and the resources or information currency are also promoted. According to Internet popularity and widely used now day, the distance learn via Internet could be developed for the user executing and simulating online. In Fig-2.2, the TCP connection-oriented protocol used generally between the server-side and the client-side is presented. As client-side will connect to server-side, socket function in the client-side and in the server-side will setup first. Bind function will prescribe a port number with nothing to do and listen function will go to assigned port and wait for the request from the client-side. After the connection signal dispatching, accept function will receive the signal to process and the connection between server and client will establish. When the connection is built, read and write function could be used with both of them. As the requests from the client-side are finished, the socket function in the client-side will close and the next procedure in the server-side is back to listen function to wait the next connection signal.

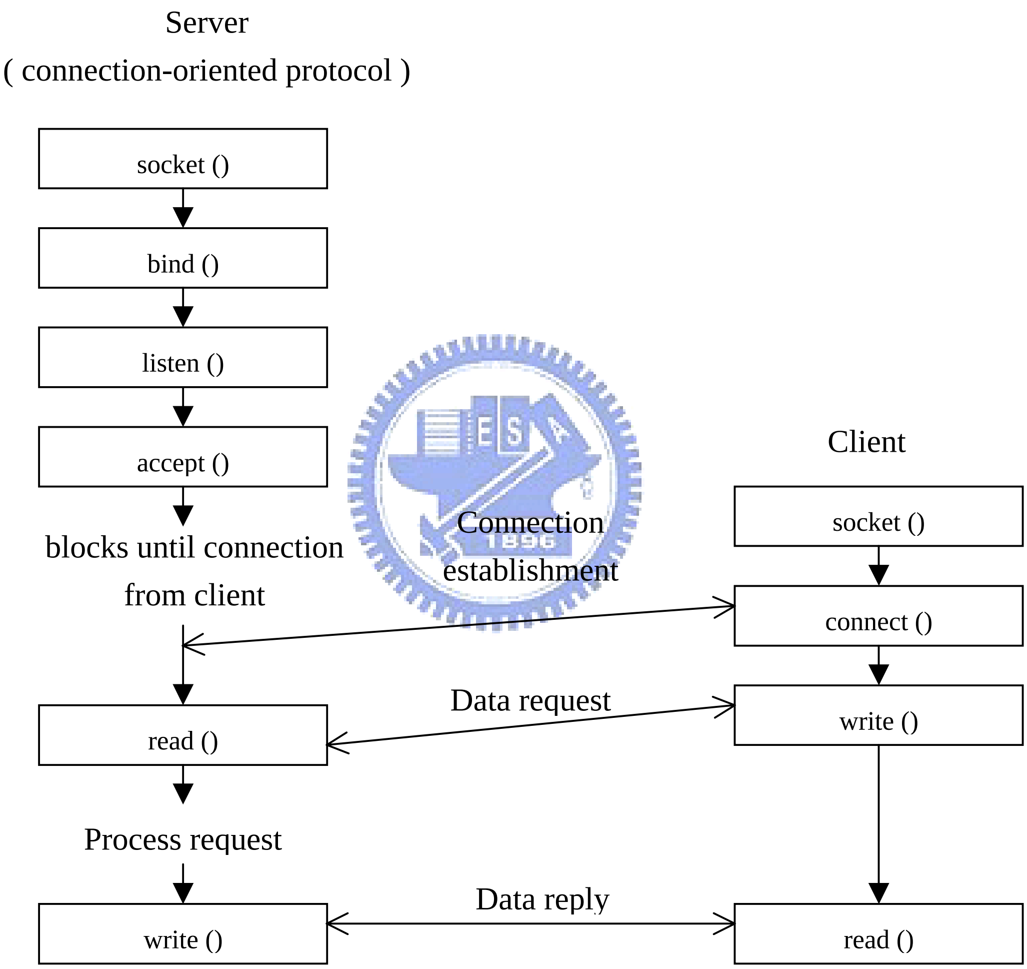


Fig-2.2 TCP connection-oriented protocol

2.1.2 Server Machine

The server machine is responsible for two things, to deal with the request from the outside world via Internet and to manage each E-lab. There are two principal provided functions in the server machine, the one is simulation online and the other one is experiment online. While the user is allowed to execute the simulation online, the server machine will handle with the request directly. If the user’s request is to execute the experimental devices, the server machine acts a transmitter to transmit the massage from the user to the E-lab and acts a manager to receive the resulted data from E-lab back to the user. The communication between the server machine and E-lab are used to Winsock program setup.

2.1.3 E-Lab

The E-lab is a small network for the connections of experimental devices managed by sub-server in the laboratory. The network is defined by “Network is a data communication system allowing a number of independent devices to communicate directly with each other, within a moderately sized geographic area over a physical communications channel of moderate data rates” according to IEEE (Institute of Electrical, Electronics, and Engineering). The construction of each E-lab is shown in Fig-2.3. The sub-server is responsible to connect with each experimental device in the laboratory, and to communicate with the main server machine, and it contains the video server inside. Due to the experimental devices are offered without boundary, there is every kind of the experimental devices could be set up and added to

E-lab. Each laboratory could form a network by itself and experimental devices in laboratory could connect to the sub-server with RS232 protocol. Thus it can be seen, each E-lab is constructed a sub-network.

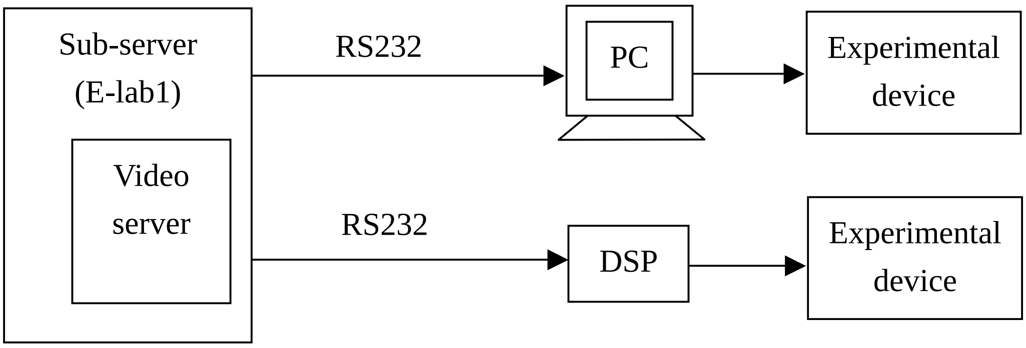


Fig-2.3 Construction of E-lab

All E-labs connect to the center controller, main server machine, by star construction shown in Fig-2.4. The connection between main server machine and sub-server is via TCP connection-oriented protocol introduced in chapter 2.1.1. Despite the distance between E-lab and main server machine, the connection can be established and the communication with each other can be built. The WinSock program for connection will be elaborated in chapter 2.3.3.

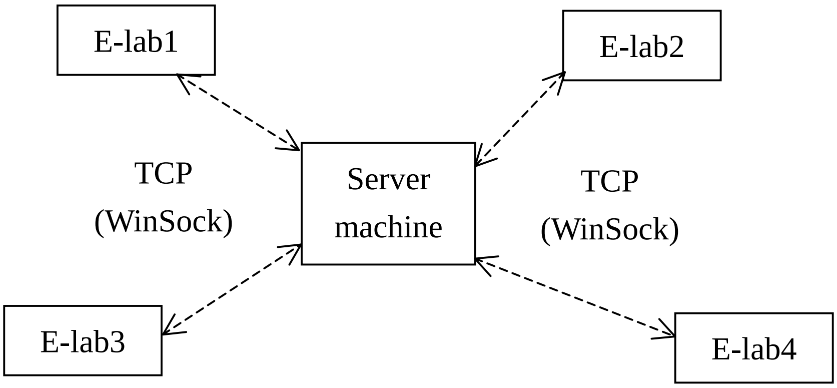


Fig-2.4 Start connection between the server machine and E-labs

2.2 Hardware Implementation

2.2.1 Server machine

The Internet-based laboratory contains the two primary parts, server machine and E-lab. The hardware of the server machine is a PC run Windows 2000 operation system and IIS inside. The server machine acts a center controller to manage all E-lab and to communicate with the external world, so this computer must be more powerful and more equable.

2.2.2 E-lab

The main hardware of each E-lab contains sub server, RS232 interface and experimental devices show in Fig-2.5. The sub server acts as a manager to manage all experimental devices in the laboratory, to process the video and to communicate with server machine. The data from the experimental devices to the server machine or from the server machine to the experimental devices is transmitted via sub server. In sub server, the principal work is to transmit the data without any operation and to process the image for views in the client-side. The hardware of the sub server contains a PC, video capture board, BNC cable and several cameras. Due to the PC just for transmission and video, the sub server needs just the standard computer. The video capture board is TTIC-1000, and the connection line between the video capture and the cameras is BNC cable. Between the sub server and the experimental devices, the RS232 interface is widely used to communicate with each other. As for the experimental devices, they could be diversification and difference, such as inverted pendulum, motor control, robot control and so on. As a result of the experimental devices and controller are not all the same, the used hardware is also different. Therefore, the hardware of this part will not be emphasized or be introduced later as the experimental devices are used.

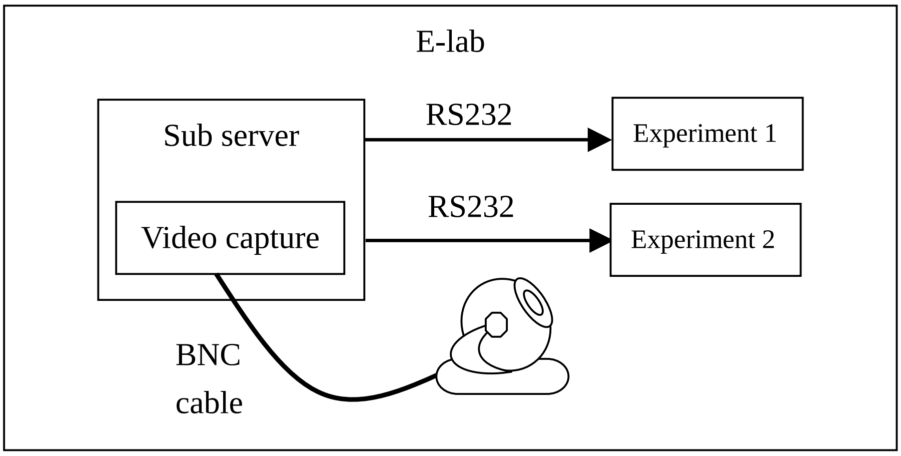


Fig-2.5 Hardware system of E-lab

2.3 Software design

The software of this system contains six parts, User interface, Simulation online, Experiment online, WinSock program, Process of the resulted data and other functions of the server-side.

2.3.1 User Interface

The design of user interface is dependent on HTML language (Hypertext

Mark-up Language), the default scripting language supported by the Hypertext Transfer Protocol (HTTP), and ASP (Active Server Page) which is operated on PWS (Personal Web Server) or IIS (Internet Information Service) in server-side. The client user could obtain the data from the database in the server-side by using the ASP program provided in server-side. The interaction between the server-side and the client-side is shown in Fig-2.6. The user in the client-side could request to gain the information in the server-side and sometimes could send the data to the database in the server-side. Sometimes, the user is allowed to run the program provided by the server machine. When the user in the client-side demands of the requests, the server machine will feedback the different resulted data or information according to the different requests.

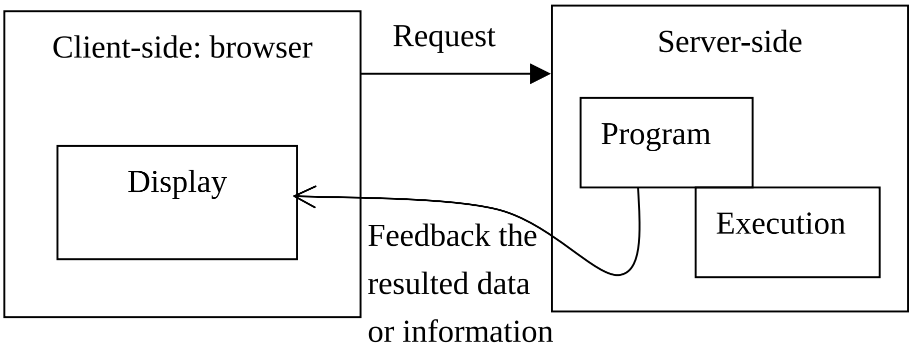
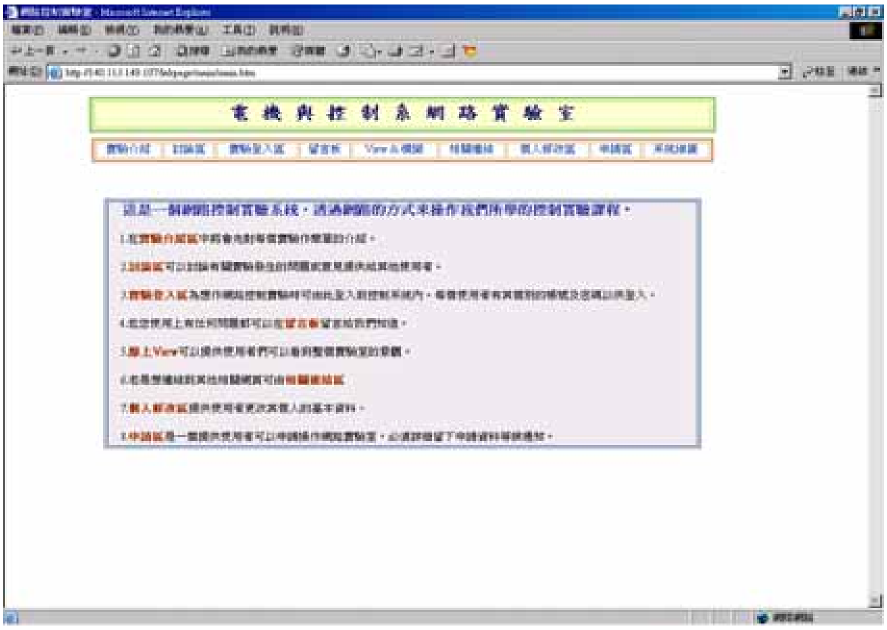


Fig-2.6 The interaction between the server-side

The user-friendly interface shown in Fig-2.7 contains eight sub-items, such as simulation and experiment online, experiment introduction, discussion field and so on, and all sub-items will be introduced simply.



電機與控制系網路實驗室

實驗介紹 | 討論區 | 實驗登入區 | 留言板 | View [illegible] | 相關連結 | 個人修改區 | 申請區 | 系統維護

這是一個網路控制實驗系統，透過網路的方式來操作我們所學的控制實驗課程。

1.在實驗介紹區中將會先對每個實驗作簡單的介紹。

2.討論區可以討論有關實驗發生的問題或意見提供給其他使用者。

3.實驗登入區為操作網路控制實驗時可由此登入到控制系統內，每個使用者有其個別的帳號及密碼以供登入。

4.若您使用上有任何問題都可以在留言板留言給我們知道。

5.線上View可以提供使用者們可以看到整個實驗室的景觀。

6.若是想連結到其他相關網頁可由相關連結區

7.個人修改區提供使用者更改其個人的基本資料。

8.申請區是一個提供使用者可以申請操作網路實驗室，必須詳細留下申請資料等候通知。

Fig-2.7 User-friendly interface

1. Experimental Introduction item could supply the introduction, mathematical model and the process for the controller design of each experimental system.

2. Discussion field item could provide a space for the user to consult about the experimental design or simulation online with one anther. Using this way could encourage the user to express their opinion and to solve the solutions of the problems.

3. Experiment login item contains login system and experiment online system. As the user demand to run the remote experimental device, the login system will display first. If the user login the correct ID and password, they could be allow to execute the experiments online. After finishing the execution, the resulted data and figure will feedback to the user in the client-side. Experiment online will be

elaborated in section 2.3.3.

4. Message board item could offer the user to leave word to the administrator, if they find the problem for login name or password, or they do not understand how to perform the user interface.

5. View and simulation item will be separated to describe.

   I. The all view online for everyone could supply the user to watch the experimental surrounding and equipments. As the user run the experiments, the views of the execution motion will be presented to the user. The view interface is shown in Fig-2.8.

   II. Simulation online could provide the user to simulate their controller design without simulation software at the user's own place. Simulation online will be elaborated in section 2.3.2.

6. Related link item has related website addresses or other useful addresses inside.

7. Personal revise area provides the user to modify their basic data, such as e-mail, phone number, login ID, login password and so on.

8. Application area could supply the user to apply the login ID and login password of the experiment online.

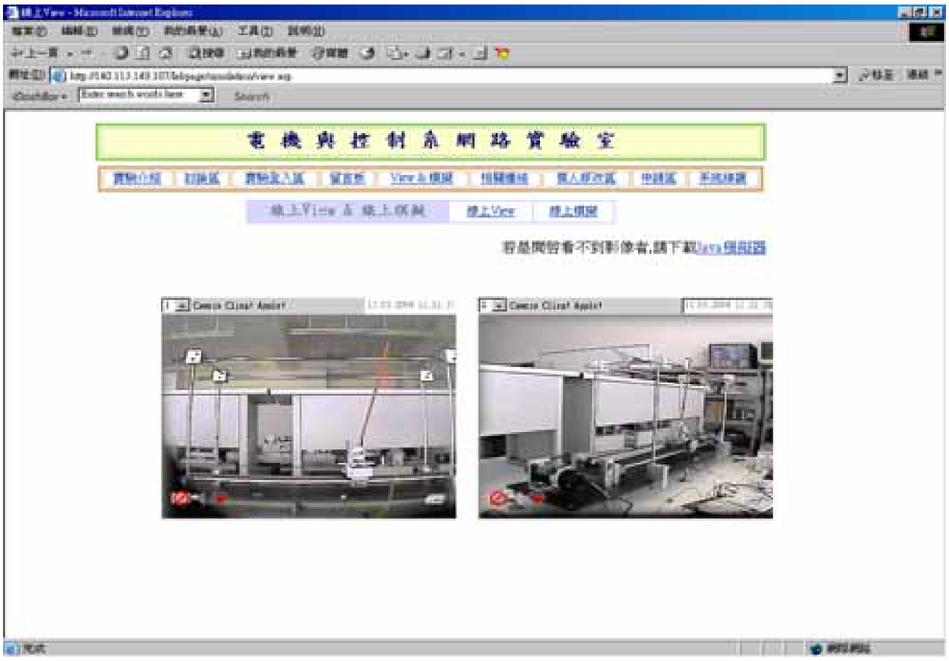


Fig-2.8 The view interface

2.3.2 Simulation Online

In order to allow the client-side users to run assigned simulation projects without simulation software at their own place, a special function, simulation online, of User-Interface will be needed, which is shown in Fig-2.9. The main concepts and work of simulation online will be elaborated in this section.

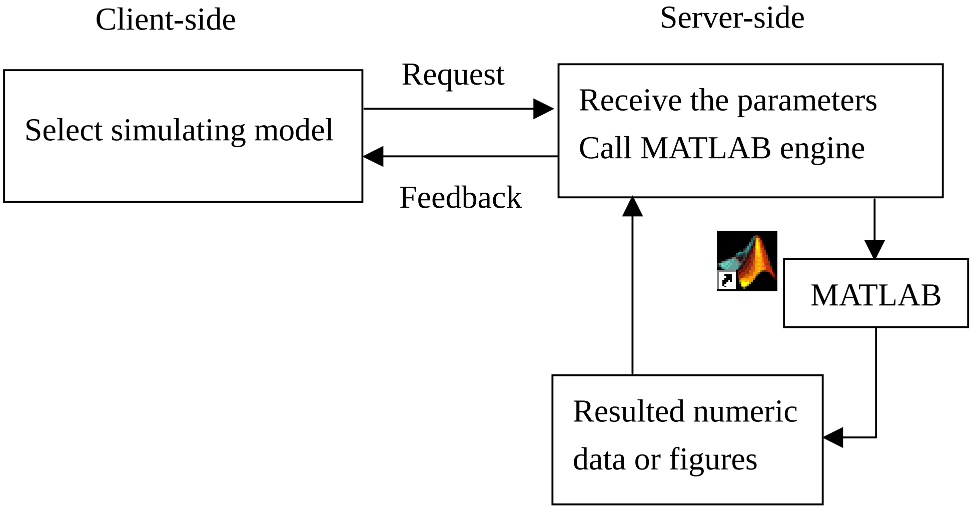
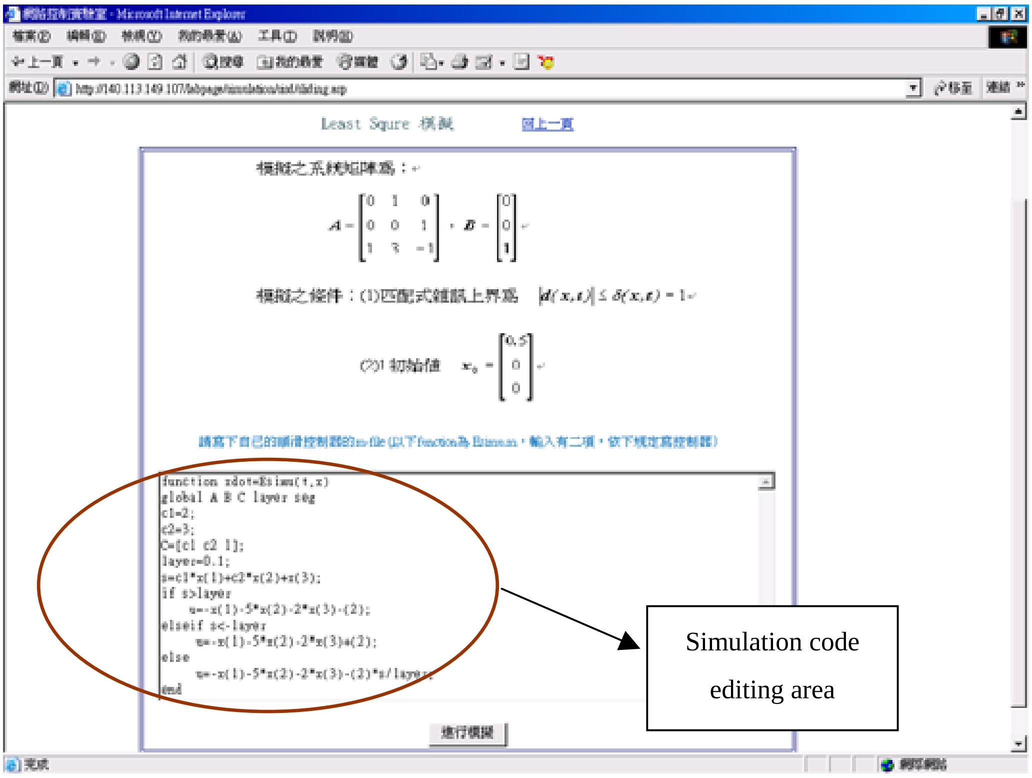


Fig-2.9 Architecture of simulation online

First, the client-side user connects the server and pushes down the simulation item in the user-interface. Then, the page will display the mathematical models, which are designed previously. These models could be online executed by the user. After the user selects the model demanded, the simulation rules stipulated will display and teach the user how to design the controller and how to use simulation online correctly. The users have to follow the rules and write the designed controller in the edition area, before running the simulation. After the controller data are written into the edition area, the users could submit the commands and the designed parameters to the server-side. The commands and parameters received in the server-side will be combined into a string separated by '!' symbol, such as "command_name!parameter!", and then store the string to the database. However, before the commands and designed parameters are received, the VB program in the server-side has been opened and waited to deal with the simulation commands from the client-side. After the VB program was opened, the VB object for MATLAB engine will be established by using CreateObject ("Matlab.Application"). As the requests from the client user have been received by the server-side, a state in the database will be checked by the VB program first to determine whether the user could use the MATLAB engine or not. If the state reveals executable, the program could obtain the string from the database and decompose the string to executed commands and designed parameters. Next, the commands and the parameters will be executed in the MATLAB engine, and no matter what the execution results are correct or incorrect, all the resulted data, figures, or wrong messages will be stored, and then fed back to the user and displayed on the browser in the client-side. Noticed that these resulted data and figures could be returned to the user's own computer, otherwise they will be canceled when the user executes at next time or a day later. If the execution is completed, the state will be changed back to a usable state for next users, otherwise, the browser will display "Program is used by other users now and please try again later".

For an example, the system is a 3order differential equation with one input and one matched disturbance. Now, design a single input sliding mode controller to control the system stable and the all states to achieve zero. As the user select this one, the introduction of the system and the controller edition area are shown in Fig-2.10 first. After the submitted commands and the designed parameters have been executed in the server-side, the execution resulted numeric data and figure are displayed in Fig-2.11.



網路控制實驗室 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
http://140.113.149.107/labpage/simulation/sad/sliding.asp
Least Squre 模擬
回上一頁
進行模擬

模擬之系統矩陣為：

$$A=\begin{bmatrix}0&1&0\\0&0&1\\1&3&-1\end{bmatrix},\ B=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

模擬之條件：(1)匹配式雜訊上界為 $|d(x,t)|\le\delta(x,t)=1$

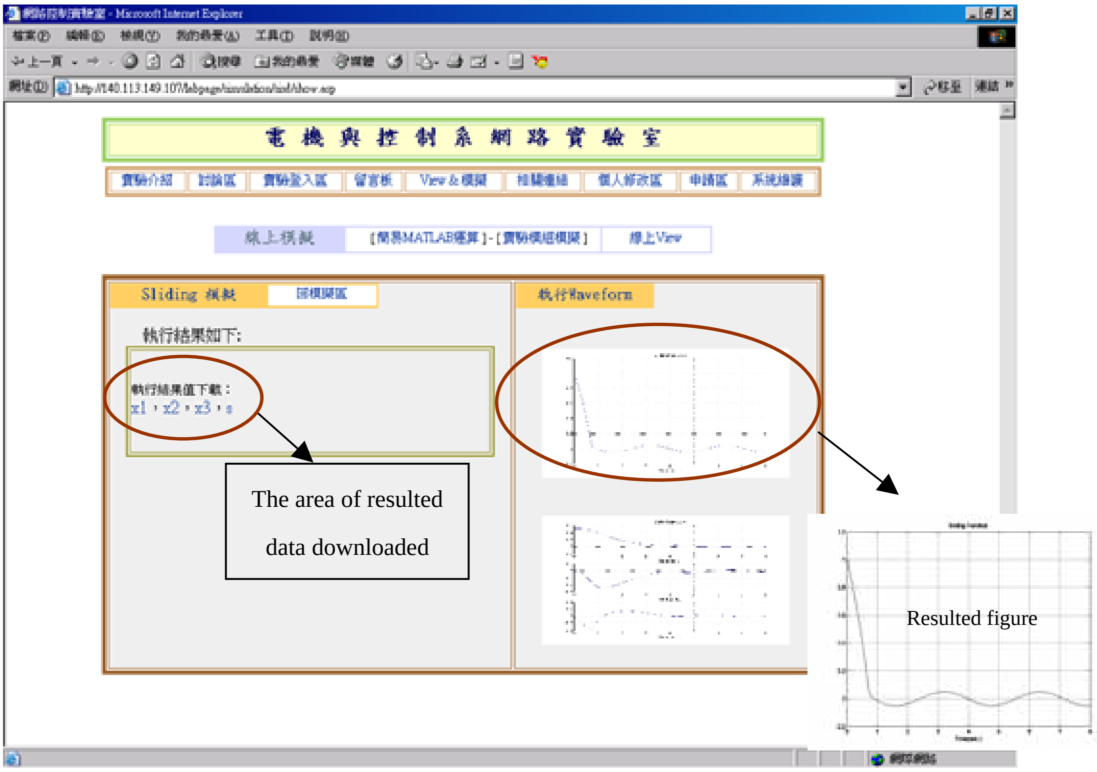
(2)1 初始值 $x_0=\begin{bmatrix}0.5\\0\\0\end{bmatrix}$

請寫下自己的順滑控制器的m-file (以下function為 Esimu.m，輸入有二項，依下規定寫控制器)

```
function xdot=Esimu(t,x)
global A B C layer seg
c1=2;
c2=3;
C=[c1 c2 1];
layer=0.1;
s=c1*x(1)+c2*x(2)+x(3);
if s>layer
    u=-x(1)-5*x(2)-2*x(3)-(2);
elseif s<-layer
     u=-x(1)-5*x(2)-2*x(3)+(2);
else
     u=-x(1)-5*x(2)-2*x(3)-(2)*s/layer;
end
```

Simulation code editing area

Fig-2.10 Edition interface of simulation online



網路控制實驗室 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
http://140.113.149.107/labpage/simulation/sad/show.asp
電機與控制系網路實驗室
實驗介紹
討論區
實驗室入區
留言板
View & 模擬
相關連結
個人修改區
申請區
系統維護
線上模擬
[簡易MATLAB運算]-[實驗模組模擬]
線上View
Sliding 模擬
回模擬區
執行Waveform

執行結果如下:

執行結果值下載：
x1，x2，x3，s

The area of resulted data downloaded

Resulted figure

Fig-2.11 Resulted data and the figure after executing

2.3.3 Experiment online

Following development of Internet, it is possible for the client users from everywhere to run the experimental devices online. The function of experiment online is in order to establish an environment that provides many kinds of remotely experimental devices and controllers for the client users, who would like to execute the experimental devices at everywhere. In this section, the interface and software design of experiment online will be introduced. The flow chart to run the experiments from the client side to the experimental devices is shown in Fig-2.12.

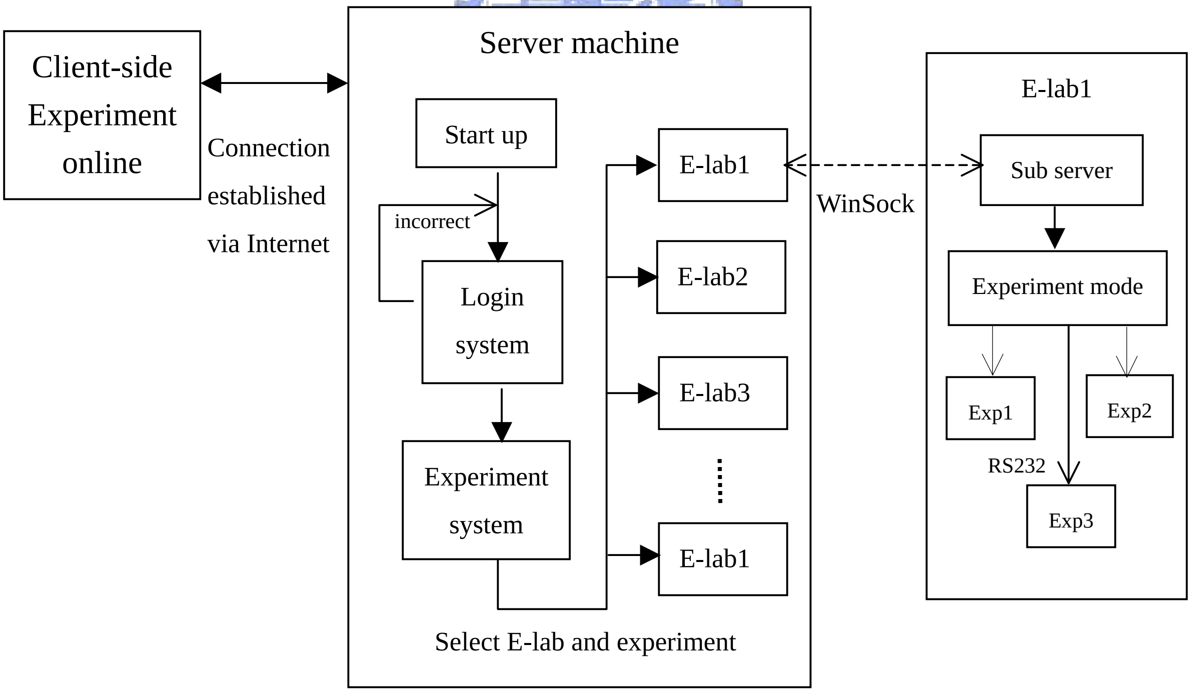


Fig-2.12 Flow chart to run the experiments online

The connection between the client-side and the server-side must be established first, if the user would like to run the experiment online. Following the connection established, the login system will display in the browser of the client-side, as the user selects experiment login item of the user interface. The login system is necessary in order to manage the use states of the users. The user, who wants to run experiment online, must fill in the application data of user interface and must pass through the application. After allowing running the experiment online, the login ID and login password will be given to the user by the server machine. While the user logins to the experiment system with the correct login ID and login password, the experiment system will be shown in the following pages. Many kinds of the experiments from different E-labs are presented, and the user could select one of them to execute. Each experiment has its model introduction and several controllers shown in Fig-2.13.

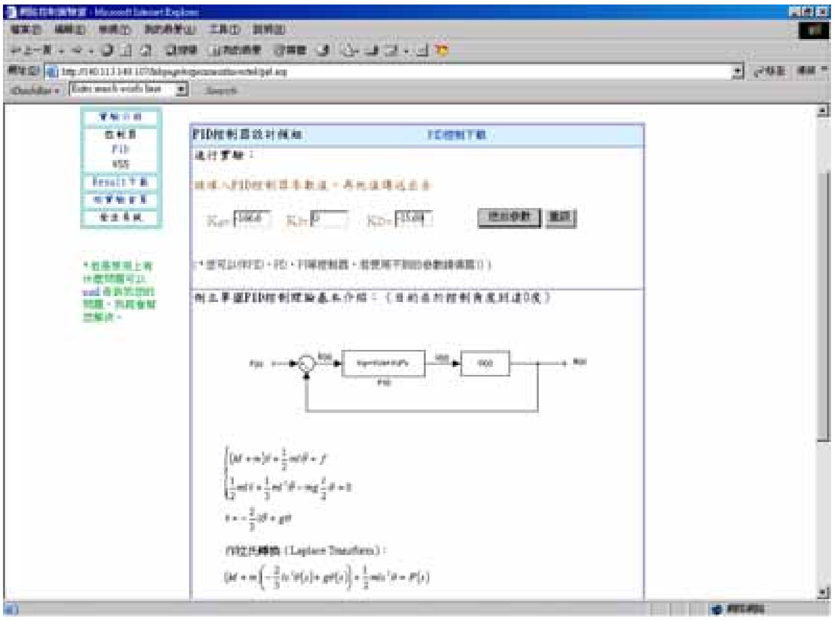


Fig-2.13 Operation interface of experiment online

As the parameters of the designed controller by the user is submitted, the parameters and the selected controller way will be composed of a string as “experiment_number!controller_name!parameters!”, and store to the database in the server machine. Here, an experimental device just allows one user to use at the moment, but different experimental devices could provide for the different users to operate at the same time. In Fig-2.14, when the user selects one of the experiments to run, the state will be changed to the executable state and be saved to the database. If the user 1 and the user 2 both select the experiment 1 at the same time, the user who selects the experiment faster could use the chosen experiment, but another one must try again after the state become usable state. The experiments maybe locate in different laboratories, and each of them has its experimental number. The experiments in the same laboratory could be managed by the sub server to become an E-lab network. As the user selects one experiment, the VB program will be selected to use according to the experimental number. One VB program is responsible to one experiment and has been opened to wait for processing the user’s demand, and then VB program will detect the state in the database executable or not. If the state is executable state, VB program will transmit the execution command to the E-lab via WinSock program. The WinSock program existed in the main server and in the sub server will present particularly in section 2.3.4. In the sub server of E-lab, the execution command, “experiment_number!controller_name!parameters!”, will be decomposed first. According to experiment number decomposed, the sub server could setup the connection and send the execution command to the chosen experiment via RS232 interface. The serial communication interface will be detailed in section 2.4. When the chosen experiment receives the execution command, the controller way and parameters of command will be decomposed first, and the experiment will be execute by the decomposed controller way and parameter. After finishing the experiment, the state will become to the usable state for the next user. Another state is used to check the experiment finished or not, and is used to notify the server machine to feedback the resulted data from E-lab to the client side. After the resulted data and the figures have been processed completely, the end command will send to the E-lab and notify the experimental devices to wait for next requests. The process of the resulted data will be elaborated in section 2.3.5.

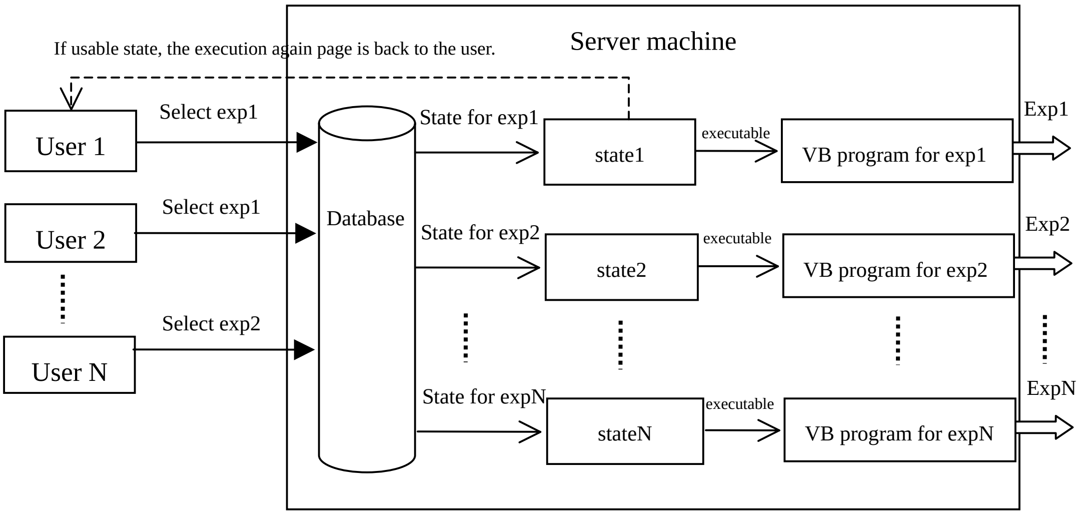


Fig-2.14 Execution process of experiment online

2.3.4 WinSock program

The connection between the main server and the sub server is using WinSock program via TCP (Transmission Control Protocol). In each E-lab, the sub server has its own IP address in order to communicate with the main server. Due to the distance, the transmission using TCP/IP is very convenient without the line between each other. The WinSock program is used in Windows operation system and contains two parts, WinSock server and WinSock client shown in the Fig-2.15.

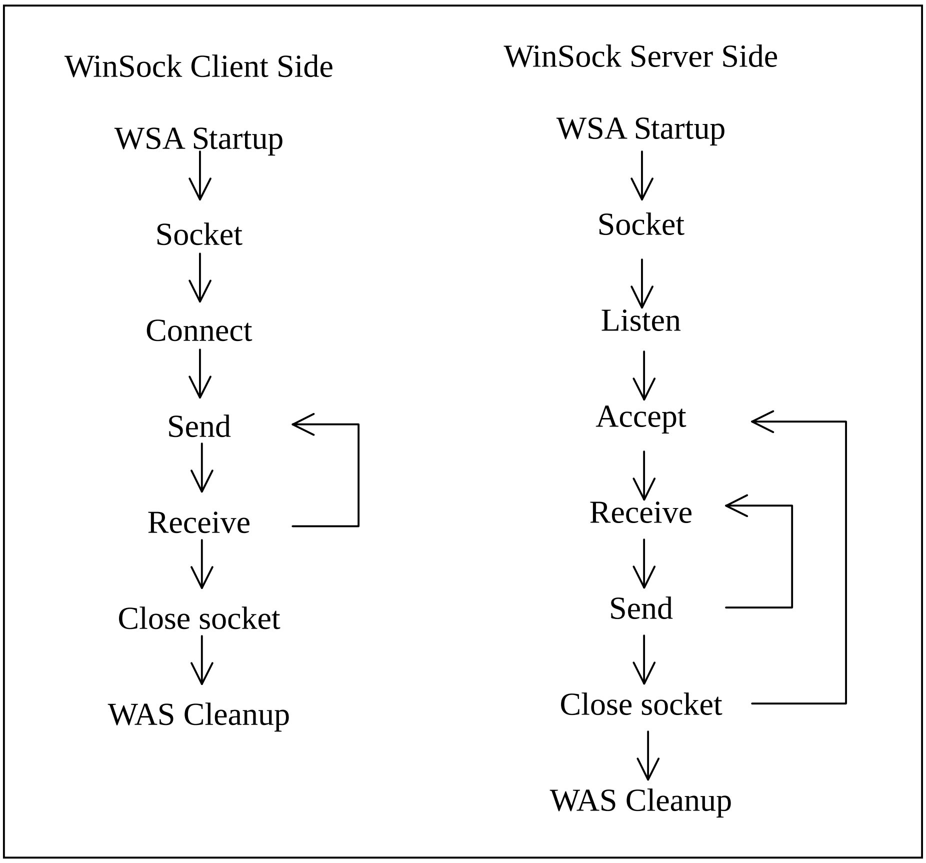


Fig-2.15 WinSock Model

In WinSock server side, the flow path of the program is to open a socket and to wait for the request from WinSock client side by listen function. As the connection is established, and the request is accepted to process, WinSock client side could receive and send the data to WinSock server side. After completing all transmission, the socket will be closed, and the state in WinSock server side will go back to listen state to wait for the next request. In WinSock client side, a socket is opened first when WinSock client side would like to send or receive the data from WinSock server side. As the connection is built, WinSock client side could communicate with WinSock server side. Until the all transmission finished, this socket will be closed. The WinSock program is implemented by VB language that contains WinSock object.

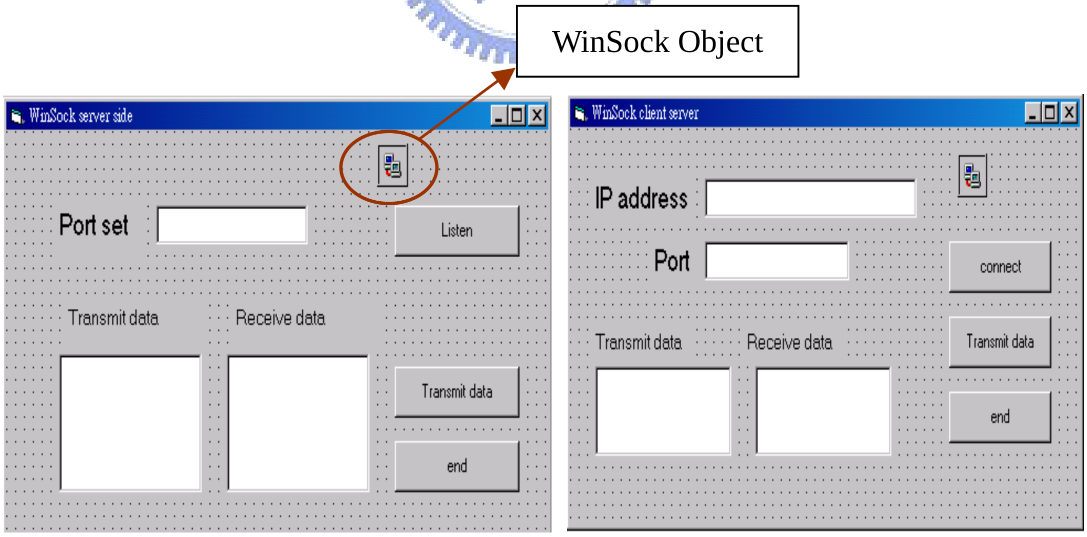
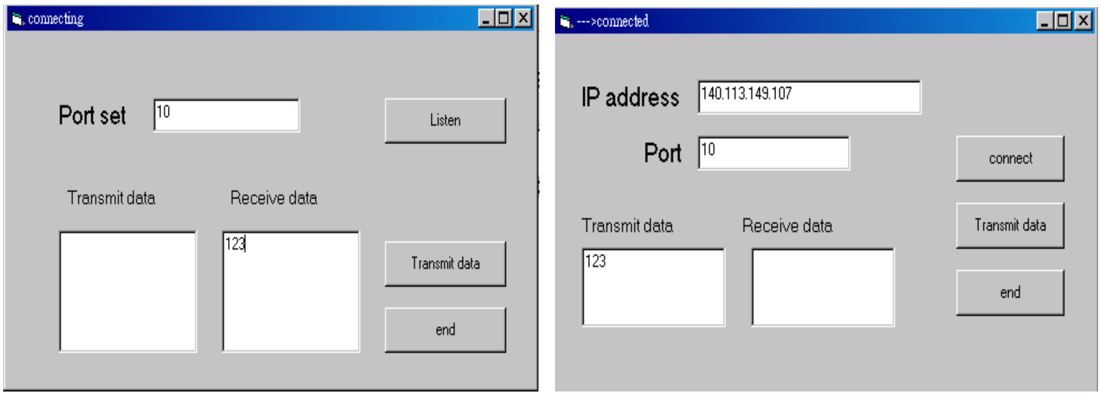


Fig-2.16 WinSock Interface

In Fig-2.16, the interface of WinSock program between WinSock server side and

WinSock client side by VB language is presented, and the read mark is the WinSock object used to communicate. In left interface, WinSock server side is provided a server to be asked to connect, so the IP address is according the IP used in this machine. Before the listen function running, the WinSock server must assign a port for the connection from WinSock client side. In right interface, WinSock client side has to fill the IP address and port number in order to build the connection. As the right IP and assigned port are filled in, and the connect button is clicked, the connection will be established. Then, the two sides could send the data and receive the data, and the simple communication between the two sides could be setup. The example for easy communication is shown in Fig-2.17.



connecting

Port set: 10

Transmit data:

Receive data: 123

--->connected

IP address: 140.113.149.107

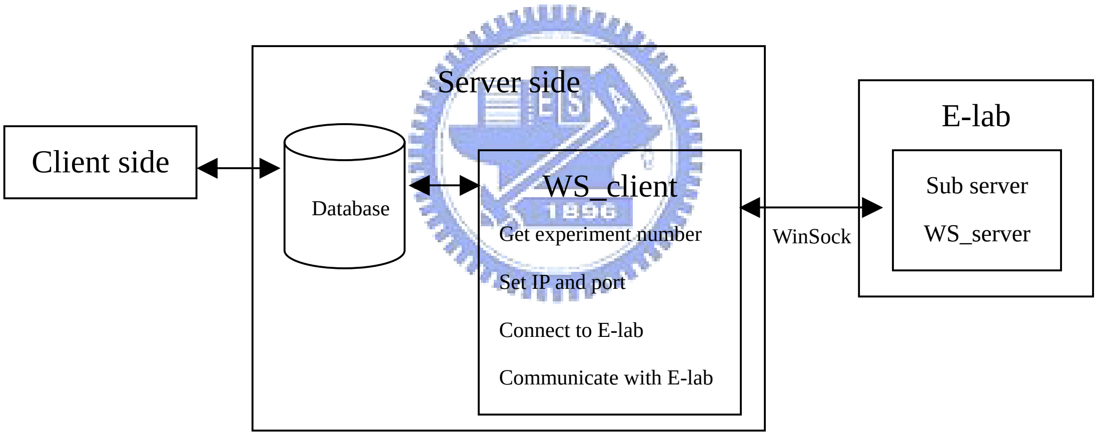
Port: 10

Transmit data: 123

Receive data:

Fig-2.17 Example for WinSock communication

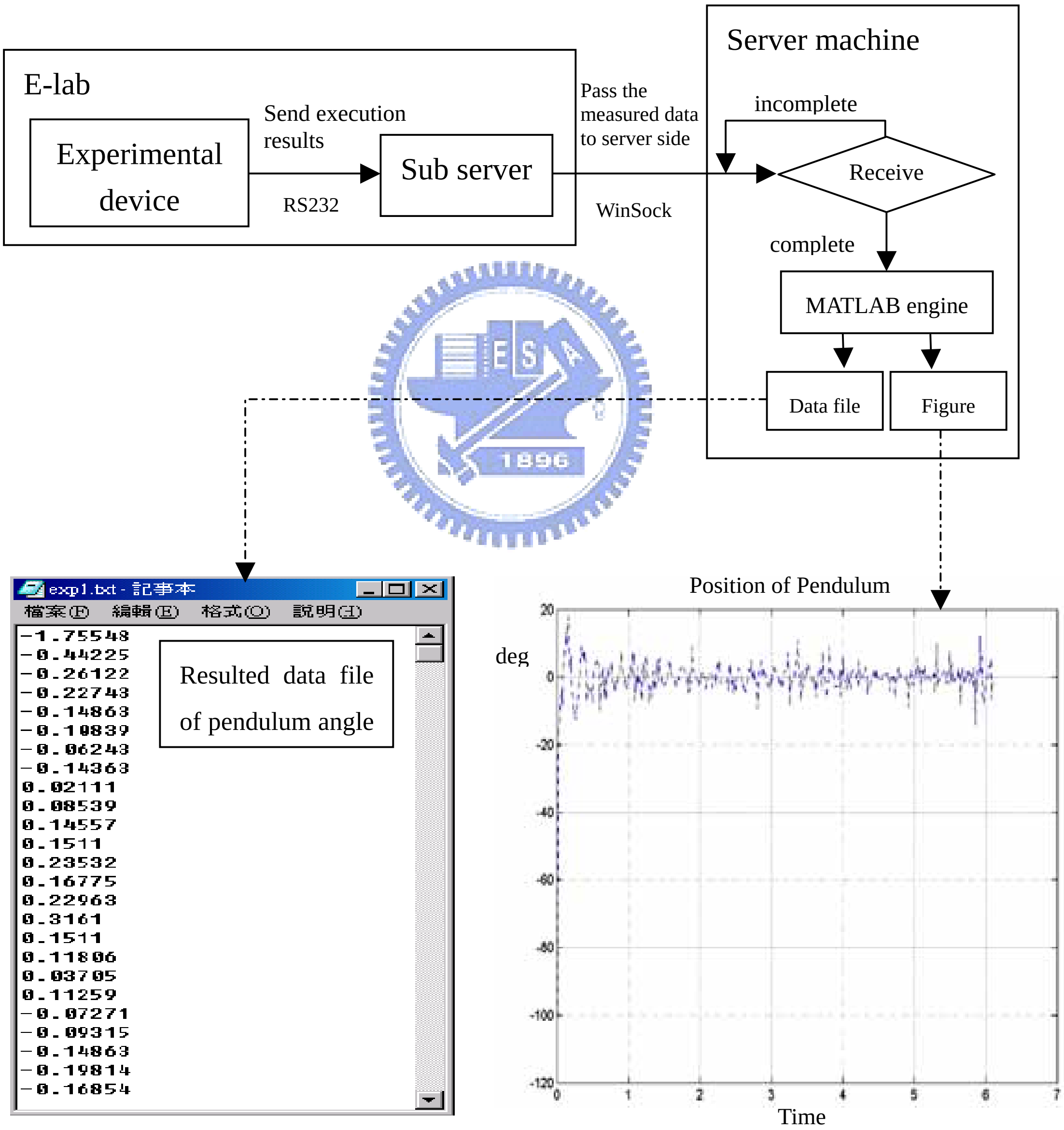
In Fig-2.18, the connection between the main server machine and sub server in E-lab is using WinSock way. The main server could run the program in WinSock client side and the sub server could run the program in WinSock server side. By this way, the main server could receive the data from the user in the client side and easier send to the connection for the chosen E-lab, or receive the data from E-lab and send to the user in the client side. If the program in WinSock server is built in the main server machine, the encountered problem is that the sub servers in E-labs are too far away the administrator to manage conveniently. According the experiment number selected by the user in client side, the main server can set different IP and port number for every E-lab, and make the connection and the communication.



Fin-2.18 WinSock in the main server and E-lab

2.3.5 Process of the resulted data

After the experimental device chosen by the user in the client side has been executed completely, the measured data will transmit to sub server in E-lab by RS232 interface. Then, the data received to the sub server will send the data directly to the main server via WinSock program. When the resulted data have received to the main server, the program in the main server will check the data correct or not and store the received data to .txt file. If whole the resulted data have acquired, the program will plot the figure of the resulted data. The approach of plotting the figure is also use of MATLAB engine. In Fig-2.19, flow chart of processing the resulted data, and the resulted data using .txt file format and the figures are shown.



```
-1.75548
-0.44225
-0.26122
-0.22743
-0.14863
-0.10839
-0.06243
-0.14363
0.02111
0.08539
0.14557
0.1511
0.23532
0.16775
0.22963
0.3161
0.1511
0.11806
0.03705
0.11259
-0.07271
-0.09315
-0.14863
-0.19814
-0.16854
```

Fig-2.19 Flow chart of processing measured data

2.4 Serial Communication Interface

The communication between two devices contains two ways shown in Fig-2.20, serial and parallel transmission way. However, the communication between sub-server and experimental devices in E-lab is via serial communication protocol because the transmission speed rate of serial port is accepted, and the transmitter using serial way is cheaper than using parallel way.

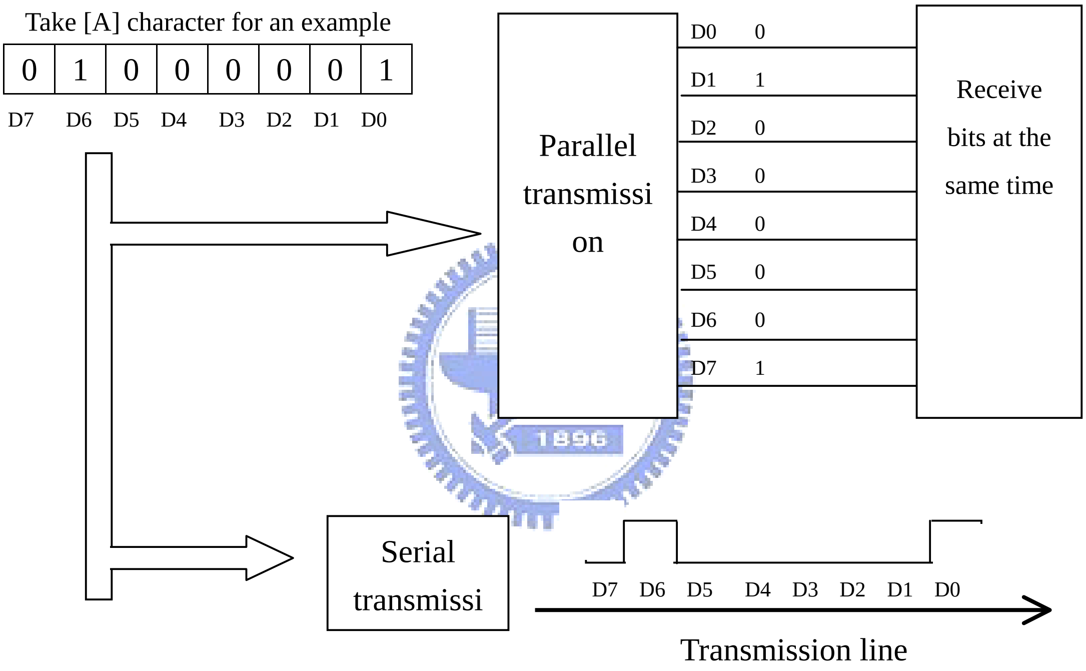


Fig-2.20 Parallel transmission and serial transmission

2.4.1 RS232 Interface Introduction

In serial communication interface, RS232 interface is popular and widely used and well developed. RS232 interface contains 9 pins and 25 pins but now the general used way is 9 pins RS232 interface. The meanings of the pins of RS232 interface are defined in Table 2.1.

Pin number	Abbreviation	Meaning
Pin 1	CD	Carrier Detect
Pin 2	RXD	Receive
Pin 3	TXD	Transmit
Pin 4	DTR	Data Terminal Ready
Pin 5	GND	Ground
Pin 6	DSR	Data Set Ready
Pin 7	RTS	Request To Send
Pin 8	CTS	Clear To Send
Pin 9	RI	Ring Indicator

Table 2.1 Meanings for each pin of RS232 interface

The common connection between two devices by RS232 interface is exhibited in Fig2.21. Pin 2 in device 1 must connect to pin 3 in device 2 and so on, because pin 2 is used to transmit the signal and pin 3 is used to receive the signal.

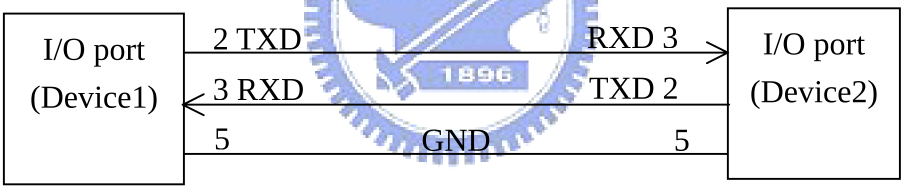


Fig-2.21 Common connection of the two RS232 interface

There are two ways to take only one pair of wires to shift data bit by bit, the synchronous technique and the asynchronous technique. Synchronous transmission is considered a more efficient approach than asynchronous. Nevertheless, asynchronous transmission is much easier to implement and work with. Here, asynchronous transmission presented in Fig-2.22 is used to connect to the two devices. The start bit and stop bit is used to determine for asynchronous transmission.

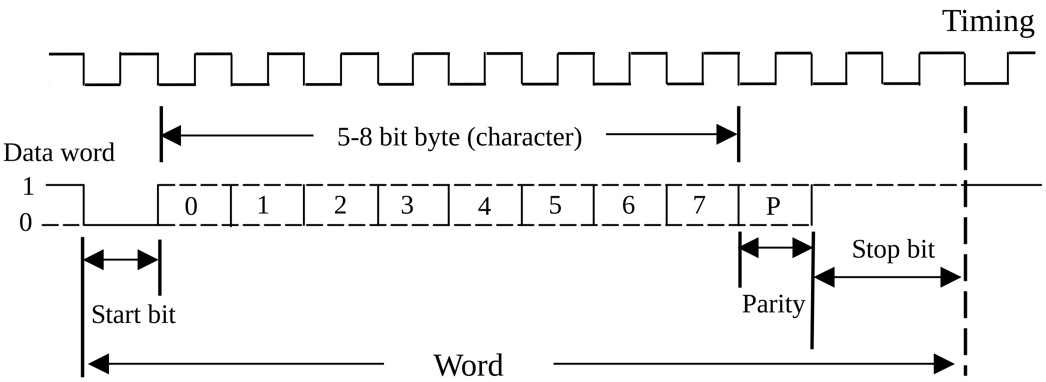
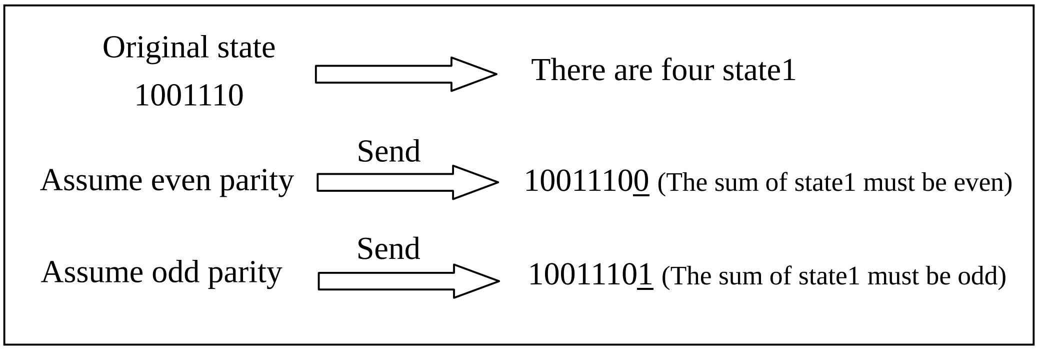


Fig-2.22 Asynchronous Transmission

In general, the transmission line without proceeding to communicate is held on logic 1. As for levels and electric characteristics of RS232: +3V to +15V stands for SPACE (logic 0) and –3V to –15V stands for MARK (logic 1). When data is transmitted by asynchronous transmission, the transmission signal is composed of one start bit, a character byte, one parity bit and the stop bit. The start bit is the logic from 1 to 0 with a clock. Following start bit is the data byte that could be 5 bits, 6 bits, 7 bits or 8 bits due to the type of the character, such as ASCII code or JIS code. Next one is parity bit used to detect the data correct or not and it has three modes, even parity, odd parity or no parity. The mechanism of parity bit is shown in Fig-2.23.



Original state 1001110 → There are four state1

Assume even parity —Send→ 10011100 (The sum of state1 must be even)

Assume odd parity —Send→ 10011101 (The sum of state1 must be odd)

Fig-2.23 Transmission with parity bit

The stop bit express the data transmission is finished, and it could be 1 bit, 1.5 bits and 2 bits. The asynchronous transmission protocol in the two devices must be the same. Another important thing is the same reference timing in order to read/write the correct data. The up bound of signal communication rates is called baud rate or BPS (Bit Per Second) meaning bit number for transmission per second. If the reference timing is different in the two devices, the data in transmit side and in receive side won't be in tune. In Fig-2.24, there are two different sampling frequencies to obtain the resulted data.

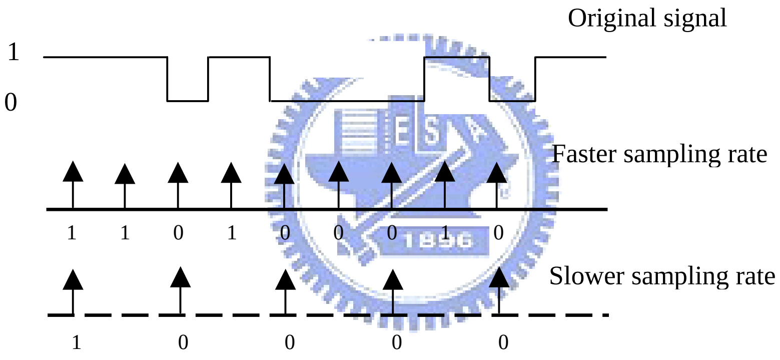
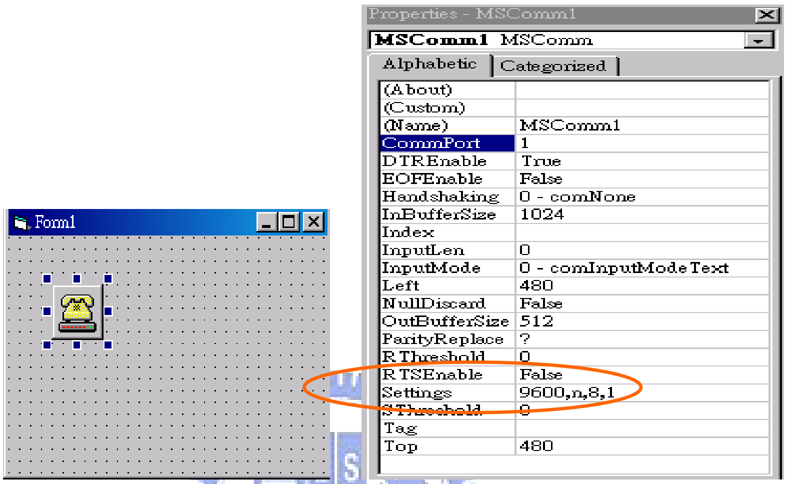


Fig-2.24 Different sampling rates

2.4.2 Software Design

The implementation of RS232 communication could use the software design or hardware design. Cause of the connection for the two PC in E-lab, the software design for RS232 interface is used. The used program for sub-server is VB program, which have RS232 protocol object inside. The RS232 object and its property are shown in Fig-2.25. The setting property could set baud rate, number of character bits, parity mode and stop bit. As to terminal devices connected to sub-server via RS232 interface, the varied kinds of devices maybe use different programming language to setup transmission protocol, such as assembly, Borland C++ Builder, VB and so on, so the software design of terminal devices won’t introduce one by one.



(About)	
(Custom)	
(Name)	MSComm1
CommPort	1
DTREnable	True
EOFEnable	False
Handshaking	0 - comNone
InBufferSize	1024
Index	
InputLen	0
InputMode	0 - comInputModeText
Left	480
NullDiscard	False
OutBufferSize	512
ParityReplace	?
RThreshold	0
RTSEnable	False
Settings	9600,n,8,1
SThreshold	0
Tag	
Top	480

Fig-2.25 RS232 object and it’s property

This shows that a transmission network in E-lab is not hard to setup. A sub-server could manage the devices in E-lab and transmit the data with each other by RS232 interface. As the command from the client user has been received, the main server machine transmits the command to the sub server in E-lab by WinSock, and the program in WinSock server side combined RS232 transmission object could communicate with the devices. The data is transmitted from the devices to the client user by the same transmission approach. When the experimental devices are executed, the measured data will send back to sub server via RS232 interface. In sub server, the received data will send back to the main server via WinSock at the same time. After running a space, the experimental devices will send the ending signal to notify the sub server and the main server to finish the execution.

Chapter 3 Experiment of Inverted Pendulum

Here, the inverted pendulum is used to be the experimental device in E-lab. In this chapter, the total system and mathematical model of inverted pendulum will be described. Then, the controller design methods and simulation results will also be presented. Finally, the hardware and the software of the system implementation and the experiment results will be elaborated.

3.1 Mathematical Model

The schematic representation of inverted pendulum is shown in Fig-3.1 and the nomenclatures of the variables are given below. The mathematical model of the system is constructed under the following assumptions: (1) Pendulum is restricted to the vertical plane containing the line of the rails. (2) The driving force is directly applied to the cart without delay. (3) Pendulum is rigid and uniform in thickness. (4) The friction force of the cart is constant.

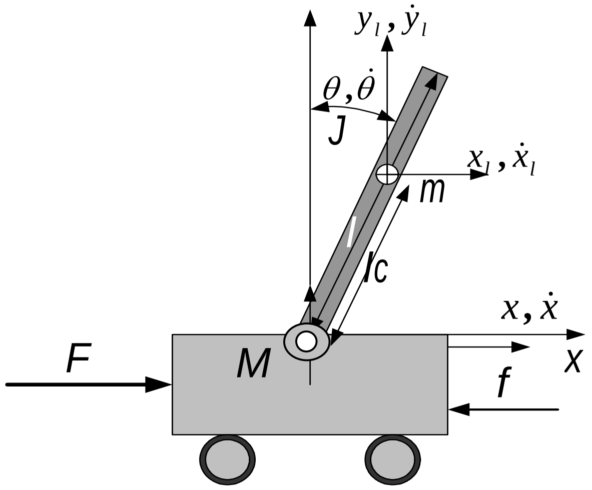


Fig-3.1 Schematic representation of inverted pendulum

Based on these assumptions, the mathematical model is derived by using the Lagrange's formulation. First, the kinetic energy of the inverted pendulum system can be found as:

$$\begin{aligned}\boldsymbol{T} &= \frac{1}{2}\boldsymbol{M}(\,\dot{\boldsymbol{X}}\,)^2 + \frac{1}{2}\boldsymbol{m}\left[\left(\boldsymbol{L}\cos\theta\dot{\theta} + \dot{\boldsymbol{x}}\right)^2 + \left(\boldsymbol{L}\sin\theta\dot{\theta}\right)^2\right] + \frac{1}{2}\left(\frac{1}{12}\boldsymbol{m}(2\boldsymbol{L})^2\right)\left(\dot{\theta}\right)^2 \\ &= \frac{1}{2}(\boldsymbol{M}+\boldsymbol{m})\left(\dot{\boldsymbol{X}}\right)^2 + \boldsymbol{mL}\cos\theta\dot{\theta}\dot{\boldsymbol{X}} + \frac{2}{3}\boldsymbol{mL}\left(\dot{\theta}\right)^2\end{aligned} \tag{3.1}$$

and the potential energy is

$$\boldsymbol{V} = \boldsymbol{mgL}\cos\theta \tag{3.2}$$

Then the Lagrange scale function is obtained as

$$\begin{aligned}\boldsymbol{L} &= \boldsymbol{T} - \boldsymbol{V} \\ &= \frac{1}{2}(\boldsymbol{M}+\boldsymbol{m})(\,\dot{\boldsymbol{X}}\,)^2 + \boldsymbol{mL}\cos\theta\dot{\theta}\dot{\boldsymbol{X}} + \frac{2}{3}\boldsymbol{mL}^2\left(\dot{\theta}\right)^2 - \boldsymbol{mgL}\cos\theta\end{aligned} \tag{3.3}$$

Applying the Lagrange method

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{X}}\right) - \frac{\partial L}{\partial X} = u - f \tag{3.4}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \tag{3.5}$$

We have

$$(M+m)\ddot{X} - mL\sin\theta\left(\dot{\theta}\right)^2 + mL\cos\theta\ddot{\theta} = u - f \tag{3.6}$$

$$mL\cos\theta\ddot{X} + \frac{4}{3}mL^2\ddot{\theta} = mgL\sin\theta \tag{3.7}$$

M: total mass of cart m: total mass of pendulum

L: half-pendulum length u: control force

f: friction of cart on track

$\boldsymbol{X}, \dot{\boldsymbol{X}}, \ddot{\boldsymbol{X}}$: position, velocity, and acceleration of cart

$\theta, \dot{\theta}, \ddot{\theta}$: angular position, velocity and acceleration of pendulum

By using the state equation way, (3.6) and (3.7) could be changed to

$$\begin{aligned}\begin{bmatrix}\ddot{x}\\ \ddot{\theta}\end{bmatrix} &= \Delta_s\begin{bmatrix}\frac{1}{3}ml^2 & -\frac{1}{2}ml\cos\theta\\ -\frac{1}{2}ml\cos\theta & M+m\end{bmatrix}\begin{bmatrix}u-f+\frac{1}{2}ml\sin\theta\dot{\theta}^2\\ \frac{1}{2}mgl\sin\theta\end{bmatrix}\\ &= \Delta_s\begin{bmatrix}\frac{1}{3}ml^2(u-f)+\frac{1}{6}m^2l^3\sin\theta\dot{\theta}^2-\frac{1}{4}m^2l^2g\cos\theta\sin\theta\\ -\frac{1}{2}ml\cos\theta(u-f)-\frac{1}{4}m^2l^2\sin\theta\cos\theta\dot{\theta}^2+(M+m)mg\frac{l}{2}\sin\theta\end{bmatrix}\end{aligned} \tag{3.8}$$

where

$$\Delta_s = \frac{1}{\frac{1}{3}(M+m)ml^2-\frac{1}{4}m^2l^2\cos^2\theta}$$

This is the state equation of inverted pendulum by nonlinear form. The part of controller design will be presented in next section.

3.2 Controller design

In this section, there are three methods for the controllers of inverted pendulum. PID controller is used to keep the pendulum in upright position. Pole assignment method is used to control the pendulum in upright position and the cart position in center position. As for sliding mode method, it is based on the pole assignment method, but this method using in resisting the matched disturbance is better than pole assignment method. As usual, the nonlinear system must be linearized, before designing the controller. Considering the unstable equilibrium point at θ =0 of the system i.e., $\theta \cong 0$, and $sin\,\theta \cong \theta$, $cos\,\theta \cong 1$, the model could change to

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} M+m & \frac{1}{2}ml \\ \frac{1}{2}ml & \frac{1}{3}ml^2 \end{bmatrix}^{-1} \begin{bmatrix} u-f \\ \frac{1}{2}mgl\theta \end{bmatrix} \tag{3.9}$$

The controller could be design by linear form, but the initial angle position should be close to θ =0 for reaching a good control in practice .

3.2.1 PID controller design

The objective of the control is to keep the pendulum in upright position. There are many methods to design a controller such that inverted pendulum locates at the upright position. One of the best-known controllers used in practice is PID controller, where PID stands for proportional integral derivative. The model block diagram of this linearized system using PID controller is shown in Fig-3.2.

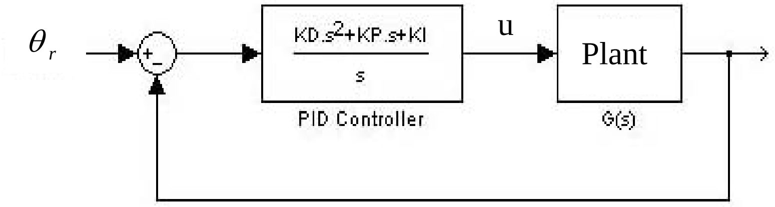


Fig-3.2 System block diagram using PID controller

The Laplace transform of the system plant about pendulum angle is

$$G(s)=\frac{\theta(s)}{F(s)}=\frac{-1}{\left(\frac{2}{3}M+\frac{1}{6}m\right)ls^2-(M+m)g}=\frac{k}{s^2-a} \tag{3.10}$$

The transfer function of a PID controller can be written

$$H(s)=K_P+K_Ds+\frac{K_I}{s} \tag{3.11}$$

and over-all closed-loop transfer function $M(s)$ could be rewritten as

$$GH(s)=\frac{\theta(s)}{R(s)}=\frac{\frac{k}{s^2-a}\left(K_p+\frac{K_I}{s}+K_Ds\right)}{1+\frac{k}{s^2-a}\left(K_p+\frac{K_I}{s}+K_Ds\right)}=\frac{ks\left(K_p+\frac{K_I}{s}+K_Ds\right)}{s^3+kK_Ds^2+(kKp-a)s+kK_I} \tag{3.12}$$

Then, let the poles of $M(s)$ in the stable roots, and put the $\boldsymbol{K}_\mathrm{P}$, $\boldsymbol{K}_\mathrm{D}$, and $\boldsymbol{K}_\mathrm{I}$ in the system to achieve the system stable. However, the Laplace transform in (3.11) for the PID controller of pendulum angle has some problems. When $\boldsymbol{K}_\mathrm{D}$ is equal zero, all the poles in (3.12) could not be assigned to the stable side. When $\boldsymbol{K}_\mathrm{I}$ is equal zero, the system with the friction will cause greater steady states error than without the friction.

3.2.2 Pole assignment design

In order to control the pendulum in upright position and the cart position in center position, this method in Fig-3.3 could achieve the goal and easily design.

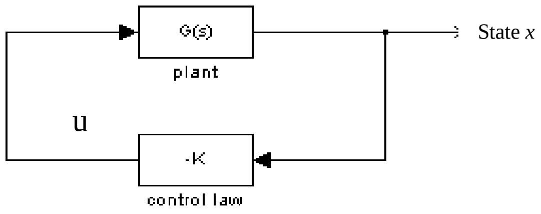


Fig-3.3 System block diagram using pole assignment

The system could change to the state equation as

$$\dot{\underline{x}} = A\underline{x} + \underline{b}(u - f) \tag{3.13}$$

Before design pole assignment, the system should be checked controllable or not first.

If the system is controllable, the states could feedback the system by $u = -\underline{K}\,\underline{x}$.

where

$$\underline{x} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{-3mg}{2(4M+m)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{3(M+m)g}{2(4M+m)l} & 0 \end{bmatrix}, b = \begin{bmatrix} 0 \\ \dfrac{4}{2(4M+m)} \\ 0 \\ \dfrac{-3}{2(4M+m)l} \end{bmatrix}, \underline{K} = \begin{bmatrix} K_1 & K_2 & K_3 & K_4 \end{bmatrix}$$

This method is used to assign the poles of the system to the stable poles and to control the states to the reference values.

3.2.3 Sliding Mode Controller Design

Sliding mode control, fundamentally, is a feedback control containing switching elements. There are two modes [11], approaching mode meaning that the trajectory of the system (3.12) contacts the sliding surface in finite time and sliding mode meaning that let the system to limit in sliding surface and to approach the goal finally. According to the two modes, there are two steps as designing SM controller.

Step 1: Choosing a suitable sliding surface $s(x)$

The sliding surface is chosen as $s = cx$, where $\underline{c} = [\text{c1} \quad \text{c2} \quad \text{c3} \quad \text{c4}]$. On the sliding surface, i.e., s=0, according to equivalent control theory, the system trajectories must be conformed as

$$\dot{s}(x)\Big|_{u=u_{eq}} = \underline{c}\dot{x} = \underline{c}Ax + \underline{c}\underline{b}u_{eq} - c\underline{b}f = 0 \tag{3.14}$$

For making the system trajectories sliding to the control targets on the sliding surface, the system should be designed to the stable system by choosing a suitable $\underline{c}$. After choosing the sliding function, it could start to design the control rule.

Step 2: Design control rules

There are two control rules, contacting the trajectory the sliding surface in finite time and generating the sliding mode. The condition could be expressed as

$$s\dot{s} < -\sigma|s|,\quad \sigma > 0,\ \text{for}\quad s \neq 0 \tag{3.15}$$

Assume the sliding function selected, the control input could be chosen as

$$u = -(\underline{c}\underline{b})^{-1}\underline{c}Ax - (\underline{c}\underline{b})^{-1}(\gamma+\sigma)\cdot\frac{s}{\|s\|} \tag{3.16}$$

where $\gamma = \|\underline{c}\|\delta(x,t) > \|\underline{c}d(x,t)\|$. For avoiding chattering, (3.15) could be modified as

$$u = -(\underline{c}\underline{b})^{-1}\underline{c}Ax - (\underline{c}\underline{b})^{-1}(\gamma+\sigma)\cdot sat(s,\varepsilon) \tag{3.17}$$

where $sat(s,\varepsilon) = \begin{cases} \dfrac{s}{\|s\|} & \|s\| > \varepsilon \\ \dfrac{s}{\varepsilon} & \|s\| \le \varepsilon \end{cases}$

As the control input is found, the sliding controller design completely. The system trajectories will be control to the origin on sliding surface in finite time. About sliding controller method, the detail could refer to reference [11].

3.3 Simulation Results

Here, numerical simulations for PID, pole assignment and sliding mode controller have preformed. The parameters of inverted pendulum are in Table-3.1.

Parameter	M	m	l
Value	$\leq$2.5[kg]	0.09[kg]	0.48[m]

Table-3.1 Parameters of inverted pendulum

Considering PID controller first, the $\boldsymbol{K}_{\mathrm{P}}$, $\boldsymbol{K}_{\mathrm{D}}$, and $\boldsymbol{K}_{\mathrm{I}}$ values could be design following (3.10) and (3.11). The simulation results with $K_P = \text{-}118.6161, K_I = \text{-}181.7905$ and $K_D = \text{-}14.9719$, initial angle $\theta = 0.2$ and the disturbance $|f| \leq 3.5$ are shown in Fig-3.4. This controller just is used to keep the pendulum in upright position. Using pole assignment method, it is used to control the pendulum in upright position and the cart position in center position. The states of the system feedback to system and achieve the control goal. In Fig-3.5, the simulation results where $\underline{k} = [\text{-}121.5369 \quad \text{-}21.8136 \quad \text{-}20.4686 \quad \text{-}23.3927]$ and initial $x_0 = 0.2$, $\theta_0 = -0.2$ is shown, but this method has some troubles when the system has disturbance. As for sliding mode controller, it is more robust in resisting the disturbance than state feedback. Following the design method in section 3.2.3, the sliding function could choose $\underline{c} = [\text{-}15.7014 \quad \text{-}2.8036 \quad \text{-}7.3026 \quad \text{-}5.9232]$, initial $x_0 = 0.1$, $\theta_0 = -0.08$ and the disturbance $|f| \leq 3.5$. The simulation results using sliding layer $\varepsilon = 0.1$ are shown in Fig-3.6 and the simulation results using sliding layer $\varepsilon = 0.01$ are shown in Fig-3.7. In Fig-3.6 and Fig-3.7, due to the different layers chosen, the smaller layer, the faster chattering. However, comparing to state feedback method and sliding mode control method as the disturbance existing in the system model, the sliding mode control method is better than another in resisting the disturbance.

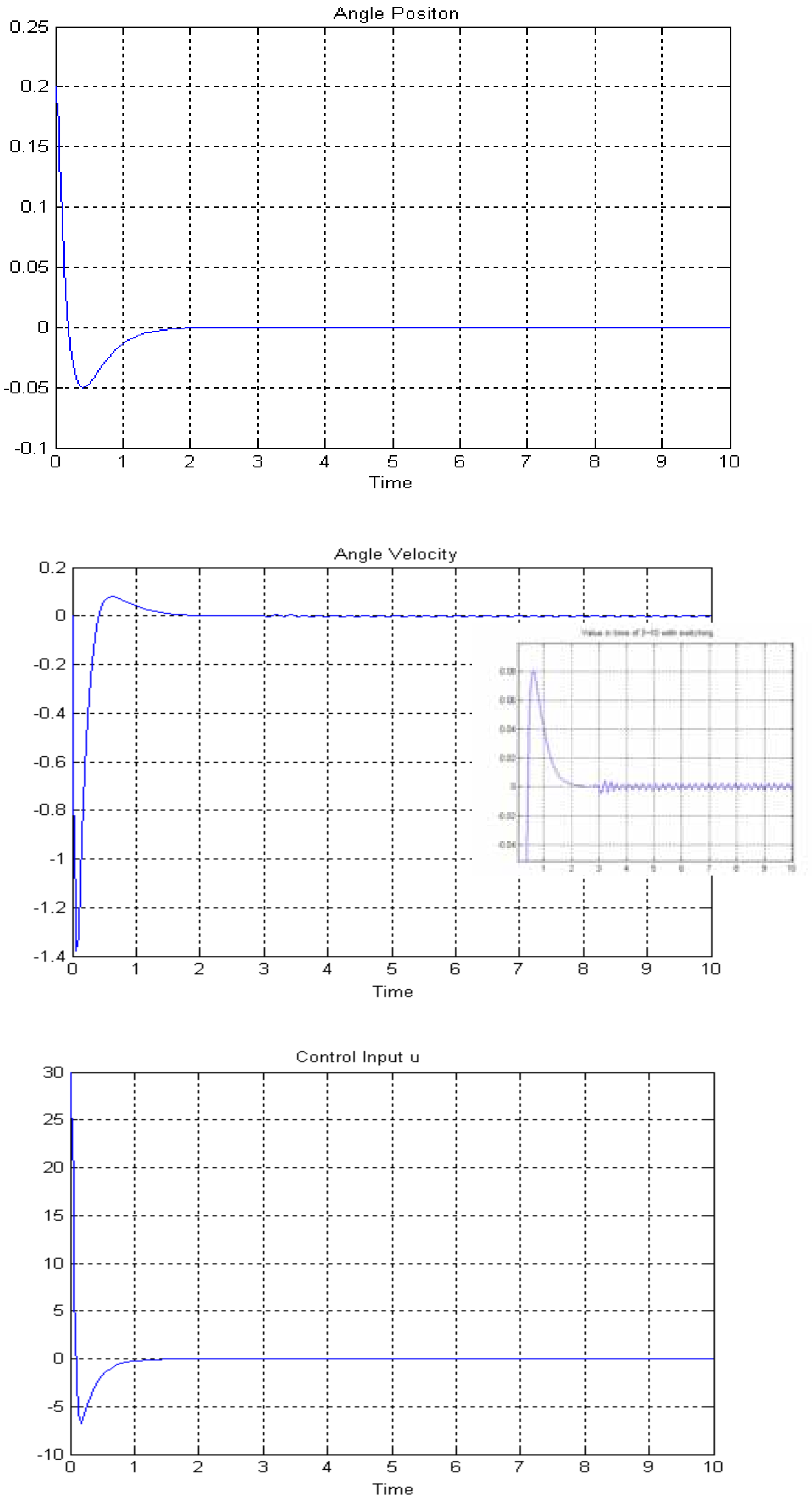
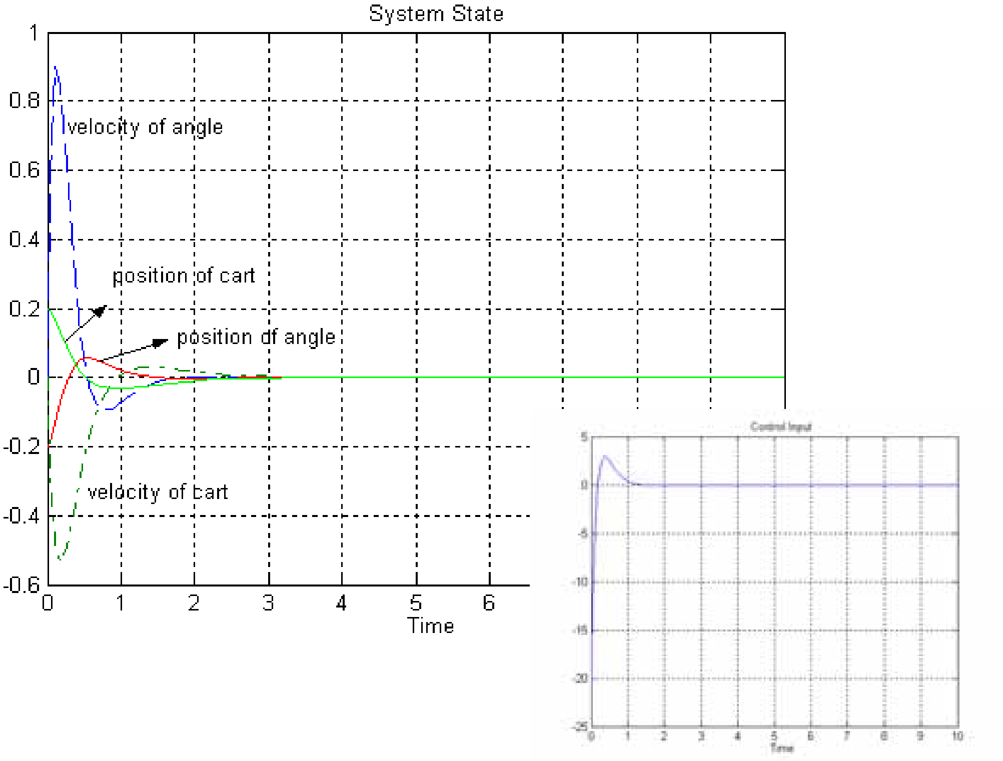


Fig-3.4 Simulation results of PID controller

Without disturbance



Without disturbance $|f| \leq 3.5$

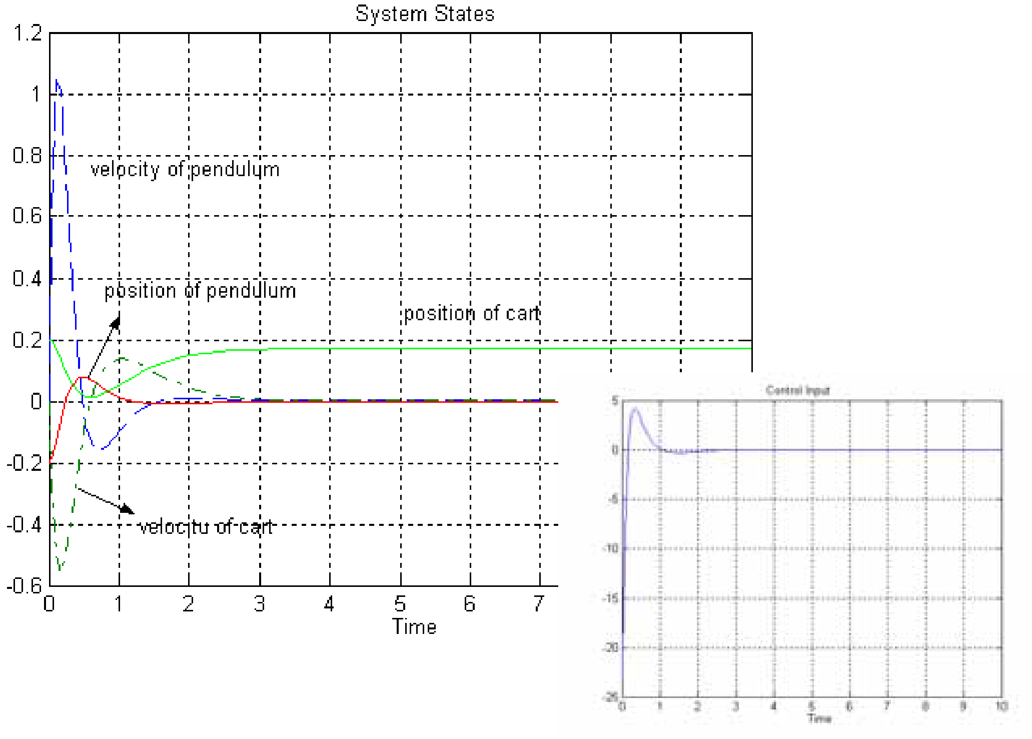


Fig-3.5 Simulation of state feedback

$\varepsilon = 0.1$, $|f| \le 3.5$

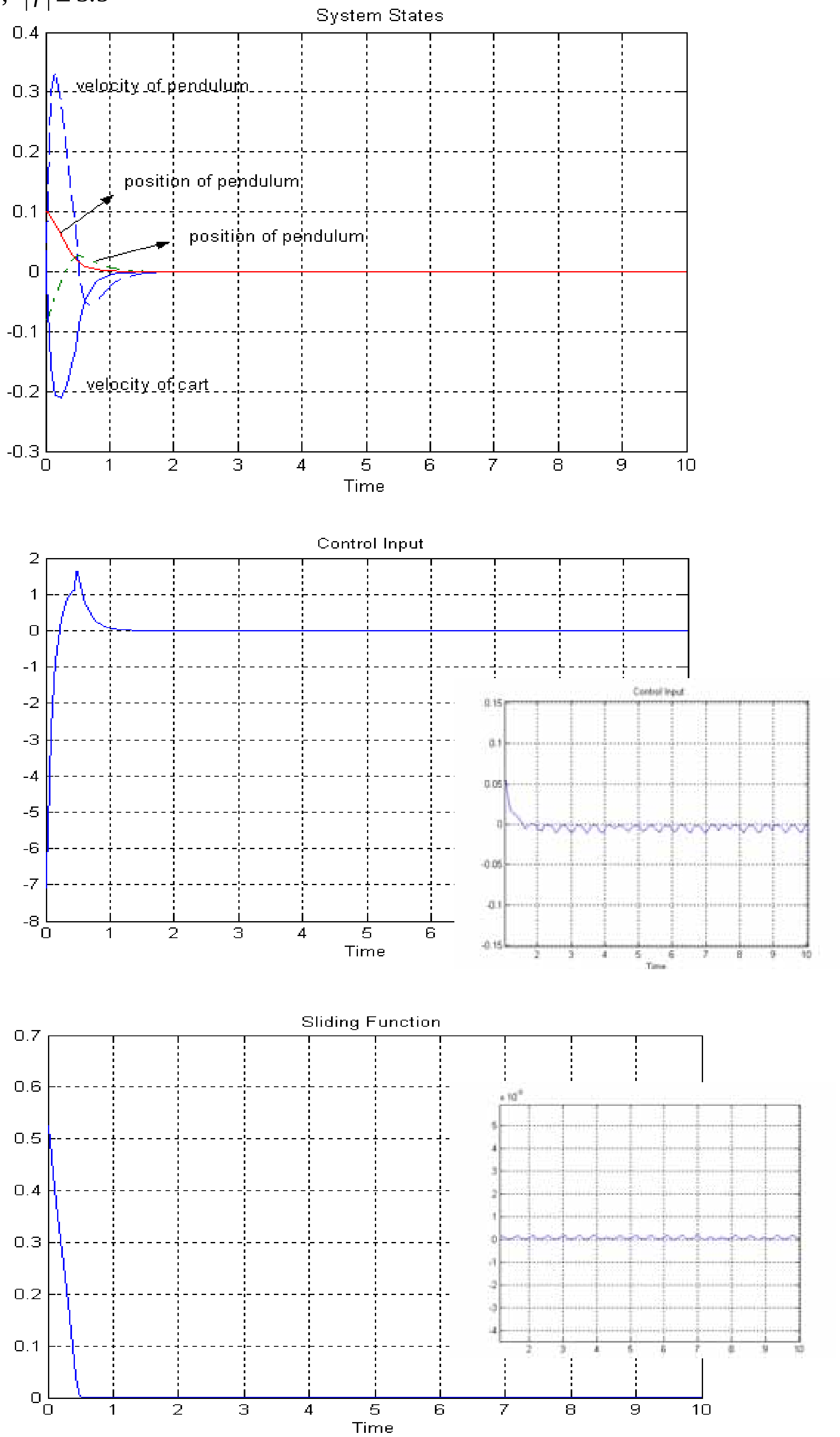


Fig-3.6 Simulation results of sliding mode

$\varepsilon = 0.01$, $|f| \leq 3.5$

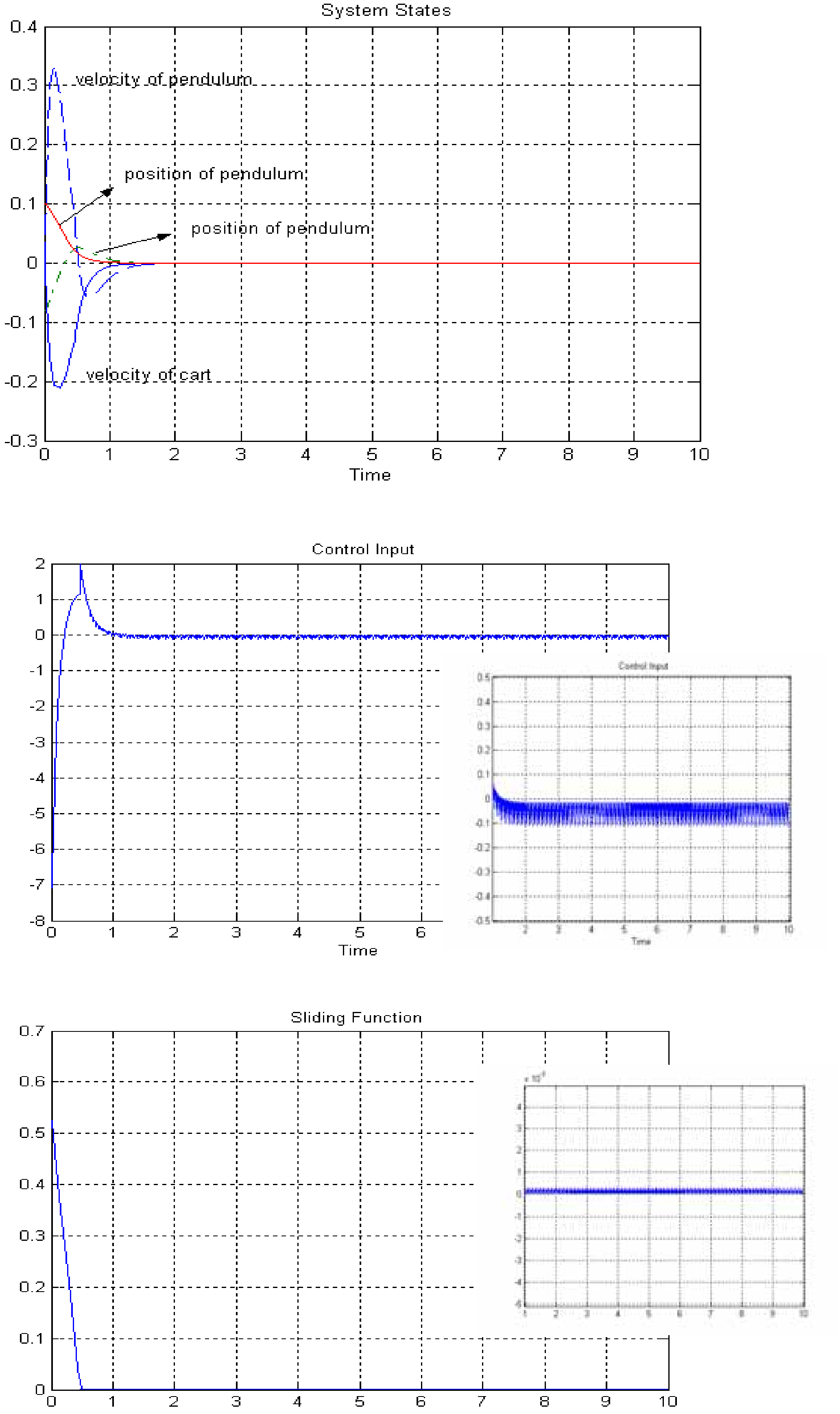


Fig-3.7 Simulation results of sliding mode

3.4 Implementation of Inverted Pendulum

After obtaining the mathematical model of inverted pendulum and simulating, the next thing is how to implement this system physically. The experimental inverted pendulum system is

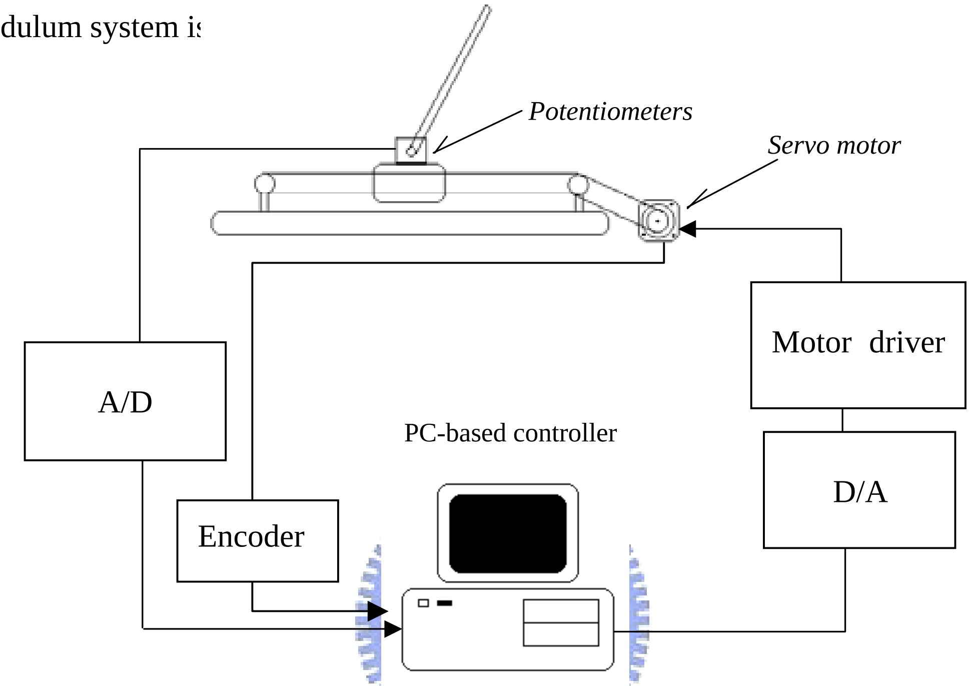


Fig-3.8 Experimental inverted pendulum system

This experimental inverted pendulum system consists an inverted pendulum system, a motor driver, ADA card, Encoder and a PC-based controller. The hardware and software used in the experimental inverted pendulum will be elaborated later.

3.4.1 Implementation Hardware

The inverted pendulum system consists of the following components, one pendulum whose length is 0.48m hinged to the cart, one cart moving on the rails whose length is 1.0m, potentiometer fixed to the hinges to measure the angles, and one DC motor with Encoder inside mounted on the pulley for providing force to the cart through a timing belt. The controller is implemented by a personal computer run Windows 98 operation system, which is used for both the implementation of controller and communication. The control input u of the system is taken as the input to the 12-bit D/A converter and the feedback is from the 12-bit A/D and Encoder as shown in Fig-3.8. The D/A converter is adopted PCI-1202 card that is 12 bit multifunction analog and digital I/O boards for the PC and compatible computers in a 5V PCI slot. As for A/D converter captured the data from potentiometer, it is adopted A-821PGH/L card that is 12 bit multifunction analog and digital I/O boards for the PC and compatible computers in an ISA slot. The A-821PGH/L and PCI-1202 multi-function card provides libraries for developing the data acquisition application. The libraries could be used or accessed some register on the card in our program to finish A/D and D/A conversion. The data from Encoder is measured by ENCODER3. The actuator is using a 300W and rated current 5.2A DC motor and provides sufficient force to control the inverted pendulum. The driver is using UT-80 produced by Micro Trend with two control way, current mode and voltage mode, and the way used here is current mode The controller program for transforming the input/output data is written in BCB (Borland C++ Builder) language. The sampling frequency is set at 1KHz by a programmable counter. The angle resolution of pendulum measured from the potentiometer through the A/D converter is 1.5×10^{-4} rad. The smallest difference between two commands from D/A is 2.7μA. The potentiometer needs connect to a power supply that provides voltage range 0~+10V.

3.4.2 Implementation Software

The controller is asked for versatile, intelligence and multifunction. A program running in personal computer to achieve these performances is written. In our experiment, BCB language is used to develop the software controller. The flowchart for the software controller is shown in Fig-3.9. The major steps of the software controller are concluded as:

Step1: Setting the controlled target x_d, θ_d

Step2: Measuring the cart position and the pendulum angle from the encoder and potentiometers

Step3: Taking the states into the selected controller to generate control force

Step4: Transforming the control force into continuous type via D/A converter

Step5: Amplifying the D/A out by UT-80 and drive the DC motor

Step6: Checking the work finished or not, if else, return to step2

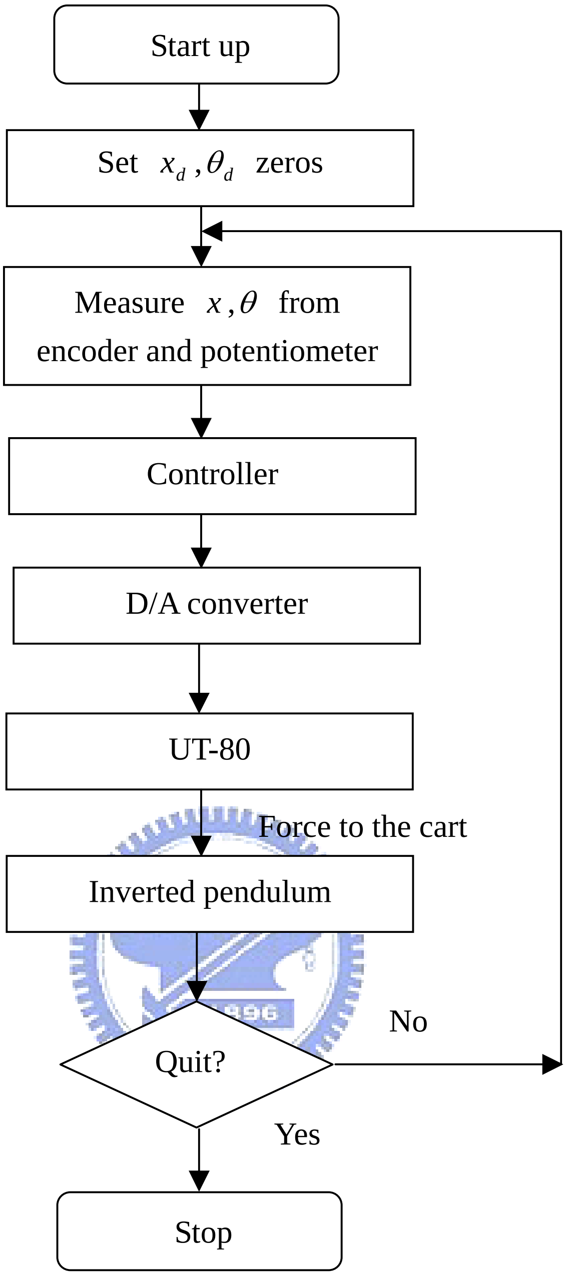


Fig-3.9 Flowchart for the software controller

3.5 System Identification

As the PC-based controllers of the experimental inverted pendulum system have been set up, the input force *u* will be calculated. However, the calculated force *u* could not transform D/A values directly as for using PC-based digitally controller. In

Fig-3.10, from the PC-based controller to the actuator are shown.

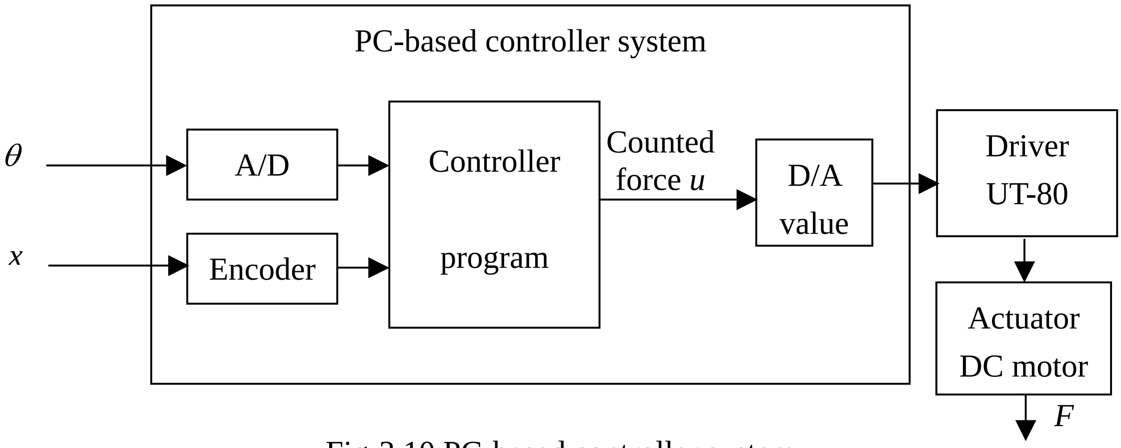


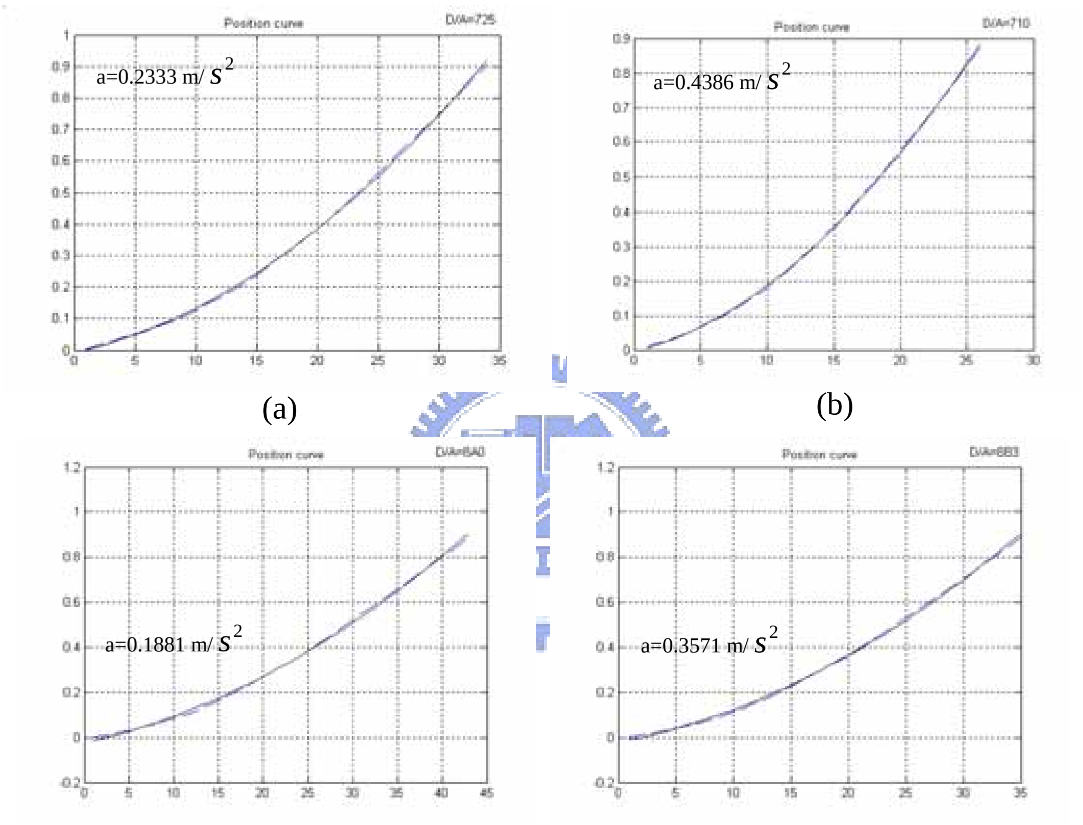
Fig-3.10 PC-based controller system

The states, θ and x, could be measured by potentiometer and encoder, and PC-based controller system could obtain the states using A/D card and Encoder card built in the PC. Then, the states are put to the controller program, and the force driven the DC motor will be calculated. But the output of PC-based controller system is D/A card, the relationship between D/A values and actual force driven to DC motor should be found. If the relationship is found, the force u counted by controller program could be transformed D/A value. As for UT-80, it acts an amplifier with two driving mode, current mode and voltage mode. Because the motor torque is direct proportion to the current generated from UT-80, the driving mode of UT-80 is current mode. The motion of the cart is straight-line motion, and the force from motor is uniform acceleration straight-line motion at beginning. The smallest difference between two commands of D/A output is 2.7 μ A, and UT-80 output is 3.125 mA. If the D/A value increase, the current of UT-80 driver will increase. As to increased current, the force generated DC motor will also increase. Due to direct proportion between current and motor torque and the cart with uniform acceleration straight-line motion, the relationship between current of UT-80 and the force *u* counted from the controller could be found and be written as *u* = *ki, k* is direct proportion value. The force *u* calculated is generated by DC motor, so the motor could be used to generate the straight-line force *F=ma* which is similar to *u*. The first thing of system identification is finding the direct proportion value *k*. Because the cart is uniform acceleration motion and the position values of the cart could be measured and stored, the time function of position could be fitted by using Least Square method. Assume the time function of position is $x(t)=\frac{1}{2}at^2+vt+x_0$, where t>0. The measured position values according to time changing could be obtained and the form of $x(t)=\frac{1}{2}at^2+vt+x_0$ could be changed to matrix form with different time and position, such as

$$\begin{bmatrix} 1 & t_1 & \frac{1}{2}t_1^2 \\ 1 & t_2 & \frac{1}{2}t_2^2 \\ \vdots & \vdots & \vdots \\ 1 & t_m & \frac{1}{2}t_m^2 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ v \\ a \end{bmatrix} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(m) \end{bmatrix}$$

, be simplified as $Ap=b$, where $0<t\le t_m$

By using Least Square method, the parameters *p* of $x(t)=\frac{1}{2}at^2+vt+x_0$ could be found, as $p=inv\left(A^TA\right)A^Tb$, and the acceleration *a* and the force *F=ma* could also be obtained. The fitted curve by real line and actual measured values by dotted line are shown in Fig-3.11. The two figures, (a) and (b) in Fig-3.11, are the position curves of the cart driving rightward with the different DA values and the different force. This shows that the two figures, (c) and (d) in Fig-3.11, are the position curves of the cart driving leftward with the different DA values and the different force.



(a) (b)

(c) (d)

Fig-3.11 Least Square curve and measured values

When different D/A commands are provided, the current for UT-80 and the force *F* generated by motor will be different. After giving a large number of current command and finding the relatively force, the direct proportion value *k* could also be fitted by Least Square method shown in Fig-3.12. However, the cart driving leftward and rightward will obtain the different direct proportion value *k* and static friction. After looking for the direct proportion value *k*, the force *u* counted by the controller could transform to D/A value in order to drive the DC motor. The higher resolution of D/A value, the force *F* more similar to the counted force *u*.

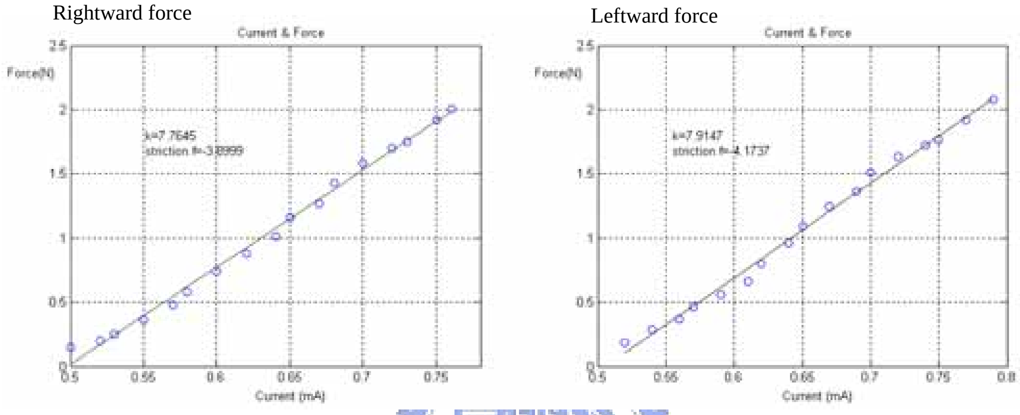


Fig-3.12 Relationship between D/A and force

3.6 Experimental Results

Here, the controller of inverted pendulum is use of PID controller in the actual experimental device. Using PID controller, the position of the pendulum could be controlled to locate at the upright position. As to the controller for the center position of the cart and the position of the pendulum at the upright position, the designed controllers are not introduced in this paper. There are some different cases of PID parameters, and the experiment resulted figures are shown. There are two cases using PD controllers, case1 is when $\boldsymbol{K}_{\mathrm{P}}$ =97.26, $\boldsymbol{K}_{\mathrm{D}}$ =13 and case2 is when $\boldsymbol{K}_{\mathrm{P}}$ =61.2, $\boldsymbol{K}_{\mathrm{D}}$

=11.02. The simulation for these two cases could find that the response time of the case1 is faster than the case2, the steady state error of the case2 with the friction is larger than the case2, and the control input of the case1 is greater than the case2. In the reality experimental results of these two cases, the above-mentioned situations could be found in Fig-3.13 and Fig-3.14. As to use the PD controller, the steady state error will exist when the inverted pendulum system has the friction. Thus, the PD controller could not control the pendulum at the upright position well. Looking for the resulted figures, there are greater noises in the actual device, so the waveform of pendulum position will cause ripples. The reasons causing the noises are the UT-80 driver, potentiometer, DC motor, the cart rail, the leather belt and so on. Some of the above reasons could use a good controller to overcome. Some of them are the problems made by the self of objects, so these noises are not easily to solve.

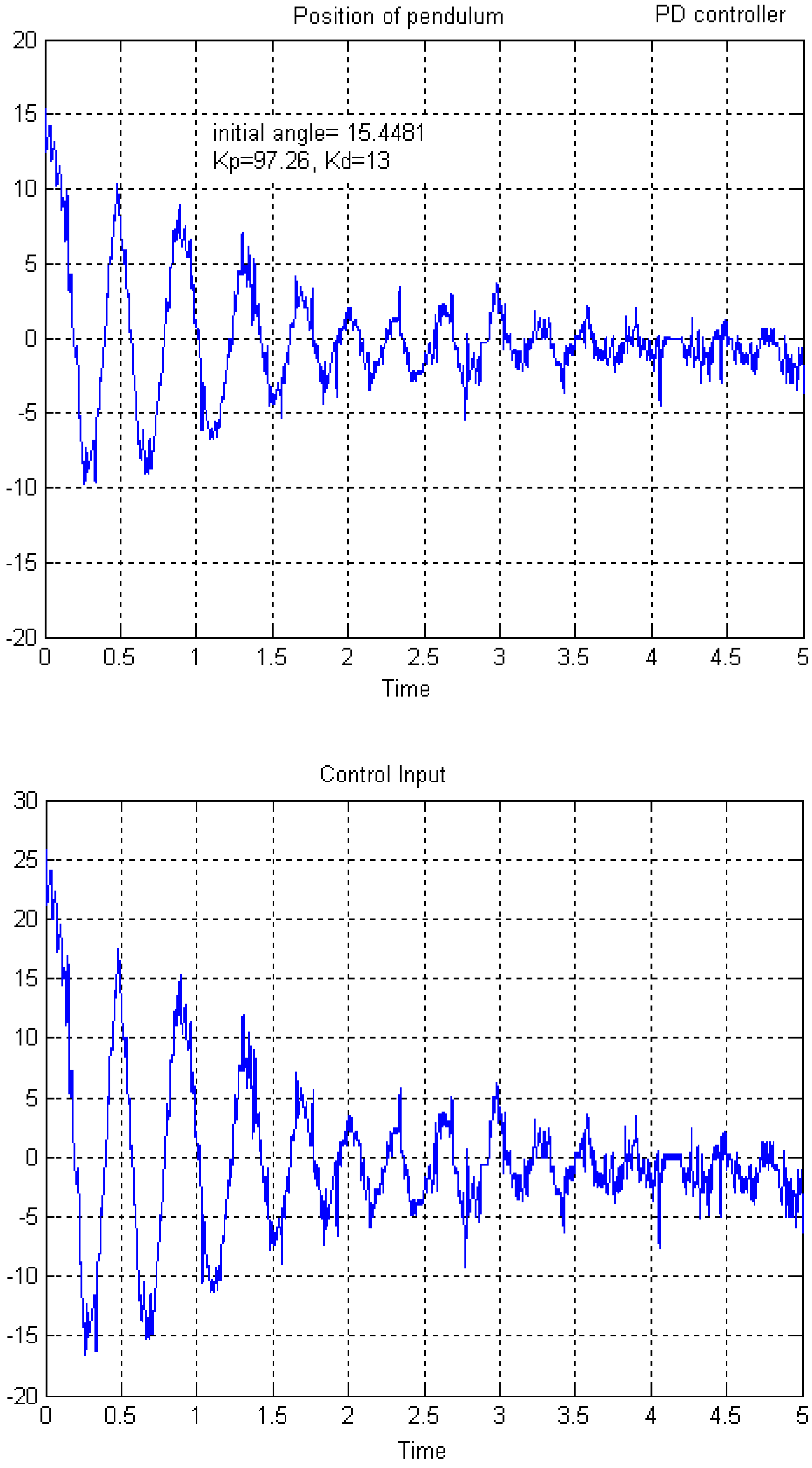


Fig-3.13 PD controller when $\boldsymbol{K}_P$ =97.26, $\boldsymbol{K}_D$ =13

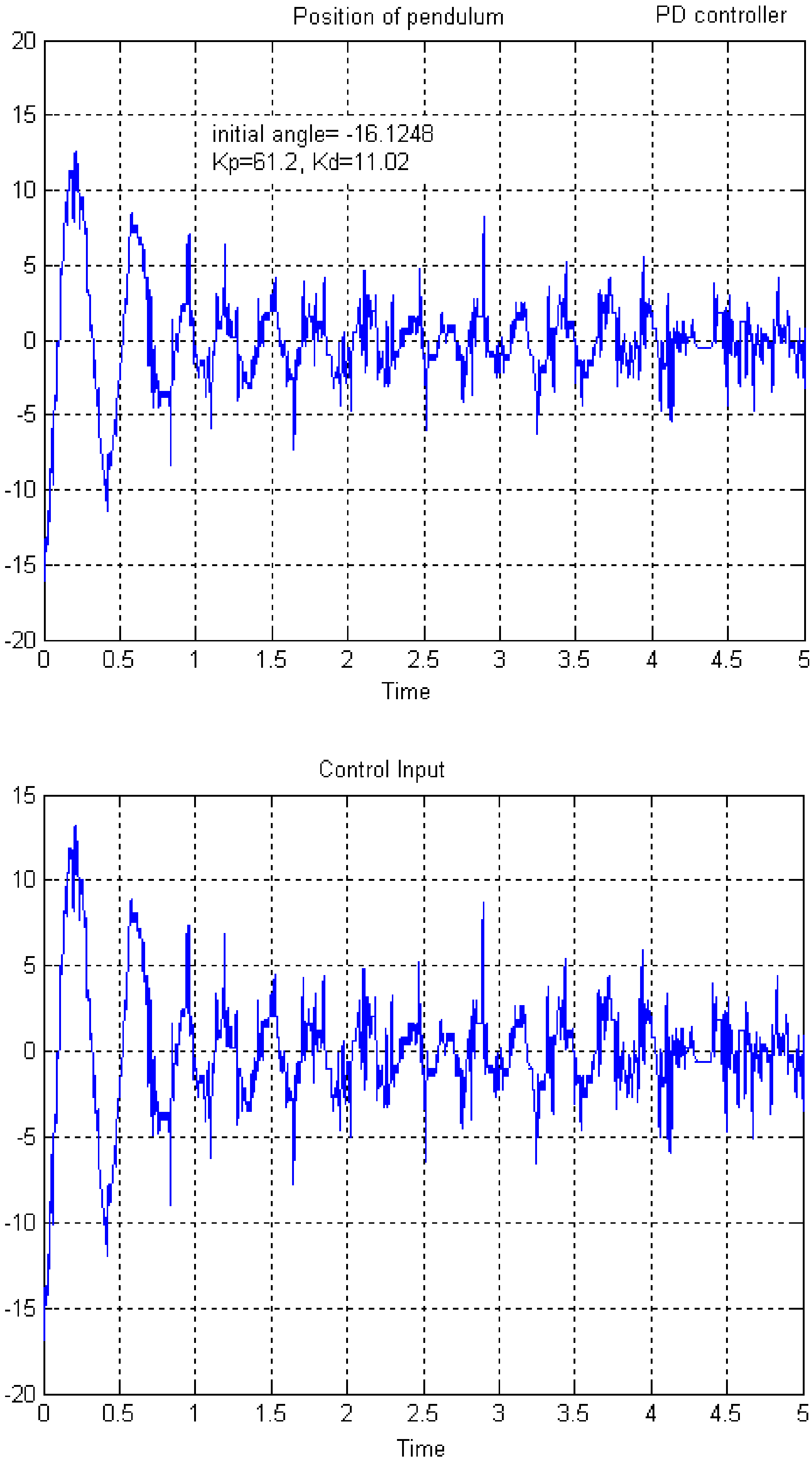


Fig-3.14 PD controller when $\boldsymbol{K}_{\mathrm{P}}$ =61.2, $\boldsymbol{K}_{\mathrm{D}}$ =11.02

Chapter 4 Experimental Devices via Internet

The all system architecture in this paper provide for the users from everywhere that could execute the experimental devices to verify the controllers designed at the users' own place via Internet. The principal objective in this paper is to establish an integral system in section-2.1 from the client side to the experimental devices through the server machine. As the communication protocol and the connection of the integral system established, the different kinds of experimental devices at everywhere could easily be added and expanded. Currently, inverted pendulum is an experimental device used to offer the users to execute online in this system.

4.1 Flow Chart

As inverted pendulum system provided for the client users to execute online, the all flow chart to operate the Internet-based control system is shown in Fig-4.1. The detail of this figure has been described before. The connected real lines of Fig-4.1 are meaning the processes when the client user sends the execution commands to the experimental devices in E-lab via Internet. The dot line (a) is meaning that sending the measured values to the sub server. The dot line (b) is meaning that sending the passing the measured values from the sub server to the main server. The dot line (c) is meaning that processing the measured values from the sub server and displaying to the client user.

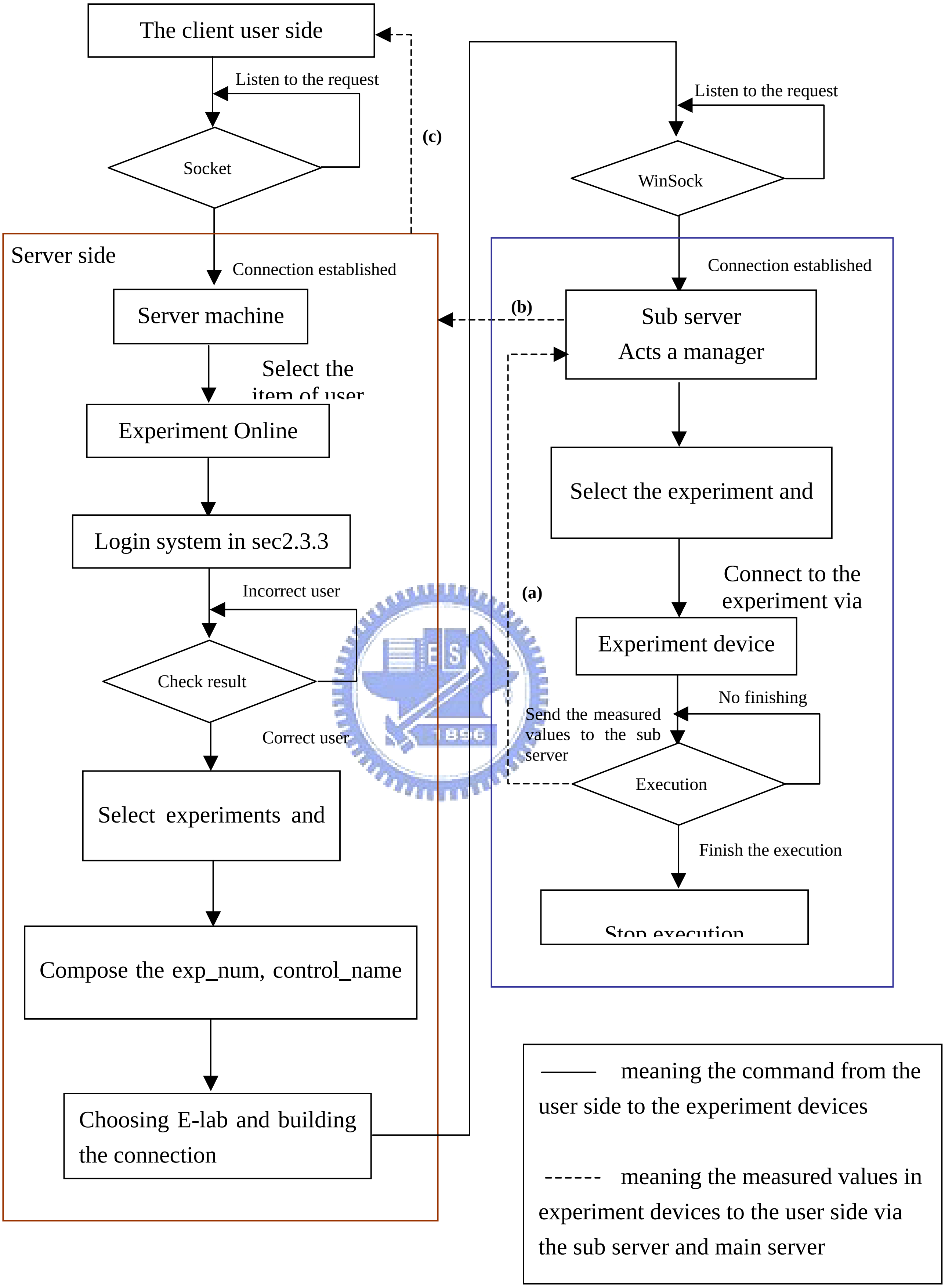
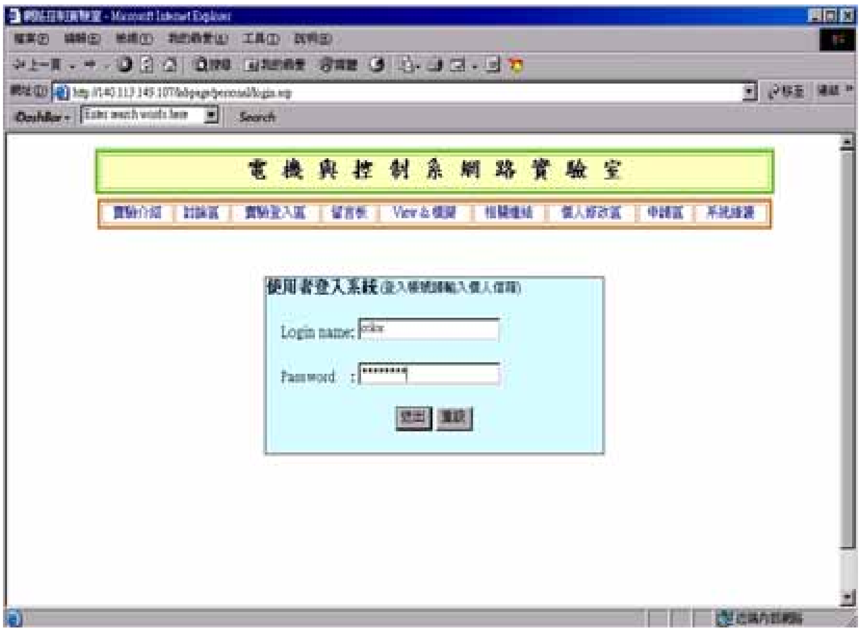


Fig-4.1 Flow chart of experiment execution via Internet

4.2 Execution Results

If the user would like to run the experimental devices online, the connection between the client side and the server side could be established first, and the experiment online item in the user interface should be chosen. When the connection is established and the item is chosen, the login system in Fig-4.2 will be shown.



電機與控制系網路實驗室

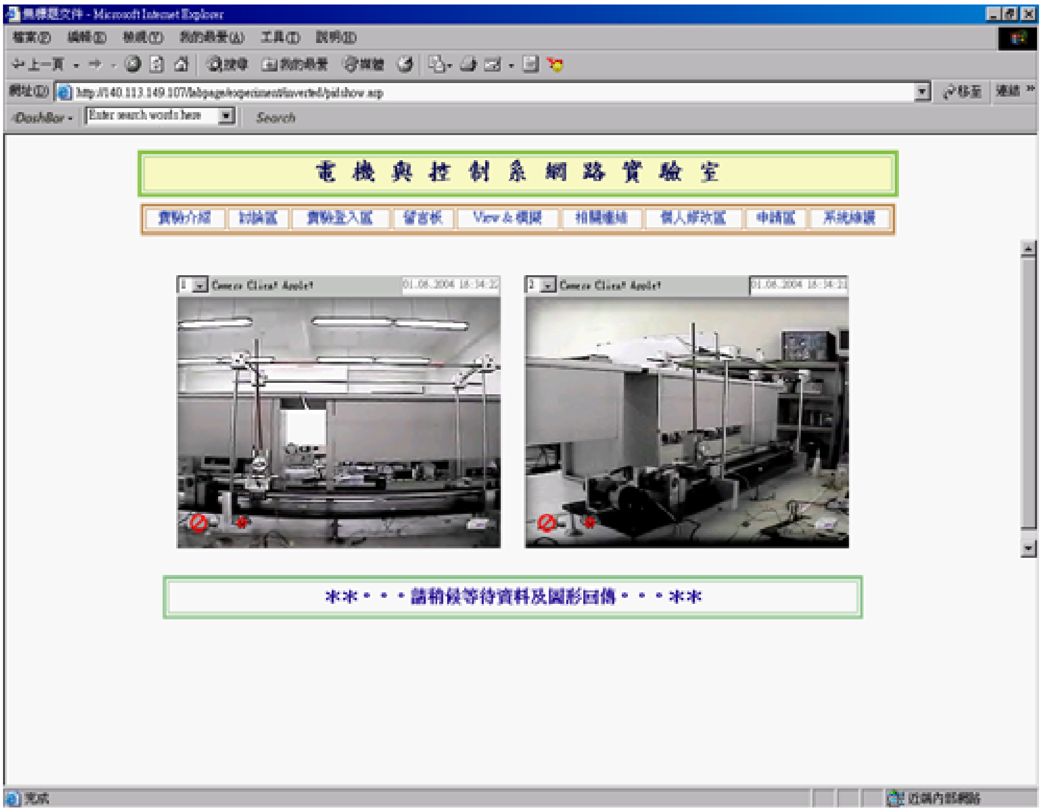
使用者登入系統

Login name: [illegible]

Password : ********

Fig-4.2 Login system interface

If the login user is correct, the experiment system will provide many kinds of the experimental devices, but now the experiment system just provides one experimental device, inverted pendulum. The operation interface of inverted pendulum is shown in Fig-2.13. As the designed parameters of PID controller are filled in and submitted, the views of running the experimental device will be shown in Fig-4.3, and the user has to wait for the resulted data and figure shown in Fig-4.4.

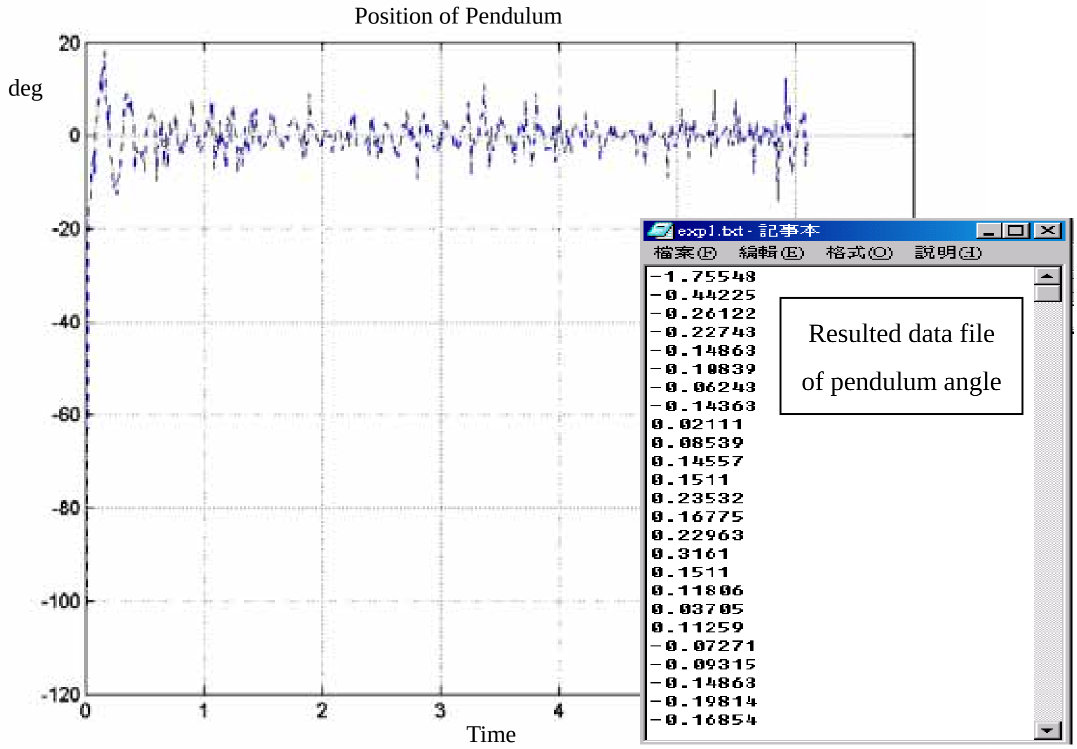


無標題文件 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://140.113.149.107/labpage/experiment/inverted/pidshow.asp
移至 連結
DashBar Enter search words here Search
實驗介紹 討論區 實驗登入區 留言板 View & 模擬 相關連結 個人修改區 申請區 系統維護
Camera Client Applet
01.08.2004 18:34:22
01.08.2004 18:34:21
完成
近端內部網路

電機與控制系網路實驗室

＊＊・・・請稍候等待資料及圖形回傳・・・＊＊

Fig-4.3 Views of running the experimental device



```
-1.75548
-0.44225
-0.26122
-0.22743
-0.14863
-0.10839
-0.06243
-0.14363
0.02111
0.08539
0.14557
0.1511
0.23532
0.16775
0.22963
0.3161
0.1511
0.11806
0.03705
0.11259
-0.07271
-0.09315
-0.14863
-0.19814
-0.16854
```

Resulted data file of pendulum angle

Fig-4.4 Resulted data and figures

Chapter 5 Conclusions

The overall objective of this thesis is to develop a surrounding where the users from everywhere could run the simulation of the designed controllers and execute the actual experiment online. The users in their own places could conveniently compare with the difference between their learned skills and factual devices, and verify their designed controllers workable or not via this approach. Using this construction, main server machine and E-labs of Internet-based laboratory, the integrated system could be easily expanded and set up no matter how the distance between the main server and E-labs is. In this integrated network system, the friendly user interface is supplied to the user facilely to operate without downloading or installing any software just working on IE, the actually experimental device, inverted pendulum, is provided for the user to operate, and video server and cameras are offered the user to view. Furthermore, the experiment results show that this integrated system works well and the inverted pendulum is controllable and executable in actual world. Finally, as following this construction of the integrated network system, the experimental devices could easily be added, and it could provide a boundless laboratory for everyone. Constructing this integrated network system, it could be either used to provide a surrounding for the users to verify the learned skills in the online education, or could be used to upgrade or modify the products as the user who buy them from the companies encounter some problems, and could also be used to control and monitor the electric equipments in the houses or in the factories as the master not at home or the manager not at the factories.

There are four primary future works presented below.

1. User interface of experiment online

Due to the user interface of experiment online, the controllers provided for the users to design are fixed input mode but could not change the designed parameters as the experimental devices running. It's could design a modulated interface using JAVA Applete. Therefore, the users could immediately observe the variation of executing experimental devices as adjusting the regulator.

2. MATLAB engine

As the users run the simulation online, there is a big problem happened. If the users' simulation codes let the system dissipated, the MATLAB engine may be down and then the other users won't run the their simulation online. This problem is concerned with MATLAB software self, so it could not easily be solved.

3. Inverted pendulum

In inverted pendulum, the execution resulted data and figures are not very perfect, because the noises from the potentiometer, actuator, the leather belt, the rail of the cart, driver and so on will make the system not to be control well. As for the driver and actuator, these two could be improved, but the potentiometer could change to encoder in order to reduce the noise. Here, the PC-based controller is used, but following the development of standalone system, the PC-based controller could be transformed into microprocessor controllers.

References

[1] Sven K. Esche, Constantin Chassapis, Jan W. Nazalewicz and Dennis J. Hromin, “A Scalable System Architecture for Remote Experimentation”, Frontiers in Education, 2002.

[2] Nitin Swamy, Ognjen Kuljaca, *Member*, *IEEE*, and Frank L. Lewis, *Fellow*, *IEEE*, “Internet-Based Educational Control Systems Lab Using NetMeeting”, IEEE TRANSACTIONS ON EDUCATION, VOL. 45, NO. 2, MAY 2002.Kin Yeung and Jie Huang, “Development of the Internet based control experiment”, Orlando, Florida USA, December 2001.

[4] C. C. Ko, Senior *Member*, *IEEE*, Ben M. Chen, *Member*, *IEEE*, Jianping Chen, Yuan Zhuang, and Kay Chen Tan, *Member*, *IEEE*, “Development of a Web-Based Laboratory for Control Experiments on a Coupled Tank Apparatus”, National University of Singapore, May 2000.

[5] J. Sanchez, F. Morilla, S. Dormido, J. Aranda and P. Ruiperez, “Virtual and Remote Control Labs Using Java: A Qualitative Approach”, IEEE Control System Magazine, 2002.

[6] Riko Safaric, *Member*, *IEEE*, Smiljan Sinjur, Borut Zalik, and Rob M. Parkin, *Member*, *IEEE*, “ Control of Robot Arm with Virtual Environment via the Internet”, Proceedings of the IEEE, 2003.

[7] Chiaming Yen, Wu-Jeng Li, and Jui-Cheng Lin, “A Web-Based, Collaborative, Computer-Aided Sequential Control Design Tool”, IEEE Control Systems Magazine, 2003.

[8] W. R. Stevens, “Internetworking with TCP/IP”, Vol III.

[9] 洪宗富, “Sliding mode controller design and implementation applied to inverted pendulum”, 國立交通大學電機與控制工程研究所碩士論文.

[10] 林嘉璋, “Teaching Auxiliary System Design on Control Via the World Wide Web”, 國立交通大學電機與控制工程研究所碩士論文.

[11] 陳永平, “可變結構控制設計”, 全華科技圖書股份有限公司,1999

[12] Charles L., Phillips H. and Troy Nagle, “Digital Control System Analysis and Design”, Third Edition.

[13] S H Yang, L S Tan, X Chen, “Requirements Specification and Architecture Design for Internet-based Control Systems”, Proceedings of the 26 th Annual International Computer Software and Applications Conference.

[14] Warren E. Dixon, *Member, IEEE*, Darren M. Dawson, *Senior Member, IEEE*, B. T. Costic, and Marcio S. de Queiroz, *Member, IEEE*, “A MATLAB-Based Control Systems Laboratory Experience for Undergraduate Students: Toward Standardization and Shared Resources”, IEEE TRANSACTIONS ON EDUCATION, AUGUST 2002.

[15] Eliane Guimarães, Antonio Maffeis, James Pereira, Bruno Russo, Eleri Cardozo, Marcel Bergerman, *Member, IEEE*, and Mauricio F. Magalhães, *Member, IEEE*, “REAL: A Virtual Laboratory for Mobile Robot Experiments”, IEEE TRANSACTIONS ON EDUCATION, 2003.

[16] Pasquale Arpaia, *Member, IEEE*, Aldo Baccigalupi, Felice Cennamo, and Pasquale Daponte, *Senior Member, IEEE*, “A Measurement Laboratory on Geographic Network for Remote Test Experiments”, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, 2000.

[17] Imad Elhajj, Ning Xi, *Member, IEEE*, Wai Keung Fung, *Member, IEEE*, Yun Hui Liu, *Member, IEEE*, Wen J. Li, *Associate Member, IEEE*, Tomoyuki Kaga, and Toshio Fukuda, *Fellow, IEEE*, “Haptic Information in Internet-Based Teleoperation”, IEEE/ASME TRANSACTIONS ON MECHATRONICS, 2001.

[18] Pedro Sánchez, Bárbara Alvarez, Andrés Iborra, José María Fernández-Meroño, and Juan A. Pastor, “Web-Based Activities Around a Digital Model Railroad Platform”, IEEE TRANSACTIONS ON EDUCATION, 2003.

[19] 黃能富, “區域網路與高速網路”，維科出版社.

[20] 范逸之，陳立元，“ Visual Basic 與 RS232 串列通訊控制”，文魁資訊股份有限公司，台北 2001.

[21] 范逸之，江文賢，陳立元, “C++ Builder 與 RS232 串列通訊控制”，文魁資訊股份有限公司，台北 2003.

[22] 白中和, “RS-232C 介面技術應用”，全華科技圖書股份有限公司

[23] 高弘毅, “RS-232C 入門與應用” 波心資訊股份有限公司