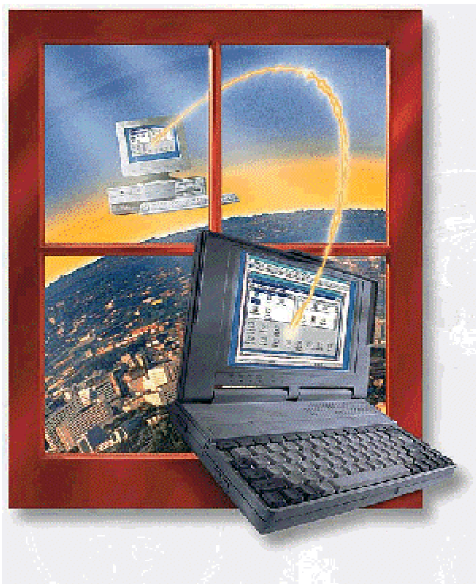
THE UNIVERSITY OF WESTERN AUSTRALIA

Department of Mechanical, Materials & Mechatronics Engineering

2001 Mechatronics Engineering Honours Thesis

Telelabs Project

Online Temperature Control Laboratory



Supervisor : Associate Professor James Trevelyan

Author : Harjono

November 2001

ABSTRACT

Laboratory classes are one of the most important elements in Engineering education. From the laboratory, students can apply their engineering knowledge learnt from the lectures to real engineering situations. However, laboratory equipment is usually expensive. As a result, engineering students have to share the equipment in the laboratory. This reduces the time students spend with the equipment and diminishes the purpose of the laboratory.

The Telelabs project was initiated in February 2000 and funded by the Faculty of Engineering and Mathematical Sciences at the University of Western Australia. The aim of the project is to develop technologies that enable laboratory equipment to be controlled remotely by students through the Internet.

The laboratory uses a streaming server to broadcast live video of the laboratory equipment. When students do their experiment online, they would get instant feedback of their control action by observing movements of the equipment from the video. This would give students a better feel of the laboratory

This thesis discusses how a laboratory equipment is designed and programmed, such that it can be controlled by students remotely online. The aim of the laboratory is to teach students simple feedback control principle.

The laboratory is divided into two parts. In the first part of the laboratory, students are introduced to the laboratory equipment and given the chance to play with the laboratory equipment for a short while. The second part of the laboratory is done online through the Internet. In this way, the use of laboratory equipment would be optimised and students learning experience would also be enhanced.

Harjono
69 Hardy Road
NEDLANDS WA 6009

The Executive Dean
Faculty of Engineering and Mathematical Sciences
University of Western Australia
Stirling Highway
CRAWLEY 6009

4th November 2001

Dear Sir,

I have great pleasure in presenting my Bachelor of Engineering Honours Thesis entitled "Telelabs Project: Online Temperature Control laboratory". I trust that this document fulfils the requirements of the honours dissertation, and the results are beneficial to the University and all those who study at UWA in the future.

Yours sincerely,

Harjono

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Associate Professor James Trevelyan for all the guidance and support he had given me throughout the year. I started the project with little knowledge of how a control system is built, let alone putting a laboratory on the Internet. At the end of the project, I managed to build the whole control system and successfully putting the laboratory remotely online.

I would like to thank the Telelabs team members, in particular: Jan Baranski, Flavio Bruni, Dr. Alex C. Le Dain, Jamie Lee and Colin Leung for all the guidance and help they provided me during the duration of the project. They also had contributed a lot of inputs to this project, especially Jan, Flavio and Alex.

Furthermore, I would like to thank Sabbia Tilli for all the trouble she went through in proof reading my thesis. Although Sabbia is not officially the team member of the Telelabs project, her contribution to the Telelabs project is very significant.

I would also like to thank the workshop technicians for providing me with their brilliant design idea of the rig and helpful feedback.

A special thanks to Olivia Widjaya, whose precious advice and encouragement has pushed me through the tough times.

Most importantly, I would like to express my deepest gratitude to my beloved parents and brothers who have supported me throughout my years at the university.

Table of Content

List of Appendices

Table of Figures

Nomenclature

e_j	The difference between desired temperature and actual temperature of the control system (error).
e_j^*	The sum of error with respect to time.
OS	Operating System.
VI	Virtual Instrument
I/O	Input / Output
IP	Internet Protocol.
TCP	Transmission Control Protocol.
UDP	Universal Datagram Protocol
RTD	Resistance Temperature Detectors.
SSR	Solid State Relay.
PID	Proportional, Integral and Derivative.
PWM	Pulse Width Modulation.
LOL	Labs On Line.
δt, dt	Time step used in PID controller.
Http	Hyper Text Transfer Protocol
T Set	The desired temperature for the control system.
T Top	Temperature of the electric iron at the top part of the plate.
T Bottom	Temperature of the electric iron at the bottom part of the plate.
T Thermostat	Temperature near the thermostat of the electric iron
T Ambient	Temperature of the surrounding area of the electric iron rig.
Temp. Setting	Temperature knob setting of the electric iron.
LabVIEW	Laboratory Virtual Instrument Engineering Workbench.

1 INTRODUCTION

Information technology is advancing at a tremendous pace, especially since the invention of the World Wide Web. By touching a button on a keyboard a person can retrieve information, send electronic mails and even have a live conference with another person across the world. With this power to communicate, universities around the world are adapting their teaching methods to include computer and Internet technology, such as teleteaching [Filler et al, 2000].

Teleteaching is a method of teaching through the Internet either by presenting on the Internet some lecture materials, laboratory simulations, real-time laboratory or online tutorial exercises.

1.1 Motivation

There are many benefits from teleteaching compared to the conventional methods. The most significant one is geographical independence [Filler et al, 2000]. That is students have the convenience of accessing all relevant lecture material, submitting assignments and completing assessment work online without physically being at the university. It has also been shown that when students have control over when and where the learning takes place, they actually learn better [Barker, 1998]. The less obvious benefit of teleteaching would be the reduction of lecturers' and students' time and cost of travelling to-and-from universities. This gives them more time to concentrate on more important tasks, such as improving teaching and learning. Other indirect benefits from teleteaching are the reduction of overhead costs, such as paper and printing cost, labour cost and reduction of traffic and pollution levels since there would be fewer cars on the road as students travel less to university.

Teleteaching consists of two main streams: virtual learning and virtual laboratory. The former is where an educational institution posts lecture materials, such as power point clips, video clips and tutorial exercises on the Internet. Students can read, download or complete the tutorial exercises in the convenience of their own home. The virtual laboratory is "a media-rich interactive environment of sufficient fidelity for conducting experimental activities associated commonly with some physical laboratory" [Rafe et al, 1999].

One example of virtual learning is the First-year Engineering Dynamics web-based tutorial at the University of Western Australia [Stone & Scott, 1997]. Students in the university were required to solve engineering problem. If a wrong answer has been provided, the computer system would attempt to give specific diagnosis feedback, rather than a simple 'right/wrong' message.

On the other hand, Virtual laboratories can be of two types: Real time operation or simulation. The former involves manipulating real equipment or matters from afar through the Internet, whereas the latter is a computer system that studies or simulates the behaviour of the real experiment [González-Castaño et al, 2001].

1.2 Objectives

The objective of this thesis was to create a virtual laboratory contains a simple feedback control system. A domestic electric iron can demonstrate the feedback control. It can be controlled by students remotely through the Internet. To achieve this, it was necessary to:

1. *Build a control system.* The control system consists of an electric iron with sensors and several pneumatic actuators.
2. *Program the local Controller of the rig using LabVIEW.* The local controller has the ability to implement PID controller into the control system. It can also utilise two types of temperature control: thermostat and Pulse Width Modulation.
3. *Incorporate the local controller program into the Telelabs system.* The program for local controller was integrated into the Telelabs system. This enabled the rig to be controlled remotely using the Internet.

1.3 Telelabs Project History

- In 1999, the initial concept of the Telelabs project was devised by Associate Professor James Trevelyan at the University of Western Australia.
- In 2000, Jan Baranski was commissioned to conduct design study to identify the key technical aspects of the Telelabs concept, research the relevant, commercially available hardware and software and evaluate their suitability in terms of capabilities, limitations and cost.
- November 2000- February 2001, a student team was formed to work on the project.
    - *Two degree of freedom Laboratory*. The rig is used in the unit: Dynamics 331 taught by Professor Brian Stone. The aim of the rig is to teach students two degree of freedom vibration. Two students, Jamie Lee and Colin Leung were given the task to develop the first prototype of the Telelabs system.
    - *Electric iron rig*. The author was given the task to design and build the electric iron rig.
    - *Sand weighing machine reworked*. A second year Mechatronics student, Flavio Bruni was given the task to mechanically redesign the rig for online use.
- May 2001, full specification of the Telelabs system was developed by Associate Professor James Trevelyan.
- May – October 2001, Dr. Alex C. Le Dain was responsible for engineering of the server and core software of the Telelabs system.
- March – October 2001, the author developed the electric iron laboratory. The scope included local controller and incorporation of the local controller into the core software of the Telelabs system.

1.4 Layout of the thesis

This thesis details the implementation of the electric iron rig to demonstrate simple feedback control system. It also shows how the rig was incorporated into the online laboratories of the Telelabs system at the University of Western Australia.

The first two chapters of the thesis provide the background and explain why an online laboratory is useful as part of the education system. These chapters also detail the similar work undertaken by other institutions.

Chapter 3 of this thesis demonstrates how the electric iron rig was designed, such that it can be used to demonstrate simple feedback control principle. It also illustrates the components of the control system, such as sensors and actuators and illustrates how each component interacts with the controller, in this case a computer.

Chapter 4 starts by introducing the software used to program the local controller of the feedback control system and why it had been chosen for the project. It goes on to discuss how the controller was programmed, such that it can implement PID controller and Pulse Width Modulation on the temperature control. Some programming issues are raised in this chapter. This chapter ends by briefly introducing other software that had been used in this project, namely the streaming server for visual effect and simulation program.

Chapter 5 details the components of the Telelabs system and how local controller of the electric iron rig was incorporated into the Telelabs system. The roles of each component and how they interact with each other are explained in here. This chapter also illustrates how a user interacts with the laboratory equipment through the Internet.

Chapter 6 describes how the electric iron rig was used as a teaching tool. This chapter also describes the result and student feedback on the system.

The final chapter summarises the achievement of this project, presents outlook for any future work and draws conclusions.

2 LITERATURE REVIEW

A number of papers have been dedicated to online laboratories. However, most lacked depth in that they only provided a general overview of the laboratory[1]. In most of this research, the benefits of online laboratories are briefly mentioned without supporting evidence, such as surveys or student performance assessments. One possible reason is that online laboratories have not been used long enough to measure benefits.

Most researchers cite convenience and cost factors as the favourable reasons behind these laboratories. The other reason as stated by Layne & Beugelsdijk[1998] is that an online laboratory could accelerate the work on challenging scientific problems throughout the world by encouraging collaboration across local, national or international boundaries.

Burrell et al [1999] carried out a comprehensive evaluation to compare the use of actual laboratory (Wet Lab)[2] and simulation of actual laboratory online (Web Lab). The students were asked to carry out the experiments, collect data in both environments (Web and Wet), and choose one of the sets of data to use for a report. Students collaborated in using the Wet Lab experiment but worked alone in the Web Lab. They performed data analysis, graph preparation, and report writing individually also.

The effectiveness of the Web Lab was evaluated by surveying the students who completed their lab reports. The data were arranged in a contingency table[3] and analysed using several statistical methods. The criteria analysed were: preferred environment for gathering data, ease of writing reports, helpfulness of Web Lab and Wet Lab elements for writing Lab report, ease of running the experiments in both environments and usefulness as tools and time required. It was found that the Web Lab

[1] Lemckert, C. J., Florance, J. R., 1997 & Layne, S. P., Beugelsdijk, T. J., 1998

[2] The terms: Wet Lab and Web lab are used respectively in the paper to describe the actual experiment and experiment on the Internet.

could serve the experiment objectives as well as the Wet Lab. It was also found that students preferred the Web Lab because of ease in performing the experiment, speed and convenience.

Several papers also discussed the reason for using real-time process laboratory over the simulation. Although it is possible to use simulation to teach many practical skills to students, there exist several situations where the use of real equipment by student is necessary: either the development of a simulator from scratch is not feasible or real industry equipment is too complex to simulate [González-Castaño et al, 2001].

The advantages of a real-time process laboratory over simulations are:

1. Data collected is genuine,
2. It provides access to real apparatus which may enhance understanding,
3. It readily demonstrates the constraints of real-time,
4. It provides a sense of 'immediacy'
5. Its use enforces scheduling in that students are required to make commitments and keep them – thereby supporting the development of responsibility,
6. It can support the development of confidence to use real apparatus,

[Lemckert & Florance, 1997]

The University of Tennessee at Chattanooga [Henry, 2000] is one of the pioneers in real-time process laboratories. Its remote laboratory is set up using a main server, which runs Microsoft Windows NT and five client machines that run LabVIEW 5.0 from National Instruments.

When an experiment is run, the client machines pick up the command files produced by LabVIEW WEB server on the server side. The client software reads and converts these command files into operations that are understood by the DAQ cards to control the physical system [Henry, 2000].

[3] A Contingency table analysis evaluates whether two variables, measured on a nominal level, are independent or dependent of one another.

Stevens Institute of Technology in New York [Esche et al, 2000] also offers its students online laboratories. The initial design was such that the laboratory can be accessed exclusively in remote fashion. However, due to the constraint placed by the network it was decided that a hybrid on-site / remote was utilised instead.

During an experiment, students will interactively submit sets of input data for the experiment to a first computer. This computer controls a waiting queue and communicates the experimental results between the server and the clients. A second computer will extract the data sets one by one from the waiting queue and initiate their processing to the actual laboratory equipment. A video camera monitors and records the whole process in electronic form as audio and/or video files. Both the results of the experiments and the audio/video files will be made accessible to students through the Internet.

This method is claimed to offer several benefits:

- If several users entered identical sets of input data during the remote-experiment, they can be combined into one actual experiment run and its result can then be sent to all the users,
- It reduces the strain on a laboratory schedule,
- It is a much closer approximation to hands-on experience compared to a simulation,
- It captures the spirit and imagination of students who tend to be increasingly technologically inclined, and
- It promotes self-learning.

At the Swiss Federal Institute of Technology [Salzmann et al, 1999], the online laboratory is implemented slightly differently from the two institutions above. The local server is a computer located near the equipment. It is equipped with the hardware interface required to communicate with the sensors and the actuators. The sensors act as devices that capture measurements from the process and the actuators implement actions on the process. A video camera and microphone are available as additional sensors to provide visual and audio sensorial experience. The server software is responsible for receiving commands from the client through the Internet, transmits them to the real process and returning variables that define the state of the physical process to the clients. Video images and audio are also transmitted to the clients through the server. Figure 1 shows the interface designed for carrying out experiments on a servomechanism

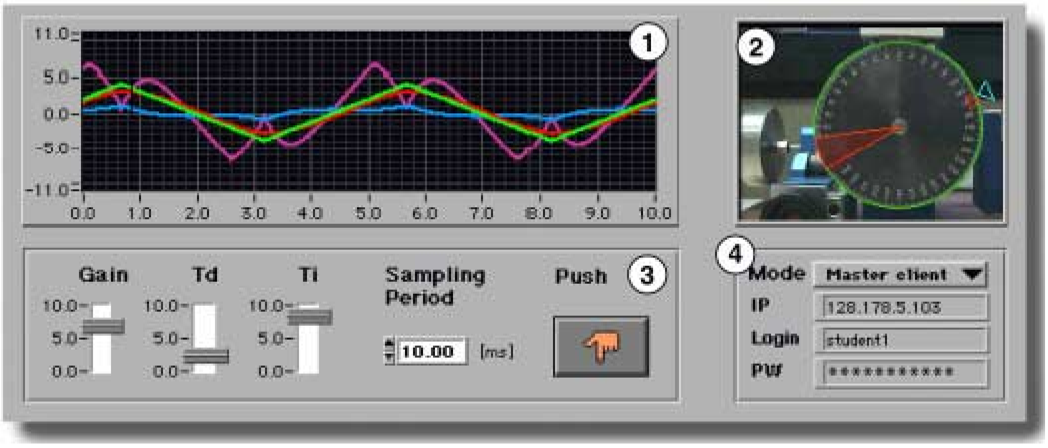


Figure 1. Learner interface for the Online Laboratory at the Swiss Federal Institute of Technology

[Source: Salzmann et al, 1999].

The learner interface has four distinct areas: (1) the scope area, (2) the visual area, (3) the parameters area, and (4) the administration area. The scope area enables the user to follow the time evolution of all signals relevant to the experiment. The visual area provides a video feedback of the real process enhanced with a virtual representation of the process. The parameter area allows the user to make adjustments to the physical process through the specification of appropriate values of adjustable parameters. The administrative area allows the management of the different connection stages such as user login and logout from the session, for example.

The remote client is a computer equipped with the functionality to observe and act on the physical process. The control software is a Virtual Instrumentation (VI) built using LabVIEW and compiled for the target platforms. This VI provides a complete interface between the user’s computer and the real process.

Another institution that offers an online laboratory is the department of Technology and communication at the University of Vigo, Spain [González-Castaño, et al., 2001]. The online laboratory is provided for the computer architecture students at the university. It aims to emulate a communication between a single board microprocessor, SBC68K and the user PC, such that students feel like working on PCs that are connected to a SBC68K board.

Instead of using the client / server distributed system as in other institution, it uses a concept of object distribution using CORBA where each resource is an object that is identified uniformly all over the distributed environment. All objects can move to different locations without changing their identities and behaviour. These objects are grouped into several functional modules: connection module, disconnection module, inactivity control module, terminal emulation module, remote files access module, and remote button module.

The benefit of this system was evaluated by comparing costs between conventional laboratory, low / medium availability virtual laboratory with a small number of equipment and high availability virtual laboratory with a large number of equipment (100% availability and no waiting time). It was found that in the low / medium availability case, there was a saving of 84% in initial cost; 50% saving in fixed cost (instructors) and 84% saving in fixed cost (materials). Whereas for the high availability case, there was a saving of 55% in initial cost; 10% saving in fixed cost (instructors) and 55% saving in fixed cost (materials).

2.1 Telelabs project at the University of Western Australia

The Telelabs project was initiated by Associate Professor James Trevelyan at the University of Western Australia in February 2000 and scheduled to be completed in February 2002. The aim of the project is to develop technologies that enable students to control laboratory equipment remotely in a cost-effective manner. Hence, students could have more time to work with the equipment and improve their learning experience.

It was observed that most of the laboratories described above use the hybrid on-site method as described in the Stevens Institute of Technology example. This method uses batch commands, where all the commands and data are entered into the program before running the experiment. After that the user starts the experiment and wait for the results. Although this method reduces the strain placed on the network, the experiment would become less interactive and students may find the process less interesting.

With the improvements in technology over the recent years, high-speed internet products such as cable connection have become more affordable to public usage. The usage of cable connection can greatly alleviate the constraint placed on the network. Therefore, the Telelabs project focuses on making online experiments more interactive and interesting to students so they can obtain instant feedback on their control actions during the laboratory. A streaming server system is also employed where the real time movie of the equipment is streamed to the students. Therefore, the students would be able to see the movement of the equipment during the experiment, thus enhance their learning process. While a student uses the equipment, other students can still watch the process from the server. The only difference is that those students would not be able to control the equipment.

To further reduce the strain placed on the network and equipment, the students using the equipment are divided into groups. In order to do the experiment, the students would have to login into the system and queue for the hardware. When one group member obtains access to the laboratory, all other members the group already in queue are able to operate the experiment at the same time. The equipment is more effectively utilised in this way since more than one students can use it. This process also promotes co-ordination and co-operation between students since all the students in the same group would have to communicate with each other. The students can communicate with the built-in chat function in the Telelabs system. The students can discuss how to complete the experiment, determine which students should have control of the equipment at a particular time. Therefore, the experiment would have “similar” feel to the actual experiment, except students are not physically present in the laboratory.

The structure of the Telelabs system is very similar to that at the Swiss Federal Institute of Technology. Each laboratory equipment is administered by a high performance host PC, which controls the hardware with the FieldPoint I/O devices. One server machine is utilised to handle the connection between the client machine and the host PC via the Internet and the local area network (see Figure 2).

The Telelabs system consists of three main pieces of software: the LOL server, hardware master and Client master. The software on the host PC called the client master sends a control or read signal to the LOL server. The LOL server detects whether the hardware master is ready. After that, it reads or sends a control setting to the hardware master, which is located in the host PC. The hardware master receives the commands from the LOL server and makes a connection to the FieldPoint module to acquire inputs and send outputs.

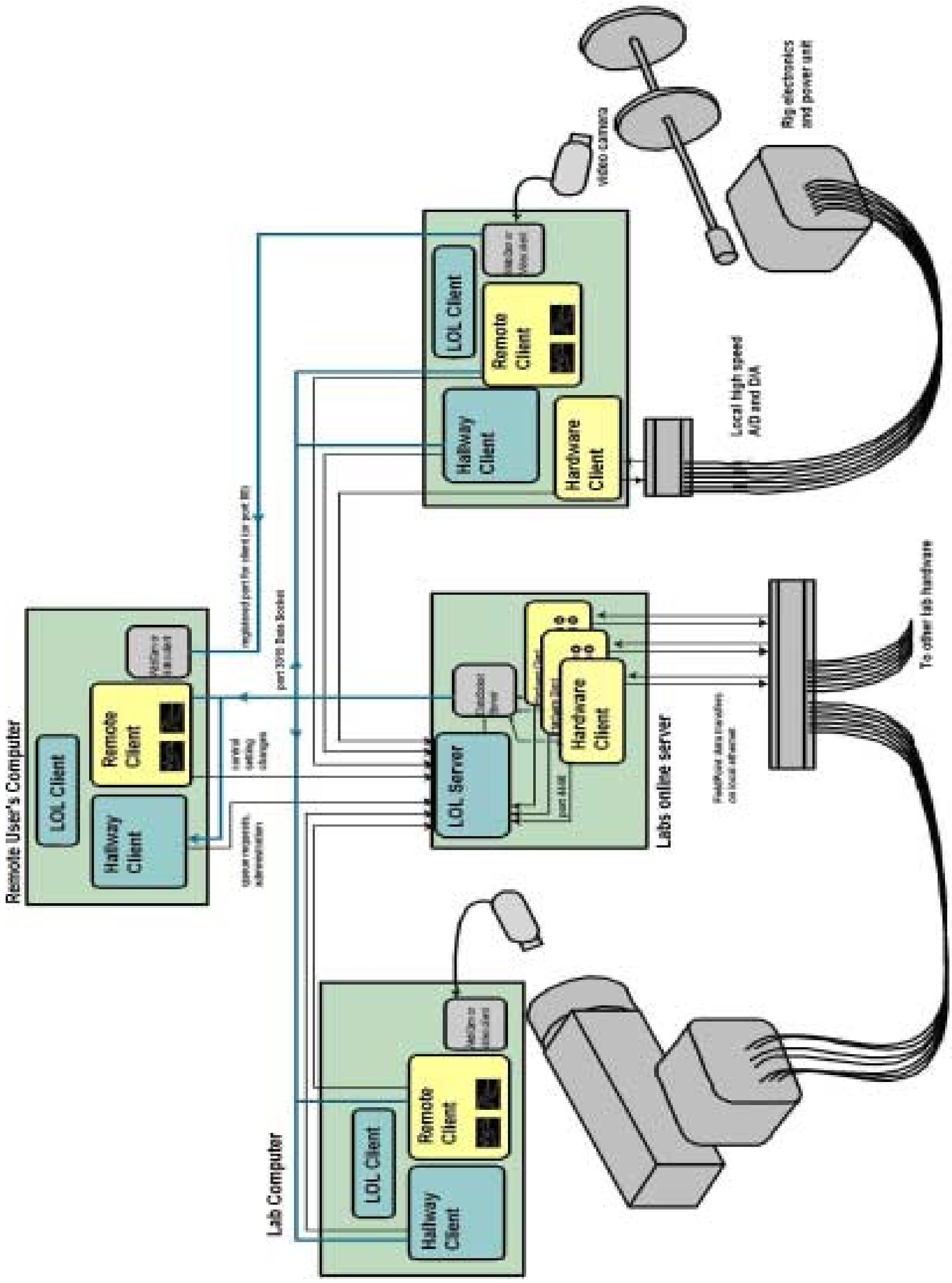


Figure 2 Structure of the Telelabs

3 HARDWARE

This chapter demonstrates how the electric iron rig was designed, such that it can be used to demonstrate simple feedback control principles. It also illustrates the components of the control system, such as sensors and actuators and illustrates how each component interacts with the controller, in this case a computer.

3.1 The electric iron rig

The aim of electric iron rig was to demonstrate simple feedback control system and to show that even the simplest feedback control system could have some interesting complication. The layout of the electric iron rig can be shown in Figure 3.

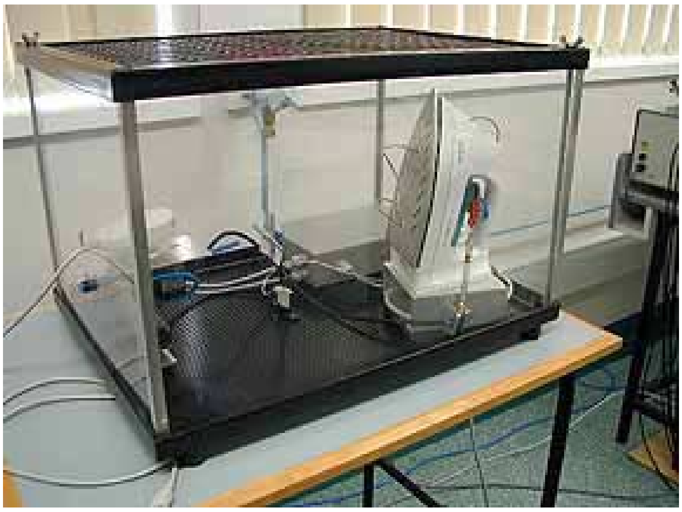


Figure 3 The layout of Electric Iron Rig

Design consideration of the electric iron included safety, visibility of the equipment and heat transfer between the rig and its surrounding. The Safety factor is the most important aspect. In order for the rig to be used unsupervised, the electric iron had been enclosed such that potential injuries could be avoided and an excessive amount of heat from the electric iron could be dissipated to the surrounding environment. Furthermore, a notice had been put up to prevent people from putting any object or paper on the top of the enclosure. The material of the enclosure also had good visibility, which allowed students to observe the control system during the experiment if the laboratory were done locally. Therefore, the enclosure was built from 3.5 mm Perspex with the dimension of 550mm x 400mm x 350mm. Furthermore, the top and bottom of the enclosure were made out of perforated metal sheet to enable hot air circulation (see Appendix A1).

The rig was equipped with three high-temperature sensors that measure the temperature of the iron at the top plate, bottom plate and thermostat of the electric iron. Another low temperature sensor was positioned in the middle of the rig to measure the ambient temperature of the rig.

The main power of the electric iron rig is switched on and off using a semiconductor switch to allow for rapid switching action. The temperature knob of the electric iron is turned using a pneumatic cylinder and solenoid valve. This allows the electric iron to have two temperature settings: low and high setting. Another solenoid valve was used to control the cool air inlet that blows onto the surface of the iron plate. The cool air plays two roles in the system: acts as a cooling medium and acts as a disturbance to the control system.

3.2 Sensors

3.2.1 RTD and Transmitter

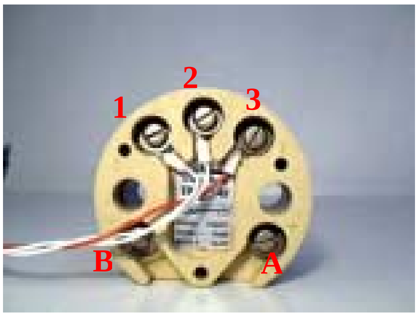


Figure 4. Temperature transmitter

The temperature of the electric iron was measured using Resistance Temperature Detectors (RTD). The RTDs are made from platinum, which have linear resistance-temperature characteristic.

A temperature transmitter (Figure 4) measures the resistance changes in the RTD. It then processes these resistances and converts them into an electric current of between 4-20 mA. The temperature can then be determined from the linear relationship between the output electric current and Temperature. For the transmitter to work properly, it needs a DC supply of 11.6 to 40V or a nominal voltage of 24V.

- Terminal 1,2 and 3 are wired to a RTD sensor. The resistance changes are measured from these terminals
- Terminal A is connected to the positive terminal of a DC 24V power supply.
- Terminal B is the output terminal for electric current of 4-20 mA.

Two types of temperature transmitters were utilised in this application: TR 48 and TR 20. The measurement range for TR 48 is between 0^{o}C and 300^{o}C, whereas TR 20 can measure temperature between 0^{o}C and 50^{o}C. The calibrations for both transmitters are shown in Table 1.

Transmitter type	**Calibration**
TR 48	$T = (i - 0.004)x18750$
TR 20	$T = (i - 0.004)x3125$

Table 1 Calibration procedure for the temperature transmitters

Note: i is the electric current in mA.

3.3 Actuator

3.3.1 Solid state Relay

Figure 5. Solid state relay

Rapid switching action in the main power of electric iron rig was implemented using a solid state relay. The solid state relay (SSR) can switch the power on the electric iron “on” and “off” in a very short period of time (20 μsec). This is the basic principle of Pulse Width Modulation.

The input used to control the SSR in this application is DC Voltage. By applying a voltage between 3-32 Vdc, the SSR is turned on and it is turned off when a voltage below 3 Vdc is applied.

- Terminal 1 and 2 are the terminals for the main AC 240V power.
- Terminal 3 and 4 are the input terminals used to turn the SSR “on” and “off”.

3.3.2 Air Cylinder

Figure 6. Pneumatic Cylinder

A spring return air cylinder was utilised (SMC CD J2 L6- 45SR) to change the thermostat knob position of the electric iron. Two-temperature settings: Low and High Temperature are achieved by using this air cylinder. A solenoid valve controls the air supply to the cylinder. When the solenoid valve is turned on, the air drives the air cylinder that pushes the electric iron’s temperature knob (high temperature). When the solenoid is turned off, no more air is in the actuator and the spring in the air cylinder pulls the shaft back to the original position (low temperature). The diagram for this mechanism is shown in Figure 7.

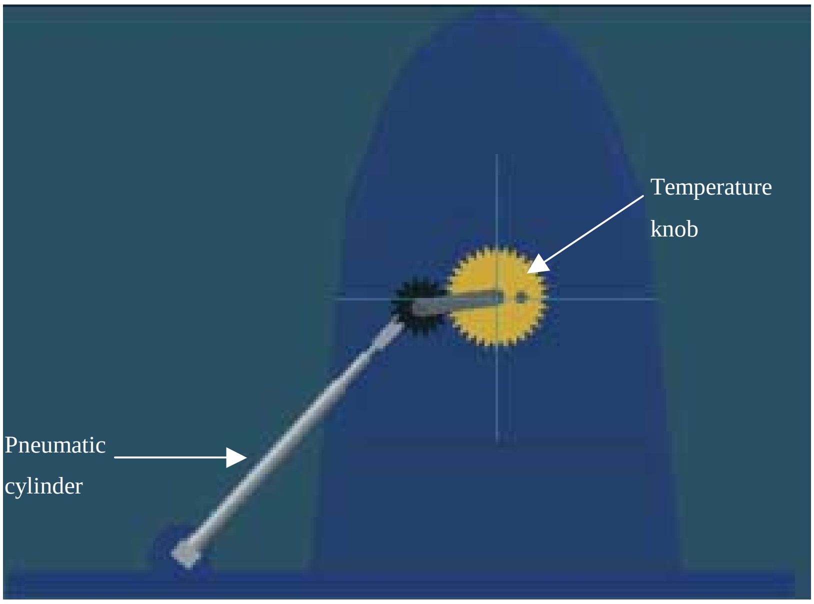


Figure 7 The mechanism to push the electric iron knob

3.4 FieldPoint module

The communication from hardware, such as the sensors, solenoid valves and SSR to the server computer was done using FieldPoint module. This module has a multiple I/O port to read the analogue signals from the RTD sensors and to send out discrete signal to the solenoid valves and SSR to turn them on and off (see Appendix A3).

FieldPoint Distributed I/O offers some advantages over a conventional Programmable Logic Controller and conventional Plug-In type data acquisition cards in that it can be easily set up and programmed over a network by configuration software. The other advantage of the FieldPoint module over the other data acquisition cards is its network capability. This capability allows any computer to control the hardware, even when that particular computer is far away from the hardware. FieldPoint module is also independent from the server computer. Therefore, any upgrade to the computer hardware on the server computer can be easily done. The server computer can even be replaced completely. All the administrator has to do is to transfer the FieldPoint explorer software and the LabVIEW program from the old machine to the new machine.

The FieldPoint modules used in the electric iron rig consisted of a network interface module *FP-1600*, one analogue input module *FP-AI-100* to read the signals from the temperature transmitters and one discrete output module *FP-DO-401* to output “on” and “off” signals to the Solid state Relay and solenoid valves.

The FP-1600 network interface module can connect up nine FieldPoint I/O modules to a high-speed 100Mbs/s Ethernet network. It also has the Onboard Intelligence for easy installation and diagnosis, such as “watchdog timer and diagnostics”. This function is very useful for the control system to prevent failure in the software. For the electric iron rig, the watchdog timer had been set such that the *FP-DO-401* module turns all the discrete outputs of the electric iron off when a failure of the system has been detected. This will provide an extra safety precaution in addition to the existing safety precautions in the Telelabs software (refer to http://www.ni.com/products/).

4 SOFTWARE

In the electric iron rig, the controller specifies output power of the electric iron based on the temperature readings from the sensors and the set temperature. A PID controller and Pulse Width Modulation were incorporated into the control system.

The beginning of this chapter introduces the language that was used to program the controller and explains why this particular language was chosen. It also details the programs that implement PID controller and Pulse Width Modulation. This chapter ends by briefly introducing other software that had been used in this project, namely the streaming server for visual effect and simulation program.

4.1 LabVIEW

LabVIEW stands for Laboratory Virtual Instrument Engineering Workbench. All the programs in LabVIEW are created by selecting icons from a menu representing controls (such as knobs, push buttons, graphs and so on) for the front panel. The logic behind the front panel is constructed from icons wired together on the "diagram" that specifies the computational operations required to make the front panel work [Trevelyan & Le Dain, 2001]. The programs created using LabVIEW are called virtual instruments (VIs) because their appearance and functionality are similar to that of the actual instruments. Figure 8 is the example of a LabVIEW VI.

LabVIEW is a very powerful programming language and it has all the functionality to implement online laboratories, such as: Internet communications, hardware connectivity and web serving technology. It also has a well-designed graphical interface and has almost the same execution performance as C or assembler programming languages. The language is also easily learned by providing the potential users with programming exercises and practical examples. A general-purpose language such as Java requires complex programming to achieve the same results. These are the reasons why LabVIEW program is utilised in the Telelabs project.

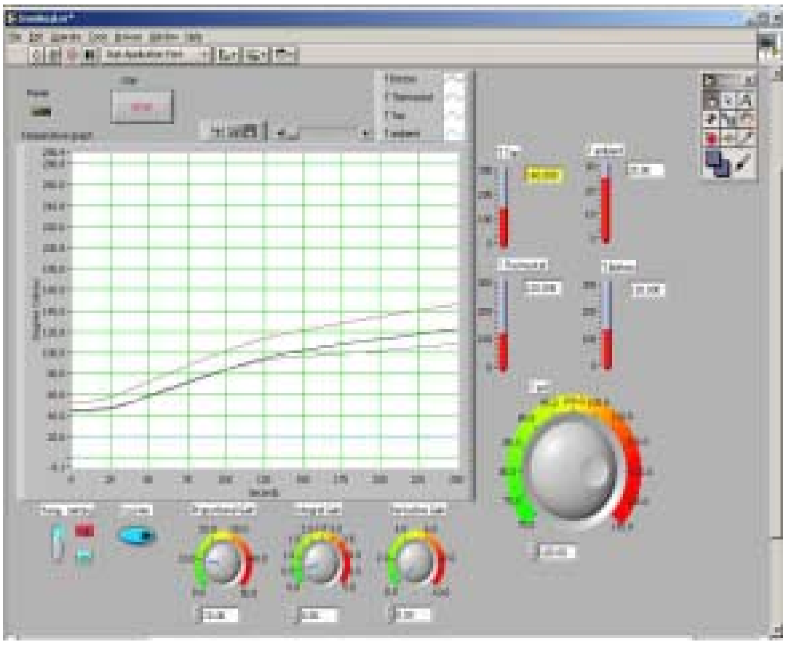


Figure 8. Front panel of the IronReal.VI

4.2 The controller of the rig

A local controller of the iron rig can be divided into two types: Simple control using thermostat and the advanced control using Pulse Width Modulation. Both methods were programmed using LabVIEW and their characteristics were compared.

4.2.1 Simple control using Thermostat

The LabVIEW program that implements simple-control using thermostat is called *SimpleIron.VI.* This program reads the inputs from the RTD sensors and plots them on the graph with respect to time. The hierarchy of this VI is shown in Figure 9.

The setting of the thermostat can be changed using "Temp. Setting" switch on the program. There are two settings for the thermostat: Low and High setting. In the graphs for both settings, fluctuations as large as $\pm 15^{o}C$ in temperatures were observed. Therefore, it was concluded that using a thermostat as a temperature-controlling device could result in a large error (see Appendix B1).

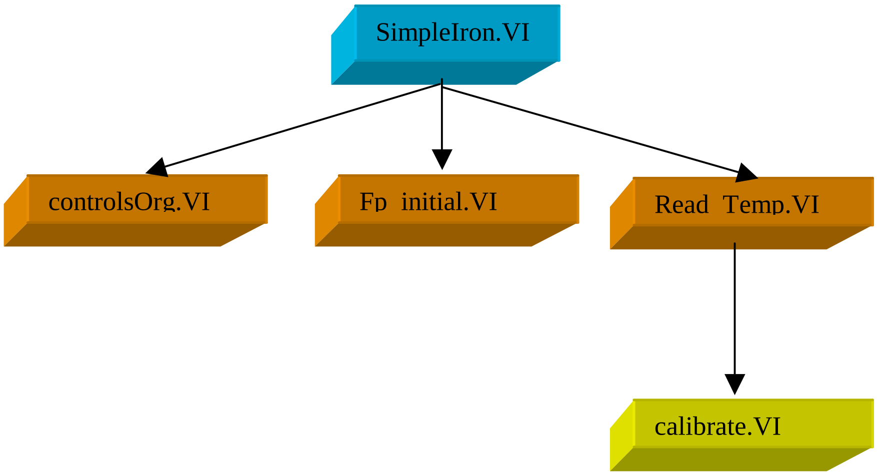


Figure 9. The VI hierarchy of simple control using thermostat

4.2.1.1 Fp_initial.VI

The communication between the electric iron rig and the server PC is specified by *Fp_initial.VI*. This sub-VI[4] contains commands to open the communication session to the FieldPoint server. It also contains settings to specify the correct modules to use and which channels on the module to read or write to (see Appendix B2).

4.2.1.2 controlOrg.VI.

This sub-VI contains all the discrete control for the electric iron rig and carries out all the initialisation and writing to the FieldPoint module. When a button is in the "up" state, a numerical value of "1" is sent to the FieldPoint module, which turns the particular device "on". When the button is in the "down" state and a numerical value of "0" is sent to the FieldPoint module, the device is turned off (see Appendix B3).

4.2.1.3 Read_Temp.VI

The *Read_Temp.VI* sends read commands to the FieldPoint module to read the signal coming in from the temperature transmitters. The signals are then passed to the

[4] Sub-VI is a term used in LabVIEW programming language to address a sub program used in another program.

calibrate.VI to convert them into temperature in degrees Celsius. The procedure for temperature calibration is discussed in section 3.2.
(see Appendix B4 & B5).

4.2.2 Advanced control using Pulse Width Modulation

The LabVIEW program that implements advanced control using Pulse Width Modulation is called *IronReal.VI* . The front panel of the program is shown in Figure 8. The main program to control the iron experiment locally (*IronReal.VI)* consists of three sections: *Control section, Input Section and PWM control section.* The three sections are separated using three 'while' loops in LabVIEW (see Appendix B6). Since the temperature vs time graph displays the temperature of all sensors with respect to time in seconds, it is desirable to have readings that change every second. Therefore, the execution frequency of the 'while' loop for the input section was set to one second and the program only reads the equipment once per second. The control section does not contain any delay, as instantaneous control is required to accurately calculate output from the PID controller and implement Pulse Width Modulation on the electric iron. The PWM control section uses the output from the PID controller, turns the power of the electric iron "on" and "off" for a period of time according to the inputs it gets from the PID controller.

The temperature setting (T_set) range of the electric iron rig was chosen specifically to be between 20-95^{o}C. The lower temperature of 20^{o}C was chosen specifically to be slightly lower than the usual room temperature (22^{o}C). The aim of that was to provide the user with an option to turn the power of the electric iron "off". When the temperature setting is set to 20^{o}C, the PWM control would turn the power "off" since this temperature (the desired temperature) is lower than the actual temperature. However, since the iron would never reach a temperature that is lower than the room temperature, the power of the electric iron would always be turned off.

On the other hand, as the electric iron contains a thermostat that operates between 138^{o}C –150^{o}C, choosing a temperature range below 138^{o}C would prevent the override of control from the thermostat. The block diagram of the advanced control is shown in Figure 10.

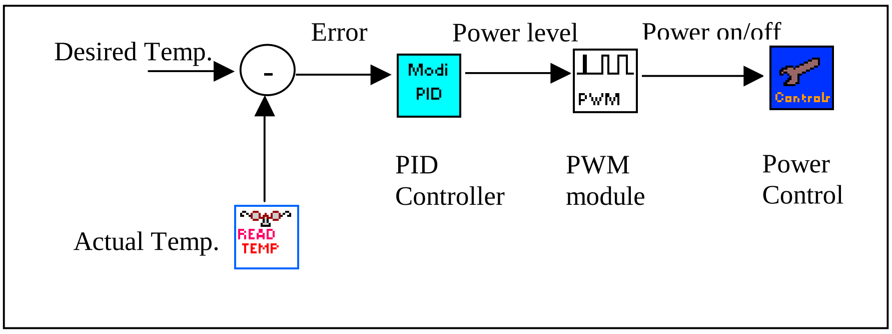


Figure 10. The block diagram of advanced control using PWM

4.2.2.1 Control section

The control section handles all the signals for controlling the electric iron: temperature knob high/low, cool air on/off signal and the main power on/off of the electric iron. This section uses two main subVIs: *modified PID.VI* and *controlOrg.VI*.

modified PID.VI

This sub-VI was modified from *simple PID.VI* in LabVIEW. Some logical error was found in the *simple PID.VI* and subsequently corrected. A windup limit was also incorporated into the sub-VI.

modified PID.VI calculates the power output of the electric iron and acts as a PID controller for the system. The time step (dt) for the PID controller had been chosen based on the execution time of the program. If the execution time of the program were slower than 40 milliseconds, the program would use 40 milliseconds as the time step. The power output is limited to a range between a lower (0 Watt) and upper limit (2000Watts). The upper limit of 2000Watts was based on the power consumption of the actual electric iron used in this project (see Appendix B7).

The proportional, integral and derivative components were calculated using the following formulas:

$$u(t) = K_p e(t) + K_I \int e(t)\ dt + K_D \frac{de(t)}{dt}$$

Where e(t) is the error function of the system.

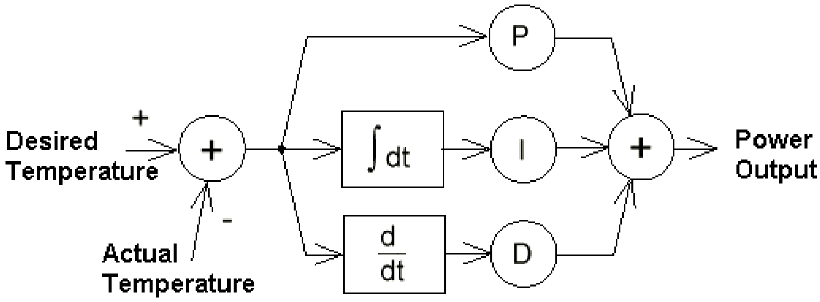


Figure 11. Block diagram for PID controller

The above equation can be rewritten as three parts in discrete form:

Error	$e_j = T_{desired} - T_{actual}$
Integral of error	$e_j^* = e_{j-1}^* + \delta t . e_j$
Proportional part:	$K_p e_j$ +
Integration part:	$K_I e_j^*$ +
Derivative part:	$K_D (\frac{e_j - e_{j-1}}{\delta t})$

Integration component can introduce excessive overshoot to the control system. This is often referred to as Integral windup. This is especially undesirable for the system, as a larger overshoot required longer cooling time, thus making the experiment longer to run. Therefore, windup limiting was incorporated to the integral component. The limit was chosen from a range of values 500, 1000 and 1500. The program was run using these limit values and it was found that the value of 1000 gave the optimum result. Therefore, the limit value was chosen to be 1000Watts. The overshoot was noted to reduce significantly after the implementation of windup limiting.

4.2.2.2 Input section

This part of the program reads inputs from the RTD sensors using the sub-VI *Read_Temp.VI* and calibrates them into temperature in degree Celsius using the program *calibrate.VI*. It had been designed to execute only once per second, such that it only reads from the RTD sensors and plots on the temperature chart every second. The user can also save the temperature data to a spreadsheet files when the program is terminated properly by pushing the stop button on the front panel of the program.

4.2.2.3 PWM control section

Pulse Width Modulation (PWM) refers to a method of carrying information using a train of pulses. The relative width of the pulses is the information being encoded. PWM is utilised in many control devices, for example on electric motor positioner. The method is very similar to the simple switching thermostat control. The only difference is that PWM is switching the device on/off at a much faster rate (microscopic level) and with a constant pulse frequency.

The power output from the *modified PID.VI* is converted to a power ratio by dividing it with a numerical value of 2000 and feeding it into *pwm.VI* (see Appendix B8). *pwm.VI* acts as a controller for implementing Pulse Width Modulation to the system. It calculates the on/off period of the electric iron based on the power ratio and subsequently turns the power of the electric iron on and off for that period of time. If the power ratio from the PID controller is 0.5 and the period of the PWM is 400 milliseconds, the program would turn the power “on” and “off” for 200 milliseconds respectively. The program also calculates the actual level of modulation achieved by comparing the time between the previous “on” and “off” state.

One of the important points to note when using an application, such as PWM is the *race condition*. This condition happens when a variable has more than one reference in the program. When the program is executed, one of the references is executed first. If the value of the reference is changing continuously, the other references may not be updated. This would produce differences in values between the two references at a particular instance although they point to the same variable. Therefore, the program had to be structured in the correct order: read inputs, calculation and send outputs. This structure would always give the correct result.

4.3 Streaming Server

The live video of the electric iron rig can be broadcast to the Internet using a steaming server to provide visual feedback during the experiment. A streaming server usually uses Universal Datagram Protocol[5] (UDP) to transfer data across the Internet or local Area Network. UDP is a protocol built on top of IP (Internet Protocol). This protocol provides high performance in data transmission, which is essential for video streaming. However, the trade-off for high performance is reliability where some data got lost during transmission. Since the transmission frequency in this protocol is relatively high, a few data lost in transmission would not be problematic.

The original plan for the Telelabs system was to use Net Meeting from the Microsoft to display the live video. However, the software only enables one user to view the equipment at any particular instant. Since students are to be subdivided into small groups in the experiment, the use of Net Meeting would only allow one particular group member to view the live video of the equipment and the rest of the group would miss out on the process.

An effort to search for better software was started in September 2001 by two third-year Mechatronics students as their third year project [Lau & Hsu, 2001]. They discovered software from “True Tech” which enables the user to broadcast live video to more than one user at a time. In fact, the free version software enables up to 20 people to view the video and the full version enables more than 20 people. The view from the streaming server is presented in Figure 12.

[5] A protocol is a set of rules or a “language” that allows two computers or devices to communicate with each other.

The streaming server can be accessed from http://130.95.52.248, with the following:

User name : streaming

Password : server

From the live video, the students would be able to see the movement of the pneumatic cylinder when the temperature knob setting of the electric iron changes. The students would also be able to see the movement of the wind indicator when the cool air in the rig is turned on.

Pneumatic cylinder

Wind indicator

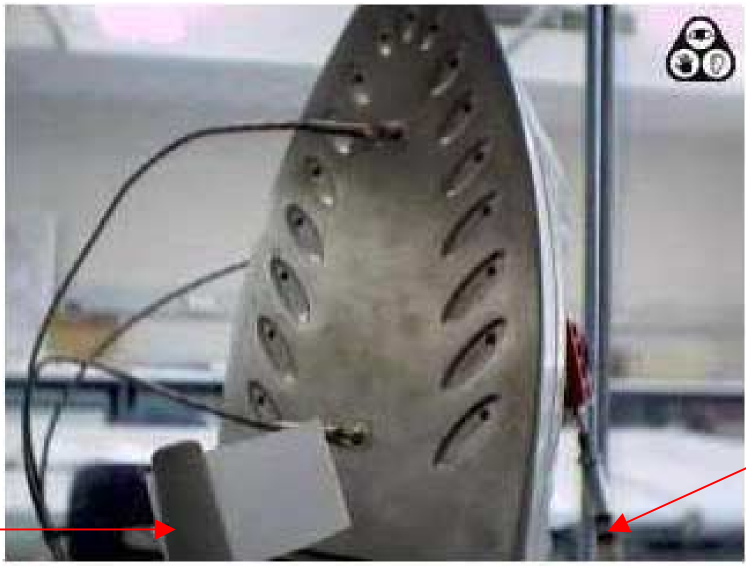


Figure 12 The view of the Electric iron rig from the Streaming server

4.4 Simulation

Simulations are usually used for the following purposes:

- To diagnose the behaviour of the system and find any potential errors.
- To discover problem in "theory" behind design
- To test for the safety of the system.

In the electric iron experiment, it would take up to 15 minutes for the system to reach the steady state with a temperature setting of 80^{o}C. Furthermore, it requires approximately 5 minutes to cool the system down using the cool air inlet. Therefore, one testing cycle for this system would take up to 20 minutes or more depending on the temperature setting.

A simulation is best suited when only limited time is available for the experiment. Therefore, the simulation program *IronSimulation.VI* was utilised to observe the behaviour of the system.

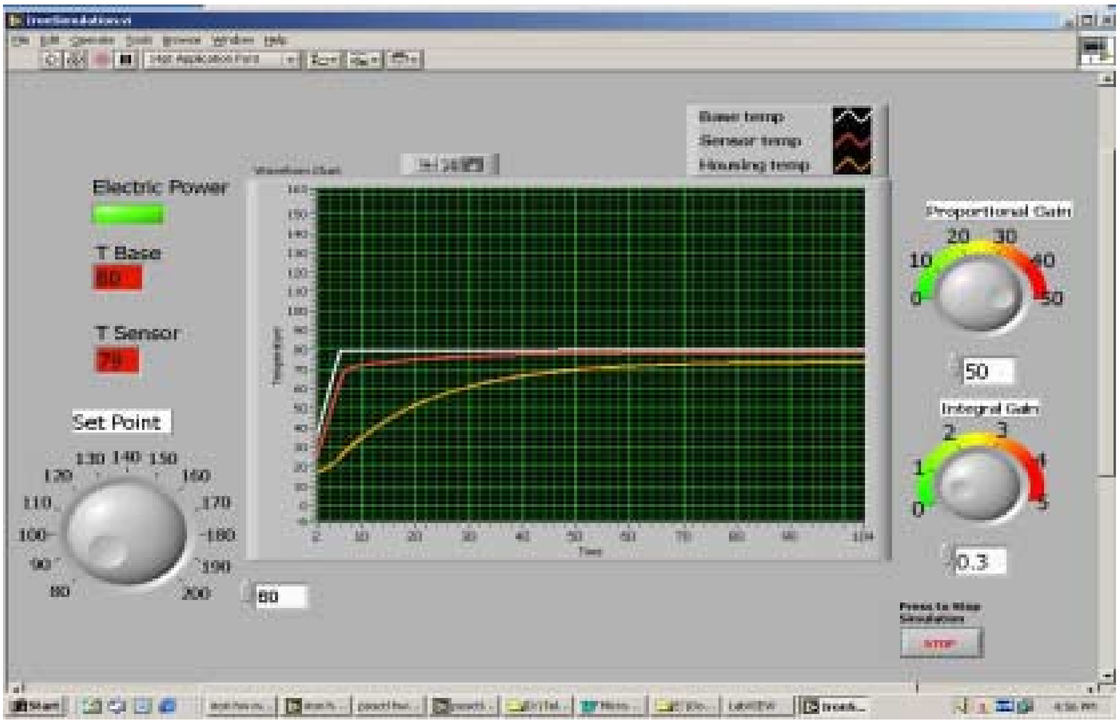


Figure 13 Simulation program for the electric iron experiment

The simulation program was developed by Trevelyan at the end of 2000 by using simple heat transfer and Euler integration. The author’s role was to incorporate the Proportional and Integral controller into the simulation program. The same windup limit in the local controller was also placed in the simulation program.

Instead of taking more than 20 minutes to run each set of data, the simulation only requires less than 5 minutes to run each set of data.

5 INTEGRATION INTO THE TELELABS SYSTEM

The next objective of this project was to incorporate the local controller into the Telelabs system such that students can control the electric iron rig from anywhere using the Internet. In order to achieve this objective, a good understanding of the whole system was essential.

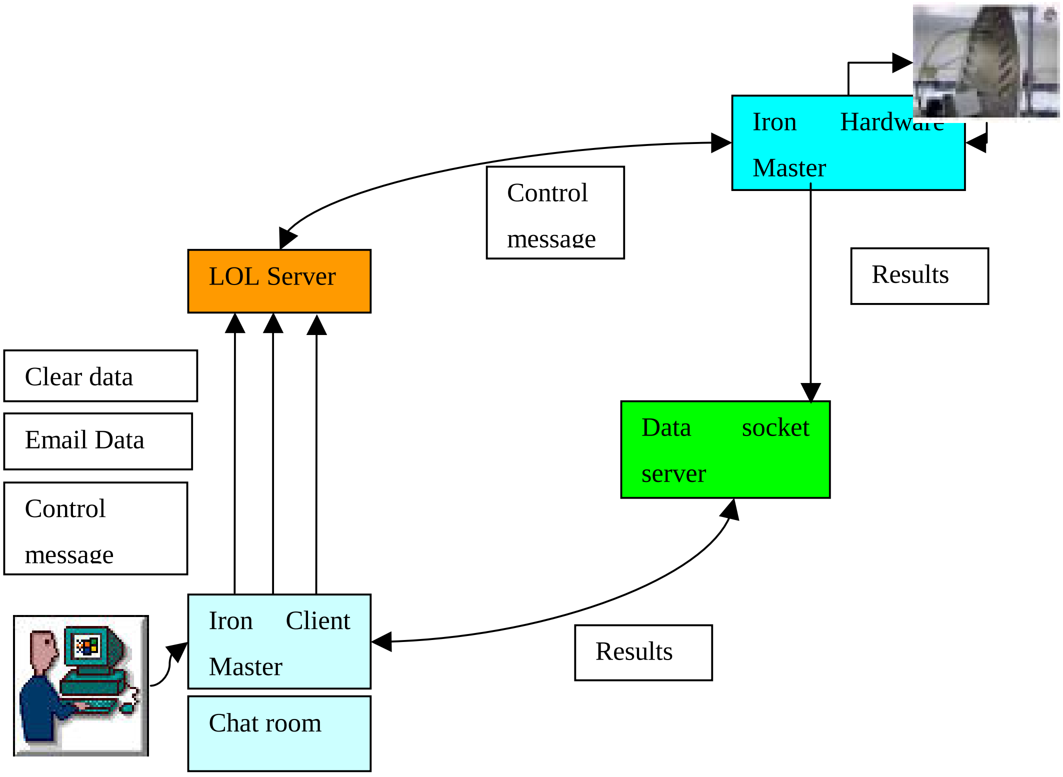


Figure 14 The Overview of communication between the Telelabs system

The software for the Telelabs system consists of three main components: the Labs On Line or LOL Server (LOL), the Hardware Client (*iron hw master.exe*), and the Remote Client (*iron client master.VI*). The LOL server handles the communications between the remote client and hardware client as well as providing all the database functionality (authentication, enrolment details and so on). The Remote Client provides the display for the remote user and the Hardware Client controls the experiment hardware. They connect using a standard client module that handles the connection with the LOL [Trevelyan & Le Dain, 2001]. The overview of the communication between the Telelabs software is shown in Figure 14.

Templates for the local controller (Hardware Master) and remote client (Client Master), along with libraries[6], were supplied in late August and revised frequently until early October. The template was given so that every program in the Telelabs project has the same overall structure. This makes the program easy to maintain and errors in the program easy to trace since there is a standard coding practice.

5.1 LOL client/server

The LOL server handles the communication between the Iron client master.VI and the Iron Hardware Master.VI. All the students data, login names, passwords are stored in the LOL server. It also has other functions, such as to email the experiment results and send control messages between Iron client master and Iron hardware master.

The LOL client is located on the user or student’s computer. The program can be downloaded from the Telelabs website[7] and installed onto student PC. To use the program, the students are assigned user names and passwords. The user names are usually their last names and the passwords are their student numbers. The program also allows students to change their user names or passwords if they desire.

When starting the LOL client program, a panel called the login client appears on the student’s computer screen asking for user name and password. After a successful login, the student sees another panel appearing on the computer screen called “Mechanical Engineering Hallway – UWA”. Figure 15 shows the front panel of the Hallway.

[6] A VI library is a special file that contains a collection of related VIs for a specific use.

[7] http://www.mech.uwa.edu.au/jpt/tele/

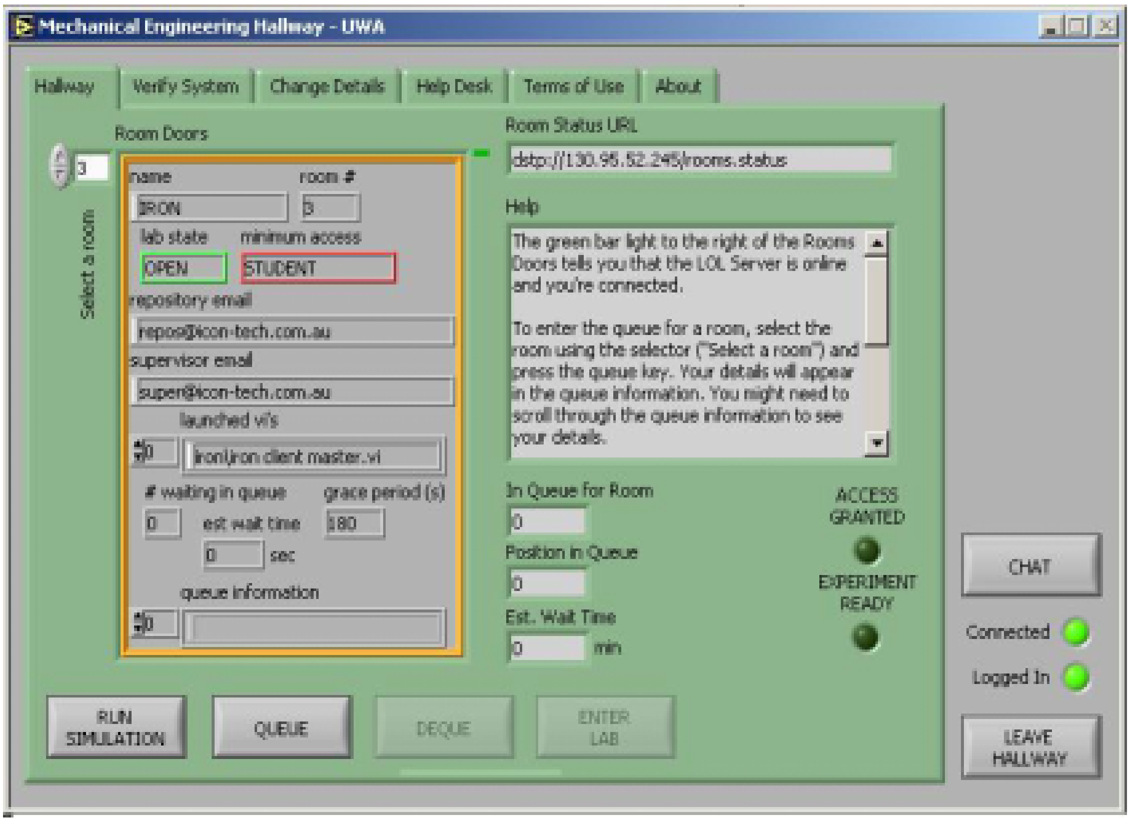


Figure 15 Mechanical Engineering Hallway

In the Hallway, the students can do the following things:

1. Access the laboratory.
2. Check their version of the LOL client software.
3. Change their details.
4. Send emails to staff for help.

At the moment of writing this thesis, items 2 and 4 were not available yet. These items would be available after February 2002.

Under the Hallway tab in the Mechanical Engineering Hallway panel, the students can access the laboratory by queuing for the equipment. The position in the queue and estimated waiting time are displayed on the panel. The students are notified when the equipment becomes available.

5.2 Data socket Server

Data socket is a technology from National Instruments to allow transfer of data across a network between LabVIEW and /or other software languages without concern for the low-level implementation details.

In the standard Transmission Control/Internet Protocol (TCP/IP), a code has to be written to be able to convert data measured by the acquisition cards into an unstructured stream of bytes in the broadcasting application. By using the data socket, the Telelabs program can avoid any management issue on the TCP/IP connections.

The other advantages of data socket are that it is protocol-independent, language-independent, and OS-independent and can be run on either a local computer or the Internet.

All the data from the iron hardware master, such as temperature reading, power level and control status of the laboratory are published to the data socket server. These data are then received by the iron client master and displayed on the computer screen of the students.

5.3 Iron hw master.exe

The main functions of this program are as follow:

1. It logs onto the LOL server when it is started.
2. Provides the LOL server with important information such as the number of identical hardware sets that are available for use.

This program manages the hardware used for a particular series of experiments. In order for this program to work, it has to be configured into the LOL server by an administrator who opens a particular room and experiment up for use [Trevelyan & Le Dain, 2001]. The room number for the electric iron experiment is 3.

To simplify the code of the Telelabs system, it had been determined that all the controls are grouped together in a single cluster of controls in LabVIEW and all the information or states, such as the temperature readings are grouped together into a single state cluster. Control or state variables can be added without extensive code changes throughout the software modules.

The control cluster and state cluster are shared and updated throughout the whole system, such that a change to these clusters at one part of the system would lead to changes to the clusters in the whole system. A simplified illustration of data flow in the Telelabs system is shown in Figure 14.

The electric iron is a relatively slow system and it takes a relatively long time to cool down and settle. If one student forgot to turn off the equipment after usage, the next student who gets to use to equipment would have to wait longer for the equipment to cool down. A precautionary procedure was incorporated into the program such that the equipment is always turned off when students have logged out of the Telelabs system. When the students leave the room, this program receives a message from the LOL server and it switches off the power of the electric iron and cool air and sets temperature setting to low (see appendix C).

5.3.1 Thin Controls cluster.ctl

The control cluster contains all the required controls for the electric iron rig, namely:

Controller name	Inputs
Temperature setting (T set)	20 – 95^{o}C
Temp. knob control (Temp. Setting)	high / low
Cool air control (Cool air)	On / off
Proportional gain	0 - 50
Integral Gain	0 - 5
Derivative Gain	0 - 50

When the students change any value in the control cluster at the iron client master side, the command is sent to the LOL server and the LOL server immediately sends the command to the Iron hardware master. The hardware master changes the setting on the hardware and reports back to the LOL server that the command had been executed (see Figure 14).

5.3.2 Thin state cluster.ctl

The state cluster contains information, namely:

1. Timestamp	2. Local/remote indicator
3. Equipment on/off indicator	4. Iron power on/off indicator
5. Desired Power level indicator	6. Actual Power level indicator
7. Temperature at the bottom of iron plate	8. Temperature at the top of iron plate
9. Temperature at the thermostat of the iron	10. Ambient temperature of the iron rig

1. Timestamp is the universal measurement of time and is time-zone-independent. The purpose of timestamp was to enable the programmer and users to keep track of the program activities every second. One special role of timestamp is in the experiment data that students sent to themselves through emails. If the experiment data sent to the students are not in the correct order, the students can still sort out the correct order using the timestamp.
2. Local/ remote indicator tells the user whether the electric iron rig is in the locally controlled mode or the remote controlled mode from the Internet.
3. The on/off equipment indicator tells the user whether the main power of the hardware has been switched on.
4. The Iron Power On indicator shows the power status of the electric iron. This indicator would continuously be flashing most of the time since a Pulse Width Modulation control is used in the system.

5. The Desired power level indicator shows the desired ratio for the on/off period of the electric iron.
6. The Actual power level indicator shows the actual power ratio achieved by the PWM program.

The hardware master publishes the information in the state cluster to the data socket server. This information is retrieve from the data socket server by the client master and displayed it onto the student's computer screen in the iron client master.VI

5.3.3 Thin HW Config.ctl

The thin HW config is very similar to the *FP_initial.VI* mentioned previously in section 4.2. It contains all the information necessary to open the communication session to the FieldPoint server. It also contained setting to specify the correct modules to use and which channels on the module to read or write to.

5.3.4 Thin Operate Hardware.VI

Thin Operate Hardware.VI is the main sub-VI that executes all the operations of the control system. The code for the local controller was copied and pasted into this program. There are four main cases for this program, namely:

- *Initialize.* All the code to initialise communication to the FieldPoint module was placed into this case. This case also sends out all the initial values to the state cluster and sets the control cluster to its default value. As its name suggests, this case is only executed when the Iron hw Master is restarted.
- *Run*. All the code to run the control system was placed in this case, namely: *modified PID.VI* and *pwm.VI*. This case also sends out the control signals and reads the input signals to/from the electric iron rig. This case is the core of the Iron hw Master.VI. There are two cases where this case would be executed: when the Iron hw Master.VI is in the local controlled mode or when the Iron hw Master.VI is in the remote controlled mode and a student is logged into the system.

- *Shut-down*. This case contains the commands to shut down the entire Electric iron rig and set the control value to the safe 'idle' position. The default values of the control cluster are also initialised here. This case would be executed in the following scenarios: when the Iron hw Master.VI had been shut down or when the Iron hw Master.VI is in the remote controlled mode and no student is logged into the system.
- *Anything else*. The case can be used to detect any errors in the code or program during operation. Nevertheless, it was not implemented by the author as it was deemed to be unnecessary at that stage of time.

5.3.5 Thin Make Datalog Array.vi

Thin Make Datalog Array.VI stores some specific experimental data from the control cluster and state cluster into an array. The data that are being stored in the array are timestamp, T bottom, T thermostat, T top and T ambient from the state cluster and T set, Proportional gain and Integral gain from the control cluster.

The data from this program are published to the Data Socket server. When the student pushes the "EMAIL DATA" button on the iron client Master.VI, the later sub-VI sends a request to the LOL server. The LOL server then retrieves the data from the data socket server and sends it to the student's email address.

5.4 Iron client Master.VI

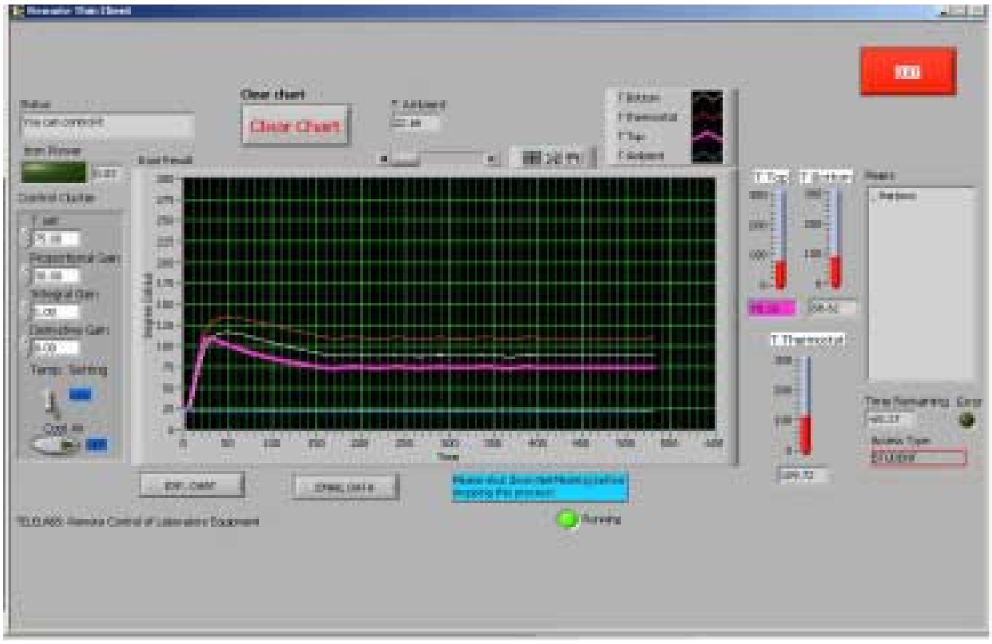


Figure 16. Front panel of Iron Client Master.VI

The results posted on the data socket server by Iron hw Master.VI are retrieved by the Iron client Master.VI. The results contain data, such as the temperature readings, iron power on/off and actual Power level indicator.

From the program, the client can observe the temperature of the electric iron rig at various parts on the thermometers. The temperature readings are also plotted on the graph to give students an overview of the system behaviour over time. The power indicator of the electric iron and its power level is also shown on the program. Appendix E explains various parts of the front panel for Iron Client Master.VI and how it is operated.

During the experiment, student usually makes some changes to the control cluster to observe the behaviour of the system under different parameters. The control message is then sent to the LOL server and the LOL server relays the control message to the Iron hw Master.VI to be executed. When the Iron hw Master.VI has accomplished the task, it sends a message back to the LOL server to indicate that the command has been executed.

The program also displays other members of the group that are in the room. The control cluster would grey out when another member is controlling or changing the setting. To avoid conflict in the experiment, the group co-operates and collaborates using the “EXP. CHAT”. They can chat and exchange information to each other using the chat function.

At the end of the experiment, the students can send the experimental data to their emails by clicking on “EMAIL DATA” button.

6 The electric Iron rig as a Teaching Tool

One of the main objectives for this project was to teach students the basic theory of feedback control system using the electric iron rig. The rig was used by students enrolled in the unit: Process Control 307 at the University of Western Australia. Due to the limited time they had for the experiment, the whole process was separated into two steps:

1. Hands on laboratory using Actual equipment and simulation.
2. Control of actual equipment through the Internet.

6.1 Local Controller and Simulation

The first stage of the experiment was actual Laboratory and the simulation. In this stage, the students were introduced to the electric iron rig and all the instruments and were given a chance to become familiar with the real equipment. They were also given a laboratory handout, which detailed the instruction of the experiment and some questions to test their understanding of the experiment (see Appendix D).

One of the problems with the experiment was time constraint. The students only had approximately 45 minutes to do the experiment. Furthermore, the electric iron is a relatively slow system and it takes a relatively long time to cool down and settle. This situation only allowed the experiment to be run once during the session. Therefore, the first stage of the experiment was based more heavily on the simulation.

Laboratory handouts were collected at the end of the experiment. It appeared from the handouts that most of the students had answered the questions in the handout correctly. This indicated that they had understood the experiment well. Furthermore, the students were asked whether the experiment was helpful in their understanding of the lecture material. Most of the students replied the question positively.

6.2 Online Laboratory

In the second stage of the experiment, the students used their results from the simulation to test the real equipment. The tests were done through the Internet, where students would log onto the Telelabs system called the LOL client. In this system, all the users who wanted to access a particular experiment type were queued to use the equipment. When a piece of equipment becomes free, the next user at the head of the queue has access to that experiment. Furthermore the current room queue is searched and all students with the same groupID obtain immediate access to that room.

Due to technical difficulty, the online laboratory was only available to students at the second last week of the semester. Students have a very high workload during this period. Therefore, the online laboratory was made optional to students. Nevertheless, They were encouraged to explore the laboratory.

One particular student who explored the laboratory had left a positive feedback to the system: "…the Iron lab that was put onto the web was beyond my expectation. At first, I did not expect to have the queue structure implemented as well as the chat room. Able to have a feature to email experimental data was unbelievable". That particular student also left some suggestions to the system. Some of the suggestions had been implemented and the rest are to be implemented in the near future (see Appendix F).

7 Conclusion

During the duration of the project, the author has implemented the following successfully:

1. Designed and built the electric iron rig. The electric iron rig was designed carefully by including safety and robustness. This enabled the electric iron rig to be used unsupervised at any time.
2. Programmed the local controllers for the electric iron rig. Two types of controller had been created: Simple control using a thermostat and advance control using Pulse Width Modulation (PWM). Both methods were programmed using LabVIEW and their characteristics were compared. The same set of equipment was used to achieve both controls.
3. Placed the laboratory online. The controller using PWM was integrated into the Telelabs system. This allows the control system to be controlled remotely on the Internet.

By completing the three tasks above, the author had successfully shown that building a laboratory from scratch and putting it on the Internet can be done in less than a year.

7.1 Future Work

This project has laid a solid foundation for the development of other online laboratories. Nevertheless, there is always room for improvements in either the hardware or the software aspects.

Possible improvements in the hardware:

1. Temperature visualisation. This has not been utilised in this project due to the difficulty in finding a material that can withstand high temperature (300^{o}C) and at the same time is reversible. Initially, there had been plans to use temperature sensitive paint to show the temperature distribution on the electric iron plate. However, it was later discovered that the paint could only withstand temperature of up to 90^{o}C. If the paint were exposed to temperature above 90^{o}C, components

of the liquid crystal in the paint would start to break down. After that, the temperature sensitive paint would be permanently changed and the colour change would become permanent. The problem of visualising the iron plate temperature is still not solved.

2. Computer hardware. At the moment, the iron hw Master.VI is running on a Pentium II 400MHz PC. Faster computer can be utilised if required.

There are potential solutions to the temperature visualisation problem, such as:

- *Using an Infrared camera*. An infrared camera could be used to observe the heat distribution of the electric iron. However, the cost of the camera might be substantial.
- *Using a heat filter*. Since the temperature sensitive paint can only withstand temperature below 90^{o}C, some sort of heat filter could be used to filter out the excessive heat. Therefore, the temperature sensitive paint could be utilised for this application.

Possible improvements in the software:

1. SMS functionality in the Telelabs software. The original idea was that the Telelabs system would send an SMS message to a student when it's his/her turn to use the laboratory. This will be implemented in the future.
2. Hardware and software diagnosis. A program could be written, such that it informs the administrator immediately when there is a critical hardware or software failure.
3. Online feedback and evaluation. A functionality can be built into the Telelabs system, such that students can send feedback and suggestions on how the system can be improved.

8 References

[1] Burrell, B., Wiggins, R. J. N., Sonwalkar, N., Kutney. M. C., Dalzell, W., Colton, C. K., 2000, ‘A Comparison of Web-Based and Laboratory Learning Environment’ in *2000 ASEE Annual Conference Proceedings*, St. Louis, Missouri.

[2] Chang, V., 2001, *Remote Collaboration in WebLab – an Online Laboratory*, Master thesis, Massachusetts Institutes of Technology.

[3] del Alamo, J. A., ‘Microelectronics Weblab used in 3 Fall 2000 subjects, and Singapore’, Massachusetts Institutes of Technology, [Online], Available: http://i-lab.mit.edu/ [13 August 20001].

[4] Di Benedetto, P. E., 1999, *Digital Motor Position Control*, Honour thesis, University of Western Australia.

[5] Esche, S. K., Prasad, M. G., Chassapis, C, 2000, ‘Remotely Accessible Laboratory Approach for Undergraduate Education’, in *2000 ASEE Annual Conference Proceedings*, St. Louis, Missouri, Session 3220.

[6] Filler, T. J., Jerosch, J., Peuker, E. T., 2000, ‘Live interdisciplinary teaching via the Internet’, *Computer Methods and Programs in Biomedicine*, Vol. 61, pp 157-162.

[7] González-Castaño, F. J., Anido-Rifón, L., Vales-Alonso, J., Fernández-Iglesias, M. J., Llamas Nistal, M., Rodríguez-Hernández, P., Pausada-Carballo, J. M., 2000, ‘Internet access to real equipment at computer architecture laboratories using the Java / CORBA paradigm’., *Computers & Education*, Vol.36, No.2, pp. 151-170.

[8] Henry, J., 2000, ‘24x7: Lab Experiments Access on the Web All the Time’, in *2000 ASEE Annual Conference Proceedings*, St. Louis, Missouri, Session 3613.

[9] Lau, J., Hsu, A., 2001, *Video Streaming Server for Laboratories*, Mechatronics Systems 210 Report, University of Western Australia.

[10] Layne, S. P., Beugelsdijk, T. J., 1998, ‘Mass customized testing and manufacturing via the internet’, *Robotics and Computer-Integrated Manufacturing*, Vol.14, No.5-6, pp. 377-387.

[11] Lemckert, C. J., Florance, J. R., 1997, 'Considerations in the design of real-time Internet mediated laboratory experiments' in *1st Asia-Pacific Forum on Engineering & Technology Education*, Melbourne.

[12] Ogata, K., 1997, *Modern Control Engineering*, 3rd Ed, Prentice-Hall, Minnesota.

[13] Rafe, G. E., Needy, K. L., Bidanda, B., 2000, 'Development and Implementation of a Distributed Virtual Laboratory for Continuous Manufacturing Education and Training' *in 2000 ASEE Annual Conference Proceedings*, St. Louis, Missouri, Session 2257.

[14] Salzmann, C., Gillet, D., Latchman, H. A., Crisalle, O. D., 1999, 'On-line Engineering Laboratories: Real-Time Control Over the Internet', in *2000 ASEE Annual Conference Proceedings*, St. Louis, Missouri, Session 2532.

[15] Stone, B. J., Scott, N., 1997, 'A flexible Web-based tutorial system for engineering, maths and science subjects' in *1st Asia-Pacific Forum on Engineering & Technology Education*, Melbourne, Australia.

[16] Travis, J., 2000, *Internet Applications in LabVIEW*, Prentice Hall, New Jersey, United States of America.

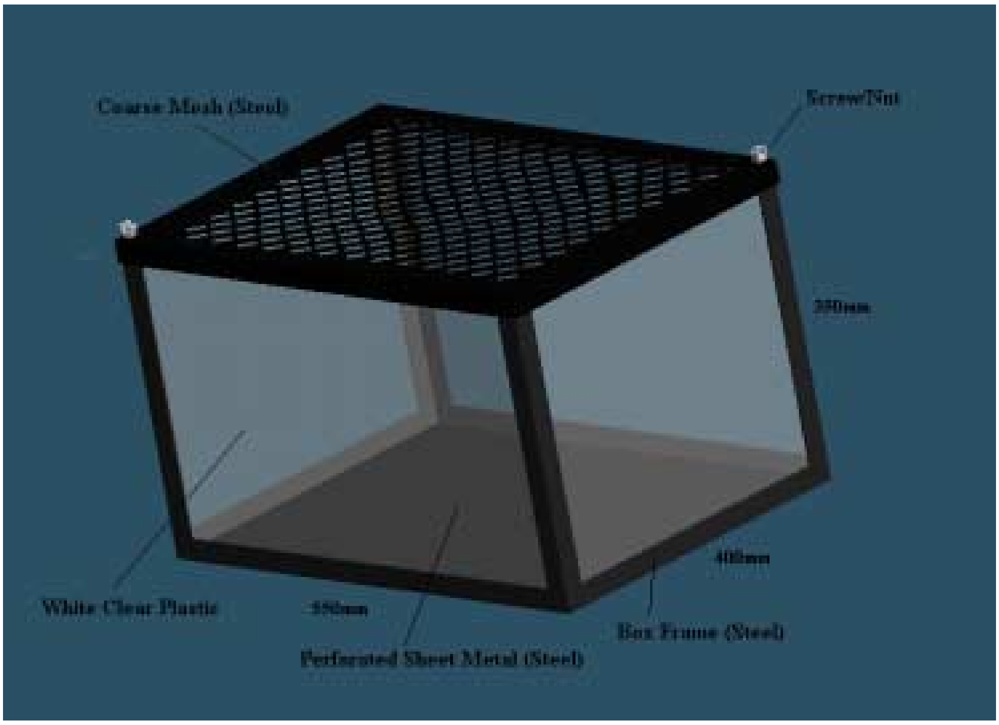
[17] Travis, J., Wells, L. K., 1997, *LabVIEW For Everyone: Graphical Programming Made Even Easier*, Prentice Hall, New Jersey, United States of America.

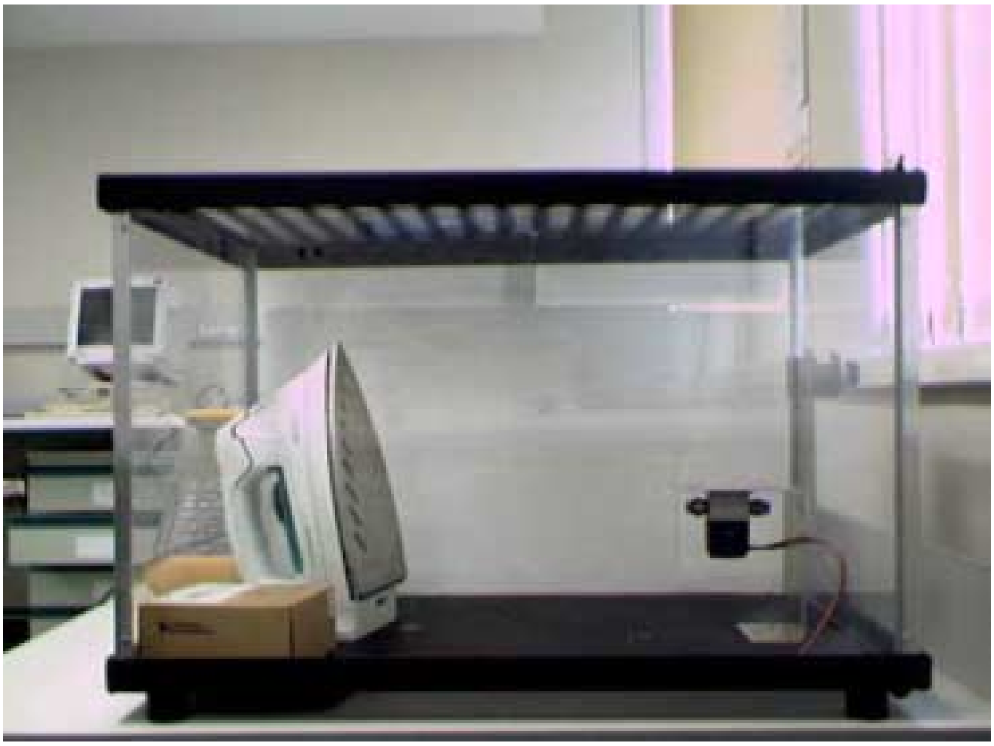
[18] Trevelyan, J., Le Dain, A. C., 2001, *Telelabs Task Specifications for Software Development*.

Appendix A:

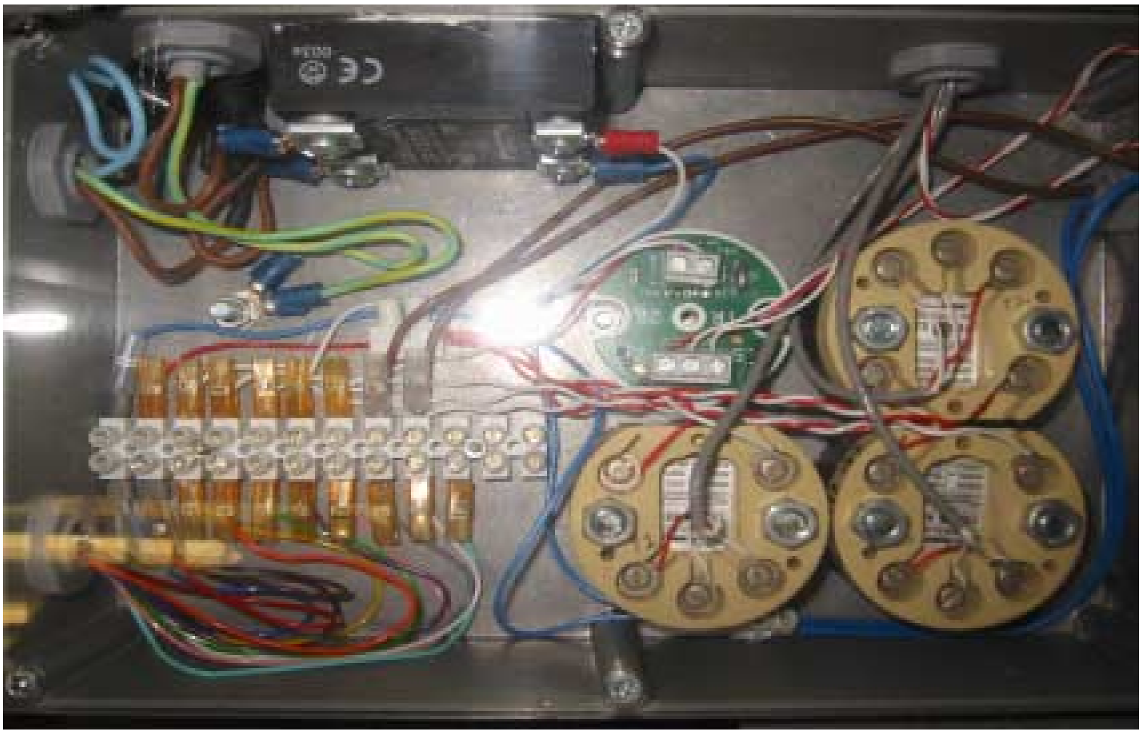
Hardware Specification

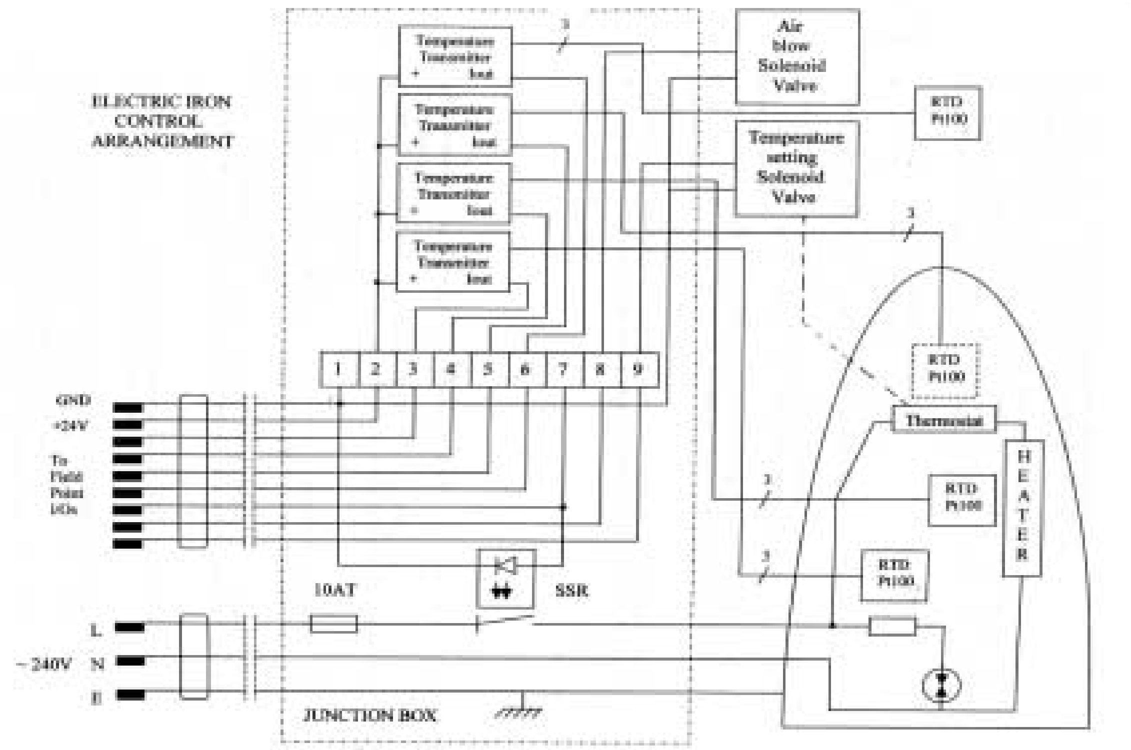
1. Design for the Enclosure of the Rig





2. Electrical Component & Diagram



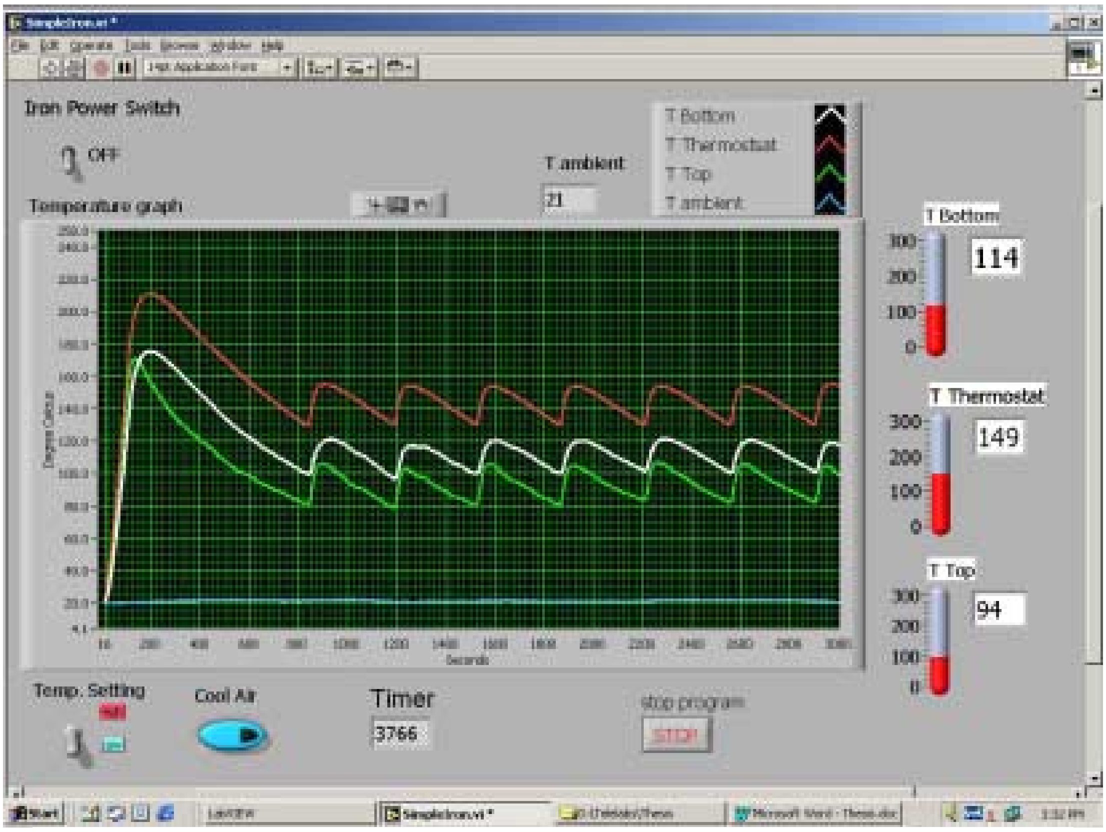


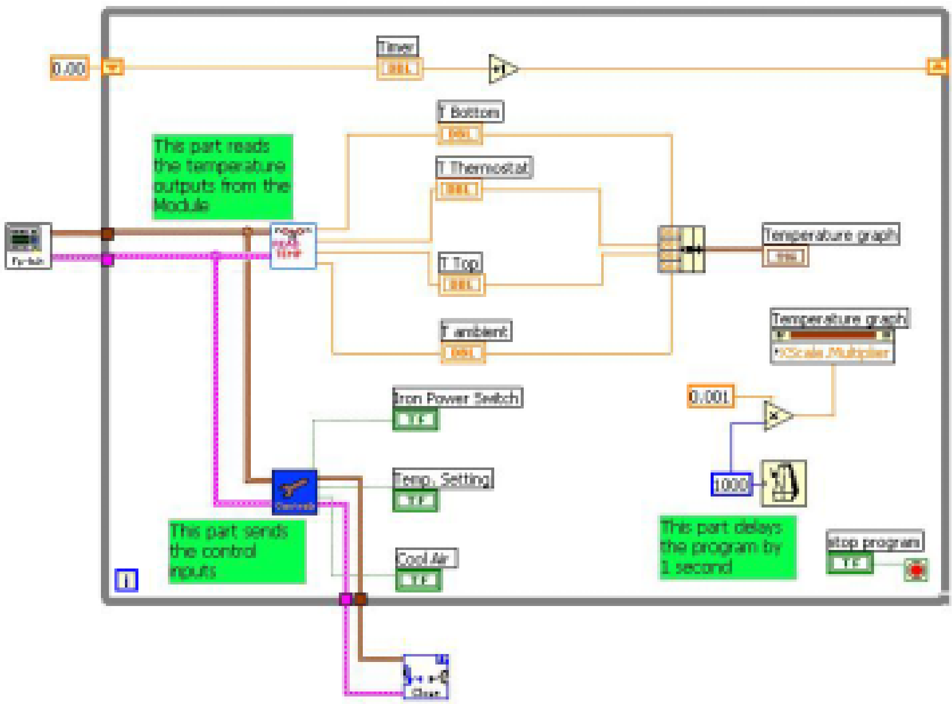
3. FieldPoint Module

Appendix B:

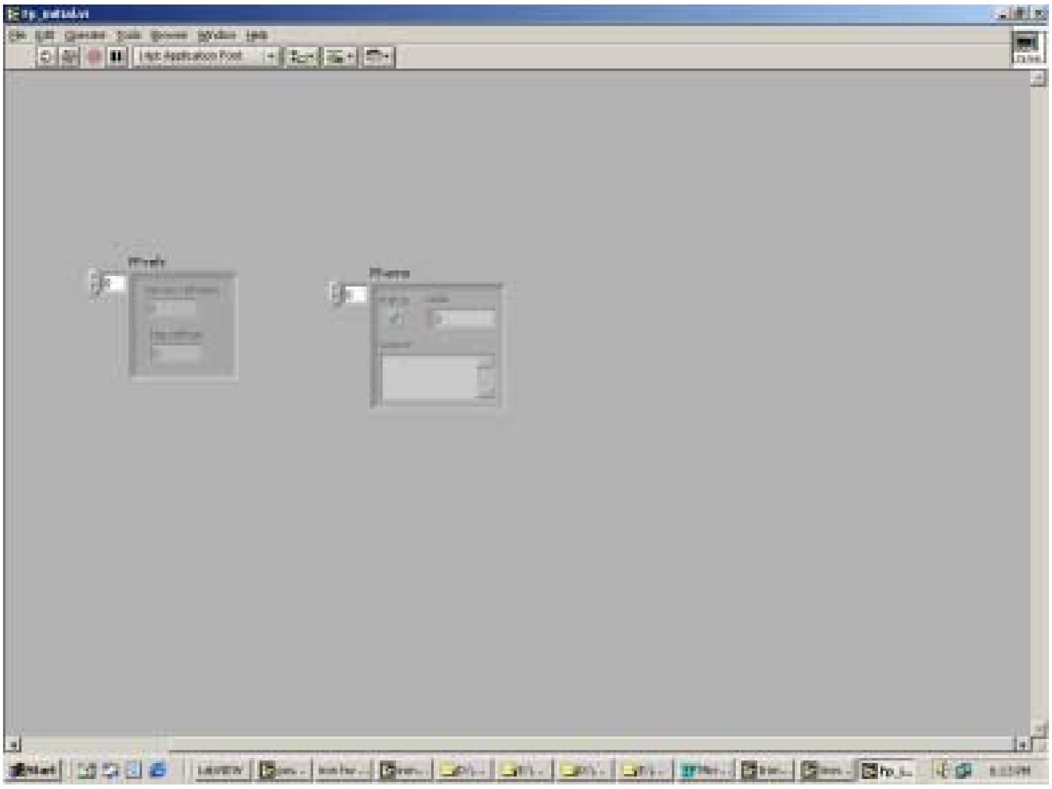
LabVIEW Codes

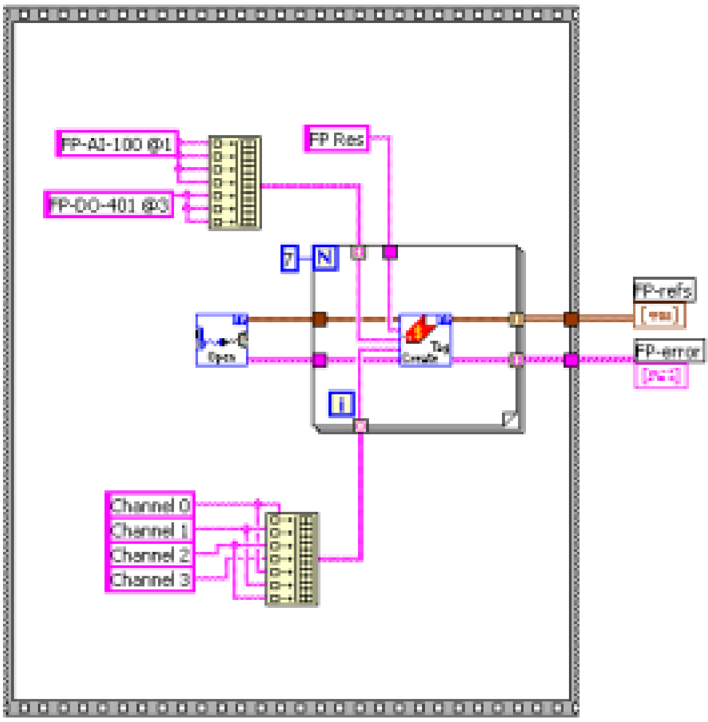
1. SimpleIron.VI



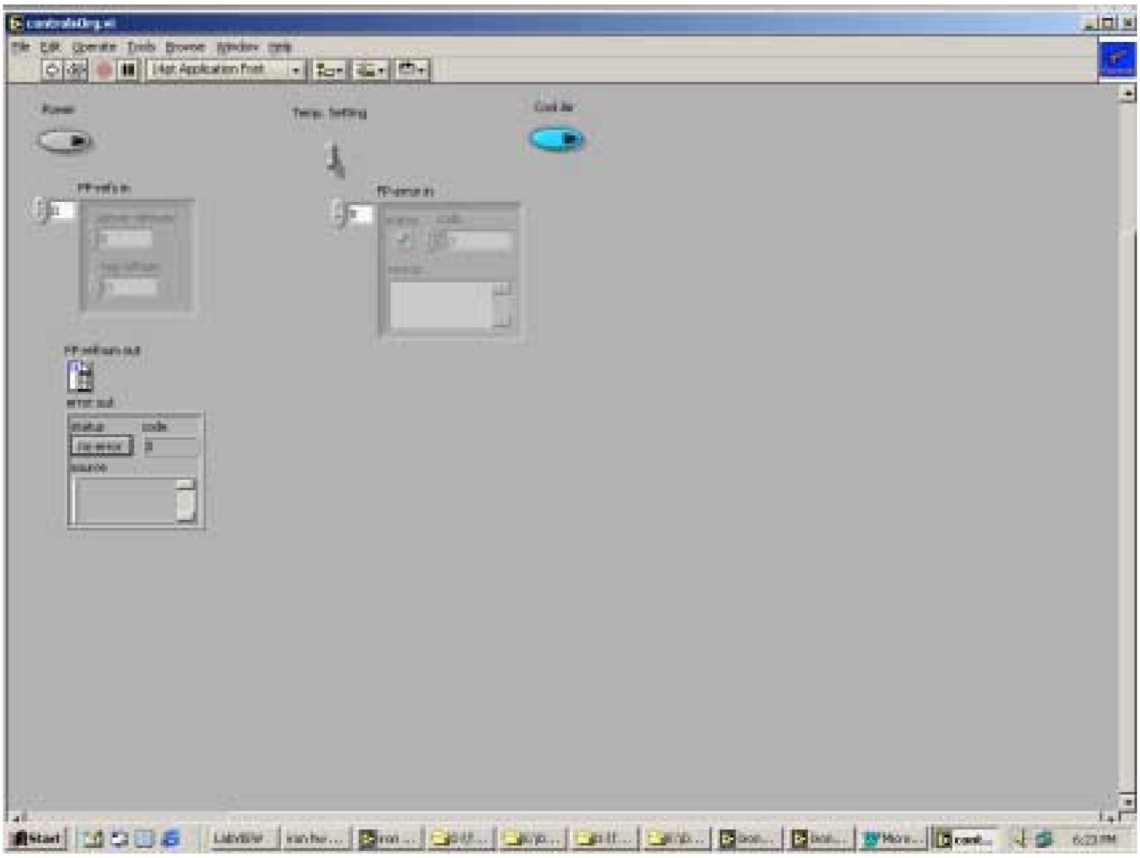


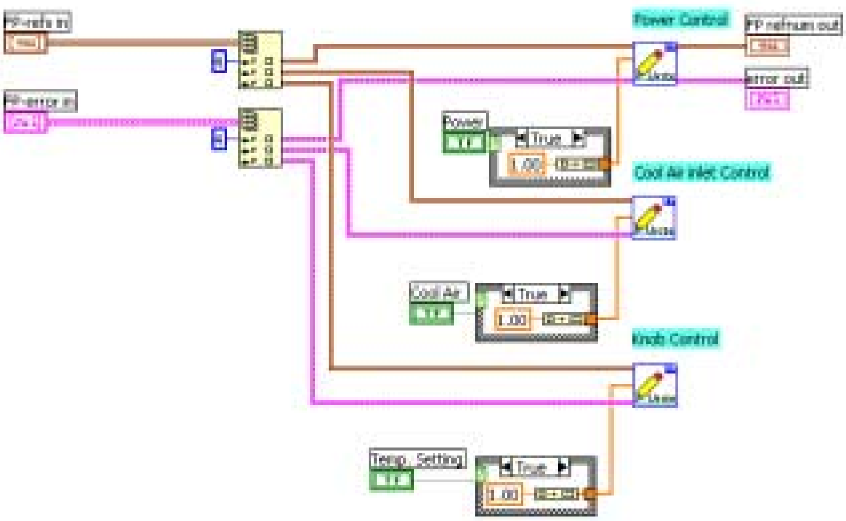
2. Fp_initial.VI



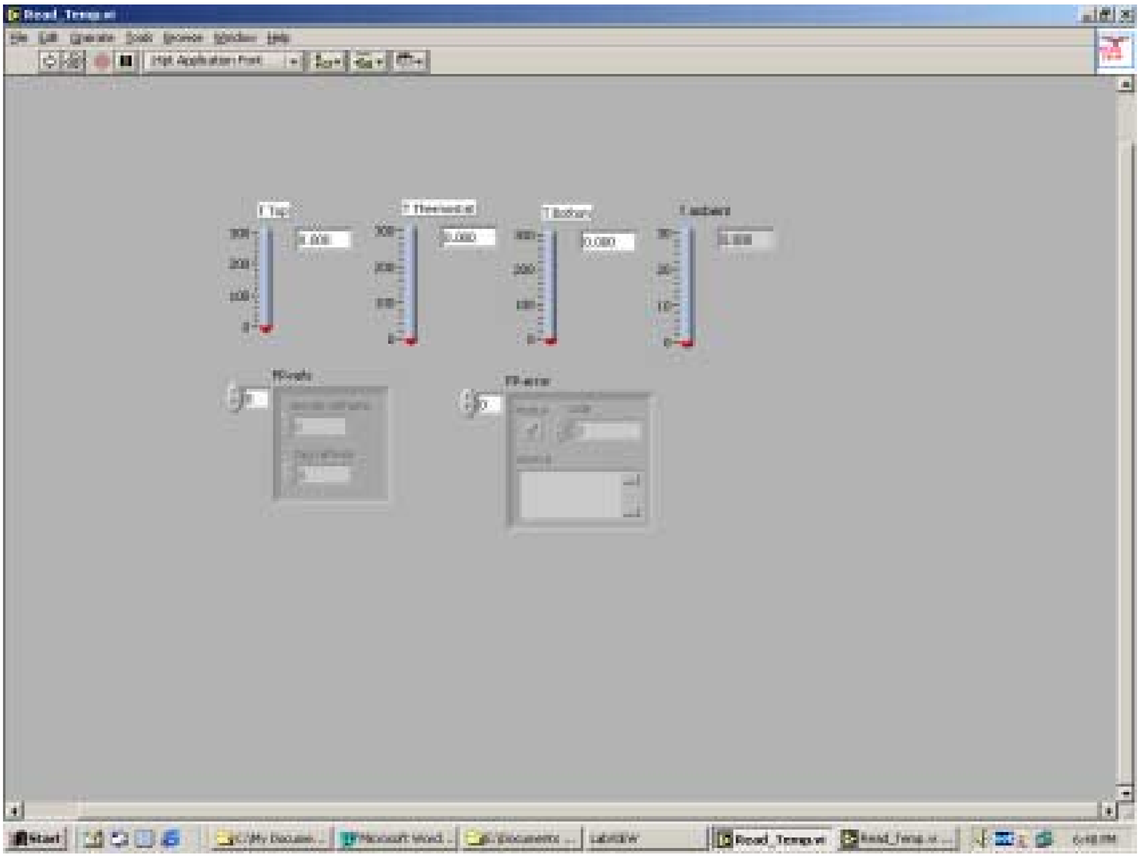


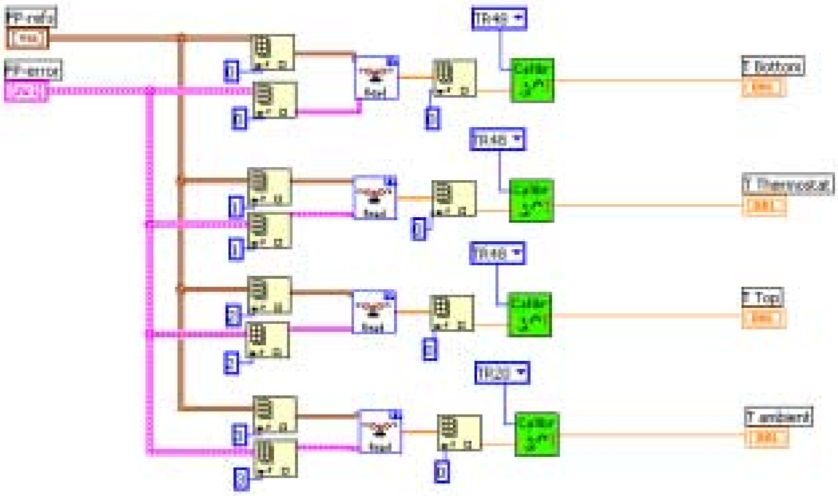
3. controlOrg.VI



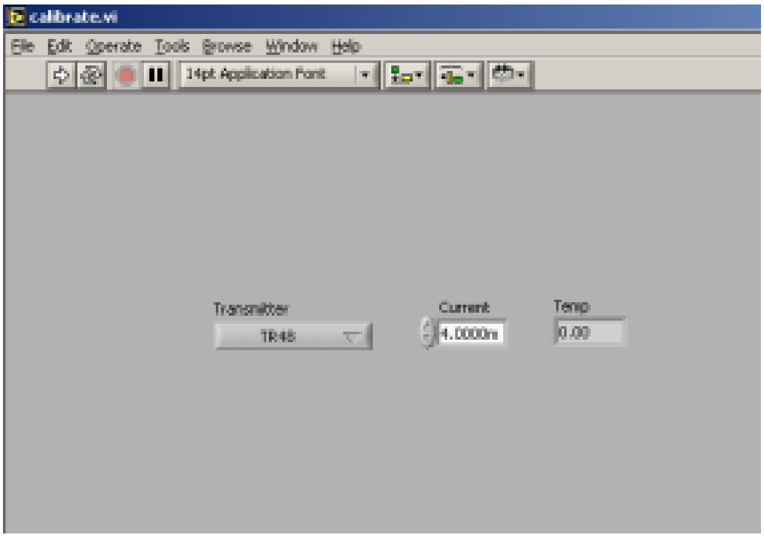


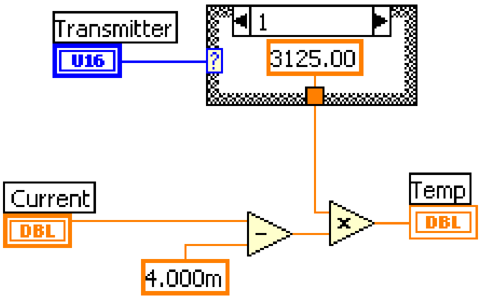
4. Read_Temp.VI



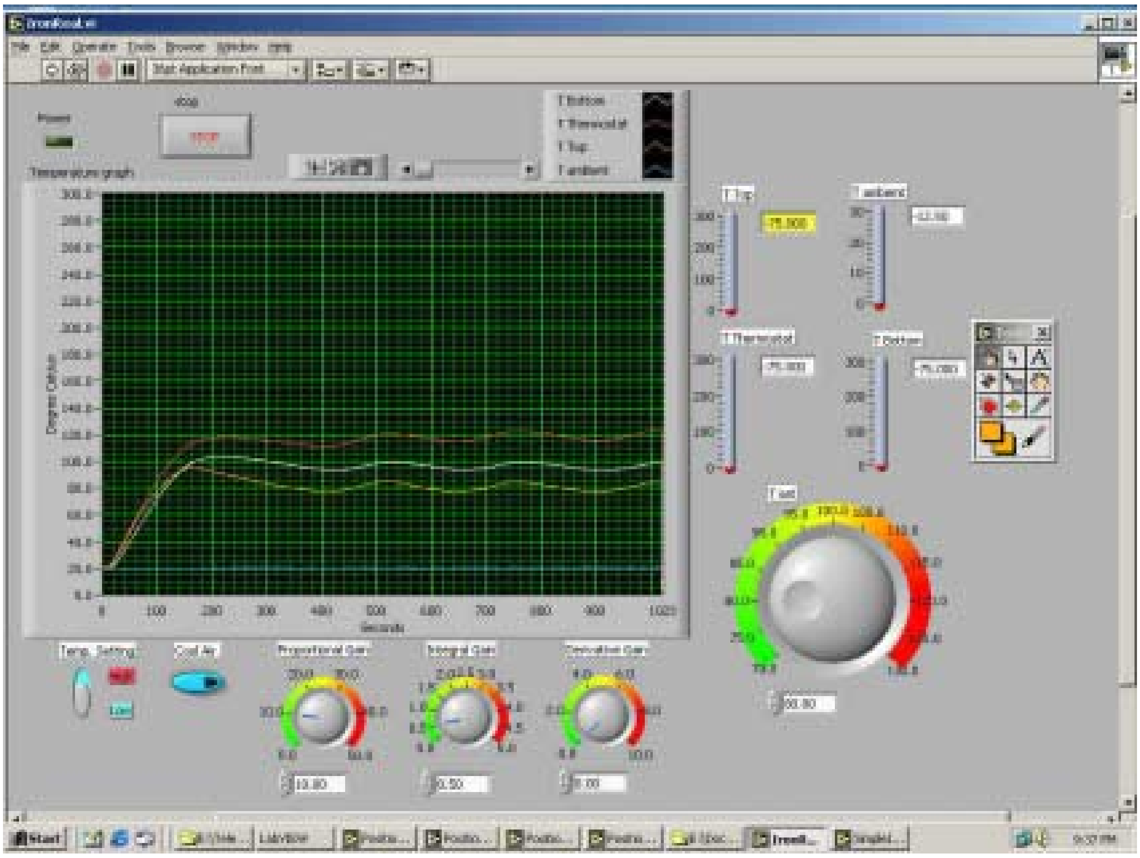


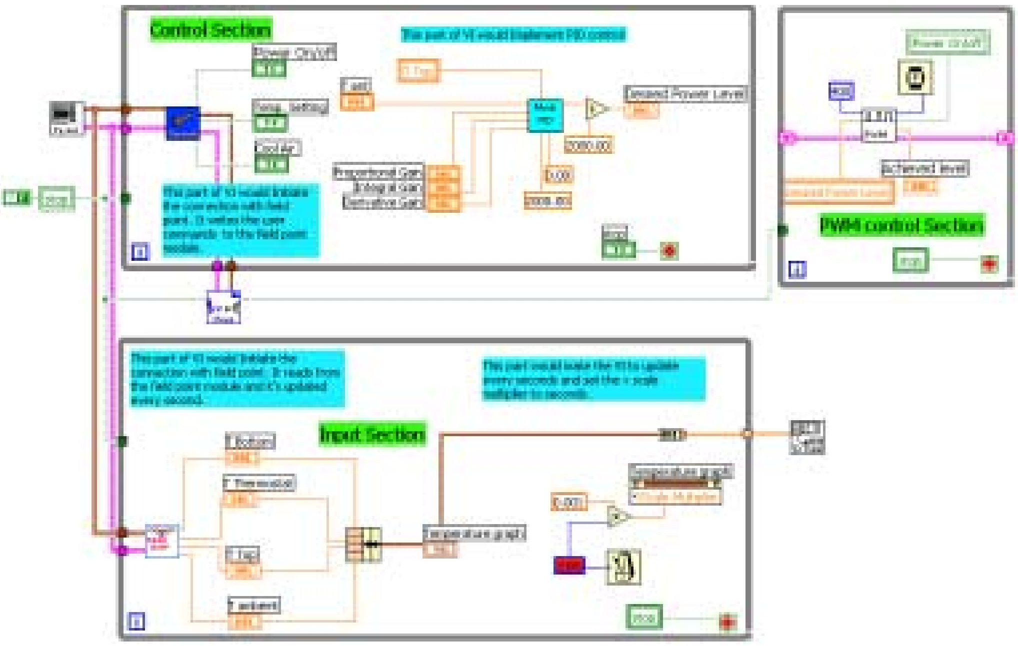
5. calibrate.VI



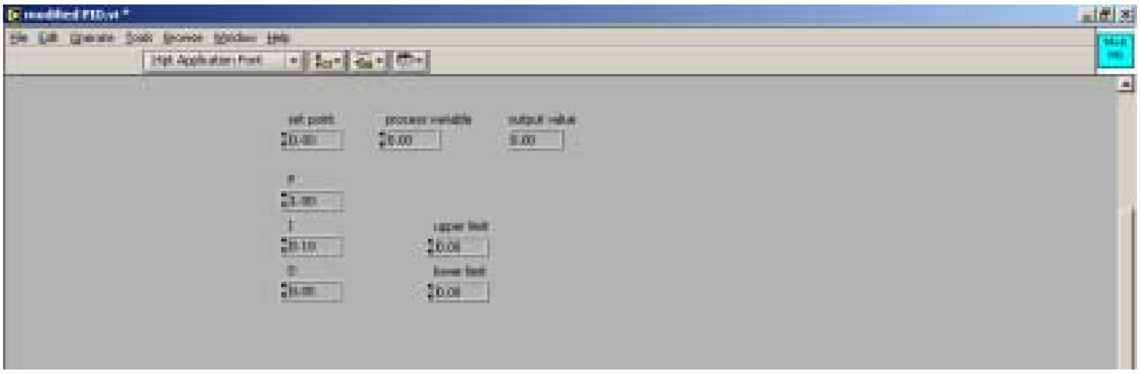


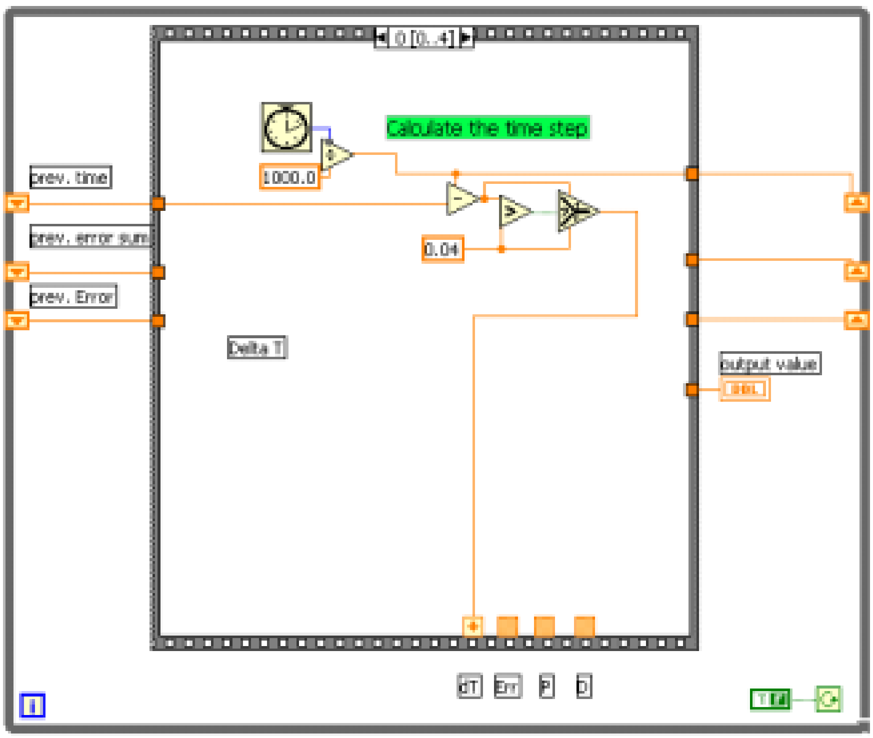
6. IronReal.VI

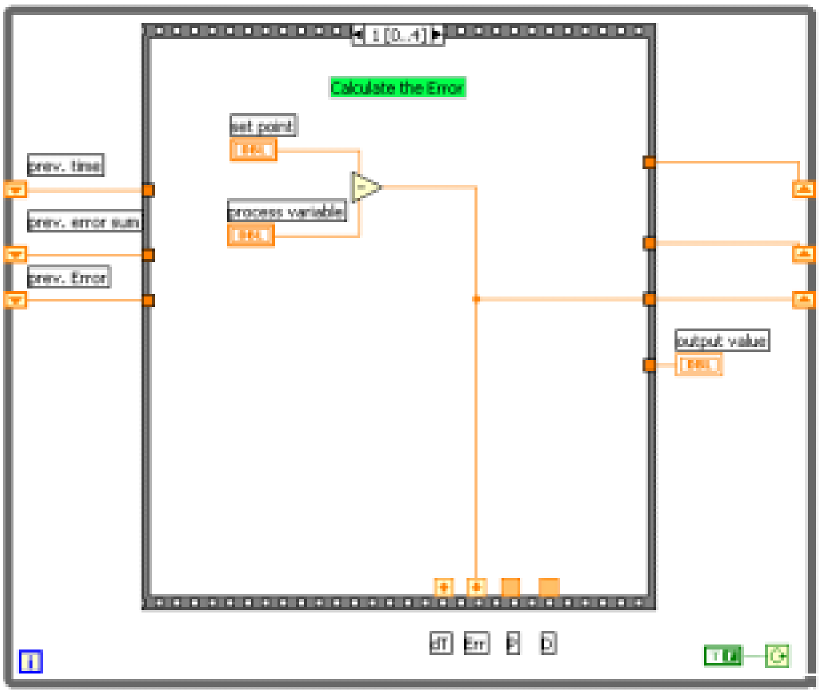


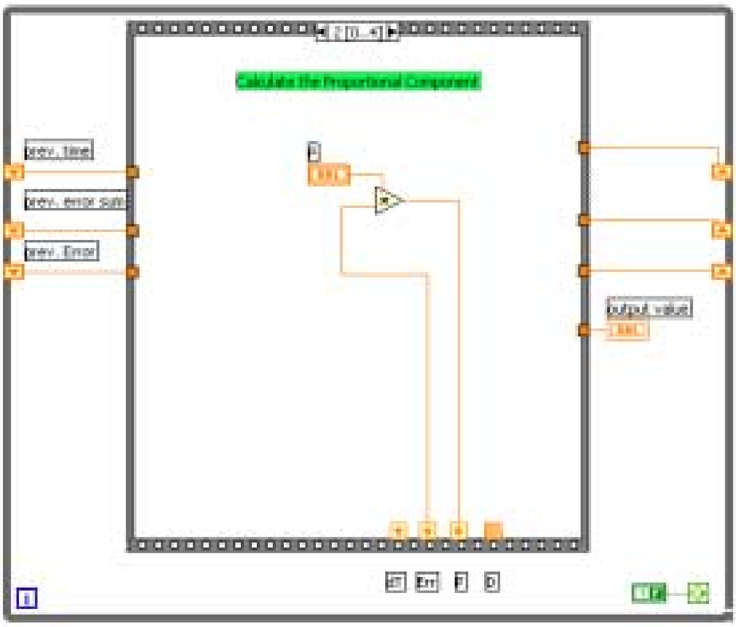


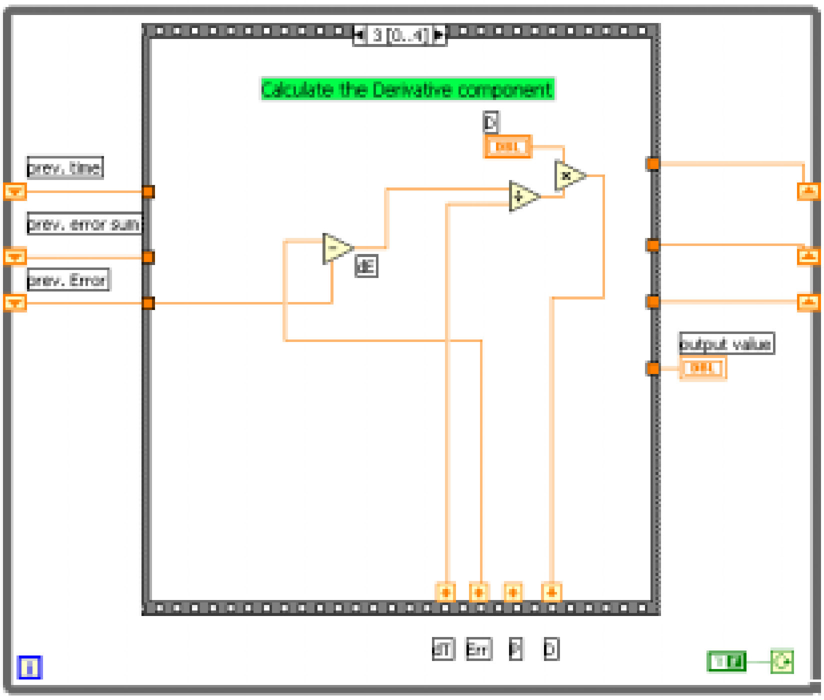
7. modified PID.VI

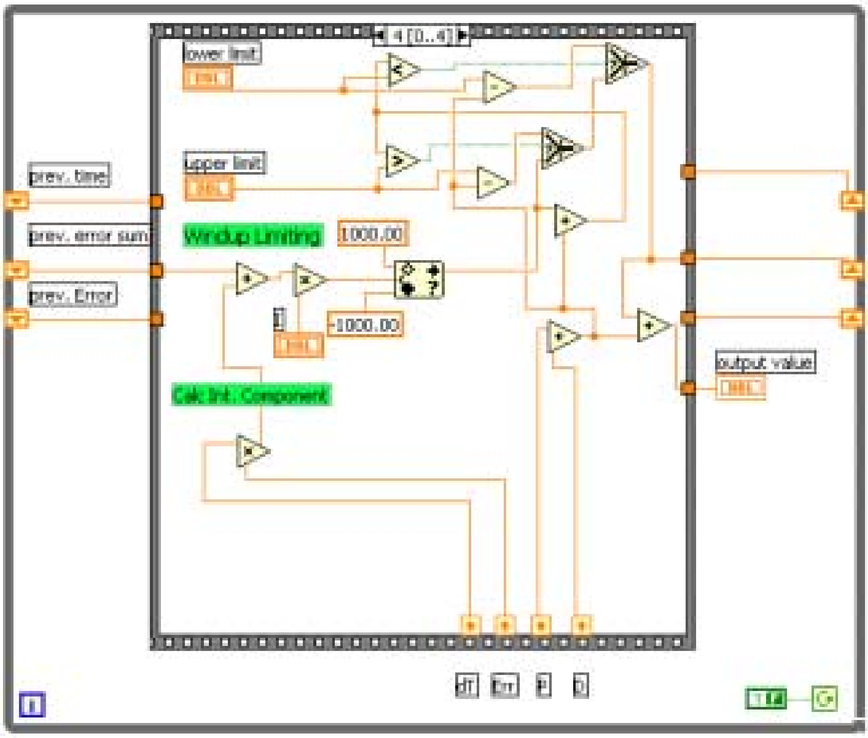




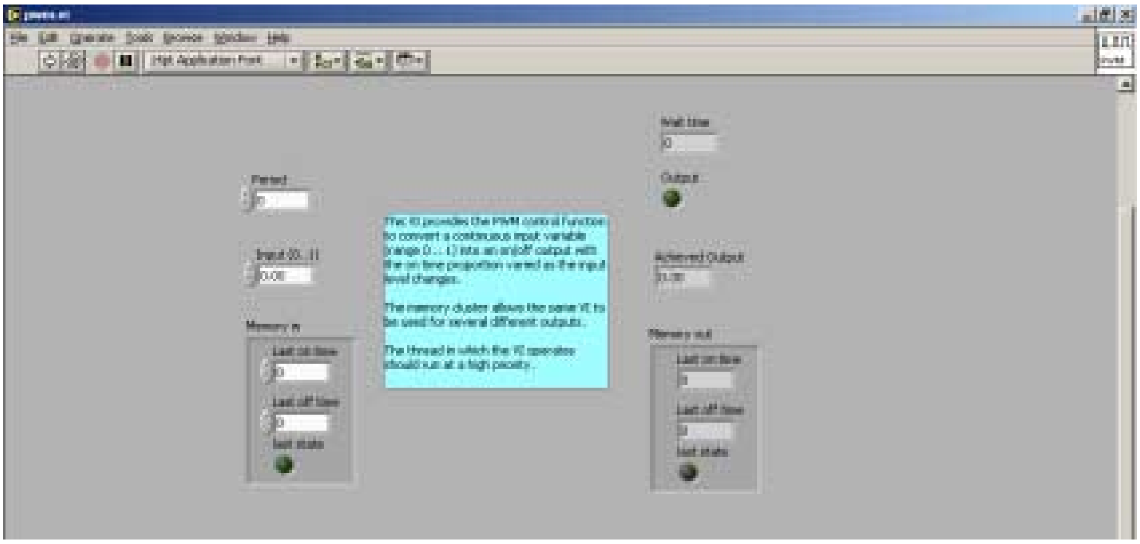

Calculate the Error
set point
process variable
prev. time
prev. error sum
prev. Error
output value
dT
Err
P
D

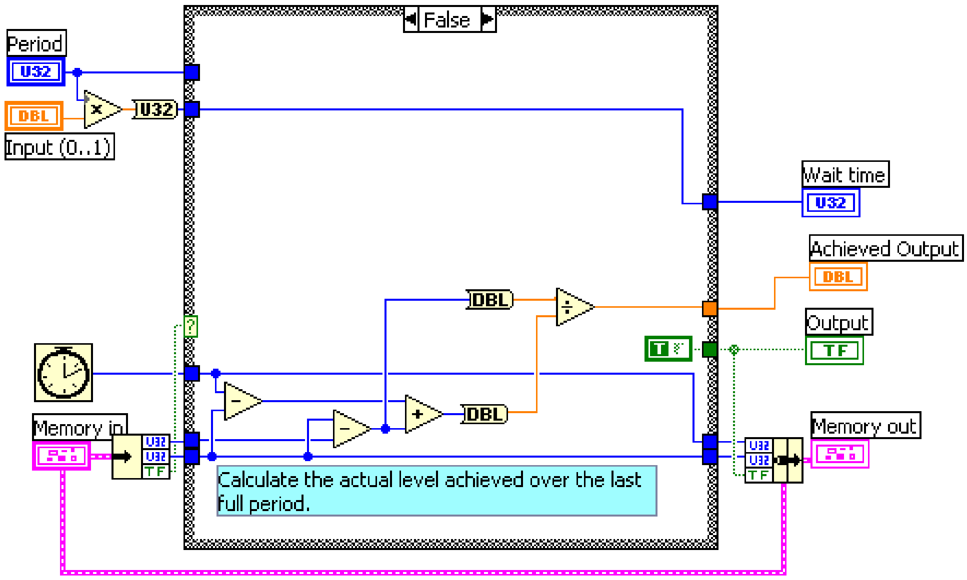

Calculate the Proportional Component
P
prev. time
prev. error sum
prev. Error
output value
dT
Err
P
D

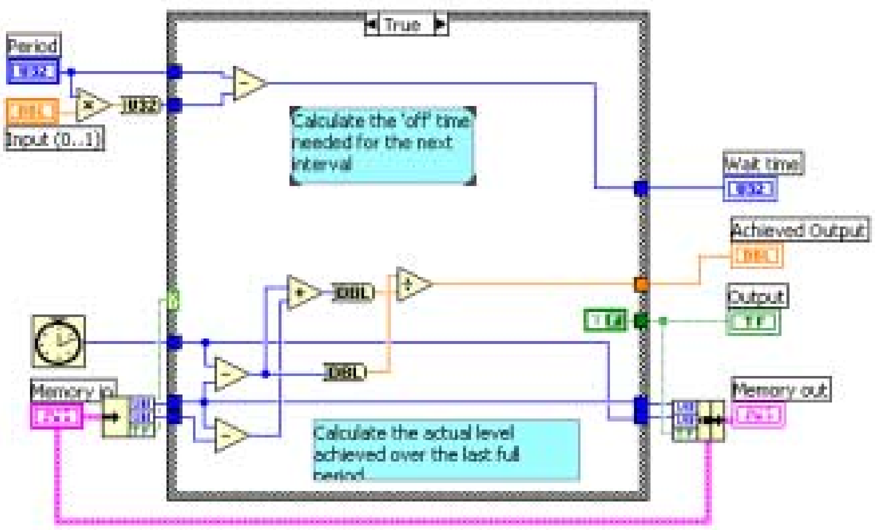

3 [0..4]
Calculate the Derivative component
D
DBL
prev. time
prev. error sum
prev. Error
dE
output value
DBL
dT
Err
P
D


4 [0..4]
lower limit
DBL
upper limit
DBL
prev. time
prev. error sum
prev. Error
Windup Limiting
1000.00
-1000.00
I
DBL
Calc Int. Component
output value
DBL
dT
Err
P
D

8. pwm.VI



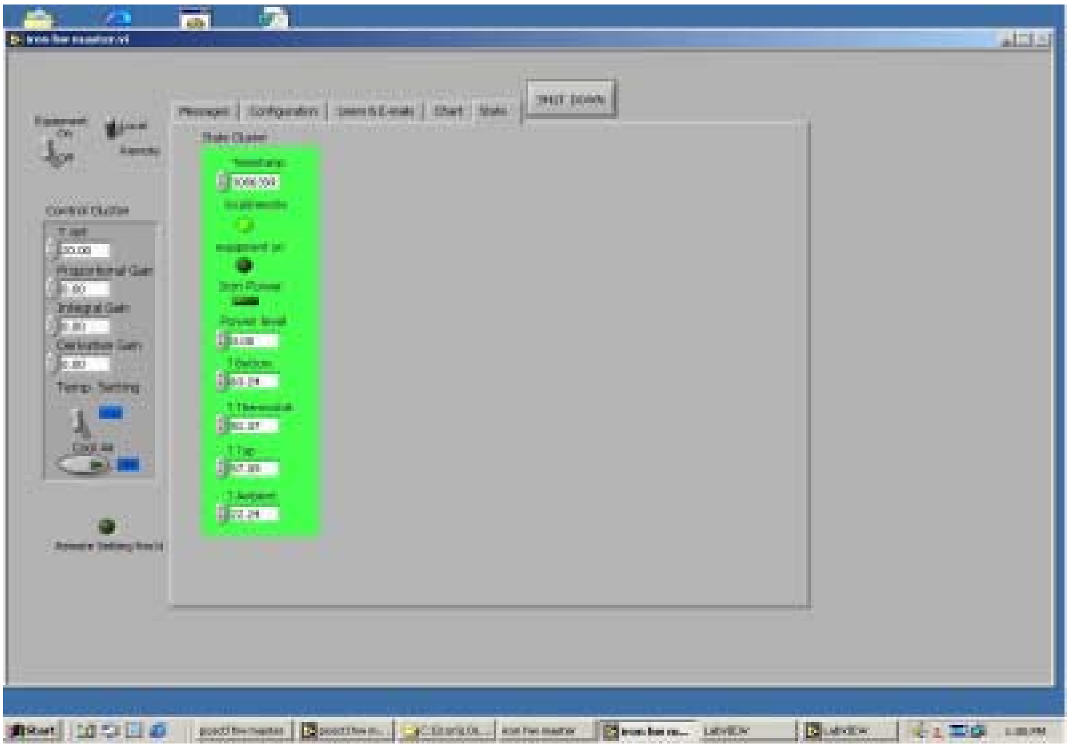

False
Period
U32
DBL
Input (0..1)
U32
Wait time
U32
Achieved Output
DBL
DBL
Output
TF
DBL
Memory in
Memory out
Calculate the actual level achieved over the last full period.

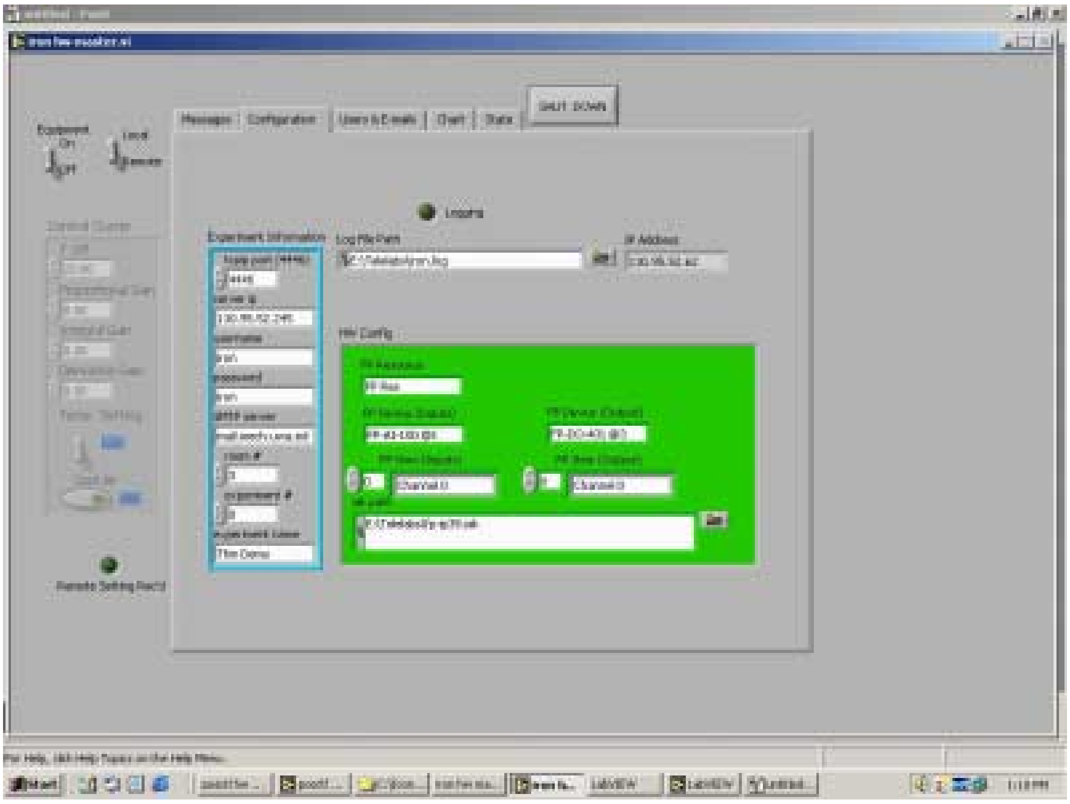

True
Period
U32
DBL
Input (0..1)
U32
Calculate the 'off' time needed for the next interval
Wait time
U32
Achieved Output
DBL
DBL
Output
TF
DBL
Memory in
Memory out
Calculate the actual level achieved over the last full period

Appendix C:

Telelabs Software:

Iron hw master.VI





Appendix D:

Laboratory Instruction

1. Aim of the Experiment

The main purpose of this experiment is to demonstrate simple feedback control and P-I controller using a domestic electric iron and simulation program.

2. Introduction

A domestic electric iron is a simple form of control system. It contains a control knob to set the internal temperature and a temperature-sensitive device (thermostat) switch that turns the heating element on and off according to the temperature setting. The controller works on the principle of on /off. The "switch on" temperature is usually a little less than the "switch off" temperature. For this difference to be small, given constant heating and cooling rates, the power will need to be switched on and off rapidly.

In this system, the temperatures of the electric iron are measured using RTD sensors and the system is controlled using LabView Virtual Instrumentation (VI).

3. Instrumentation

Sensors

RTD and Transmitter

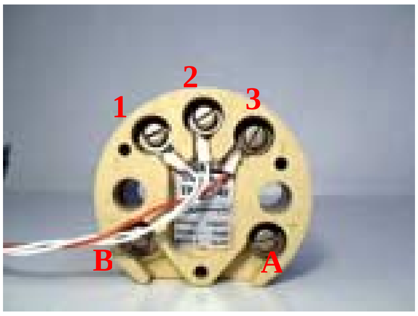


Figure 1. Temperature transmitter

Temperature of the electric iron is measured using RTD (Resistance Temperature Detectors). The RTDs are made from platinum, which have linear resistance-temperature characteristic.

A temperature transmitter (figure 1) measures the resistance changes in the RTD. It then processes these resistances and converts them into electric current between 4-20 mA. The temperature can then be determined from the linear relationship between the output electric current and Temperature. For the transmitter to work properly, it needs a DC supply of 11.6 to 40V or a nominal voltage of 24V.

- Terminal 1,2 and 3 are wired to a RTD sensor. The resistance changes are measured from these terminals
- Terminal A is connected to the positive terminal of a DC 24V power supply.
- Terminal B is the output terminal for electric current of 4-20 mA.

3.2 Actuator

Solid state Relay

Figure 2. Solid state relay

The difference between the “switch on” and “switch off” temperature of the electric iron can be minimised using a solid state relay by turning the current on and off in a very short period of time (20 μsec).

The input used to control the SSR in this application is DC Voltage. By applying a voltage between 3-32 Vdc, the SSR is turned on and it is turned off when a voltage below 3 Vdc is applied.

- Terminal 1 and 2 are the terminals for the main AC 240V power.

- Terminal 3 and 4 are the input terminals used to turn the SSR "on" and "off".

Air Cylinder

Figure 3. Pneumatic Cylinder

In order to change the thermostat knob position of the electric iron, a spring return air cylinder is utilised (SMC CD J2 L6- 45SR). Two-temperature settings: Low and High Temperature are achieved by using this air cylinder. A solenoid valve controls the air supply to the cylinder. When the solenoid is turned on, the air drives the air cylinder that pushes the electric iron's temperature knob (high temperature). When the solenoid is turned off, no more air is in the actuator and the spring in the air cylinder pulls the shaft back to the original position (low temperature).

3.3 Controller

Virtual Instrumentation

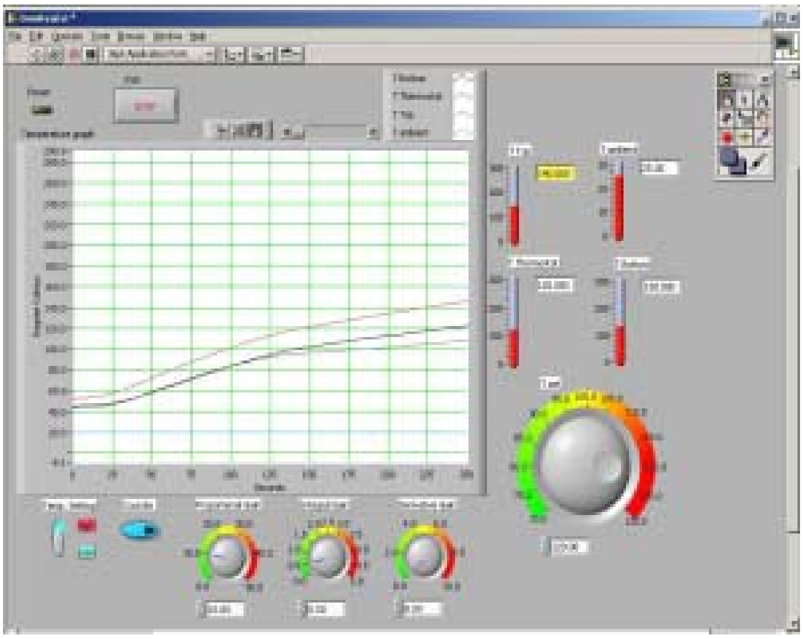


Figure 4. LabView VI

This program reads the temperature of the electric iron, plots them on the graph and displays them on the thermometer. The temperature knob can also be controlled from this program.

4. Procedure

For the temperature setting of 80^{o}C, it would take up to 15 minutes for the system to reach the steady state. Furthermore, it requires approximately 5 minutes to cool the system down using the cool air inlet. Therefore, one testing cycle for this system would take up to 20 minutes or more depending on the temperature setting.

Due to the time constrain, it is decided that only one test would be performed on the real equipment. In addition, the simulation program: IronSimulation.VI would be utilised to observe the behaviour of the system during the period where the actual system is running.

<u>Using the Real equipment</u>

Load the IronReal.VI. Before starting the experiment, make sure that you familiarise yourself with the VI and know what each button does.

1. Press the run button, which is located on the top left-hand corner of the program.
2. Set the temperature knob (Tset) to the desired value (a range between 70 – 100^{o}C is recommended, as this range would give the optimal steady state time).
3. Set the values for the proportional and integral gain.
4. After the system has reached the steady state, press the stop button.
5. A dialog box would appear asking you to save the experimental data into a file. Name the file and click the “save” button.
6. Set the temperature knob to the minimum level (20^{o}C) and push to “cool air” button to cool the iron down.

Using the simulation program

Load the IronSimulation.VI. Before starting the experiment, make sure that you familiarise yourself with the VI and know what each button does. After that, follow the instructions below:

1. Set the Integral gain to zero and the proportional gain to 5, 20 and then to 50. Observe the difference in the dynamic characteristic of the system.
2. Set the proportional gain to zero and the integral gain to 0.5, 2 and then 5. Observe the difference in the dynamic characteristic of the system.
3. Find the optimal values of PI that would give the least overshoot, Rise time and Settling time.

Observation table

Kp	Ki	Rise Time(s)	Overshoot(%)	Settling time (s)	S-S error (%)

5. Question

1. Fill in the table below according to your observation from procedure above using “increase”, “decrease”, “small change” and “eliminate”.

Response	Rise Time	Overshoot	Settling time	S-S error
Kp				
Ki				

2. State the optimal values of PI.
3. It is observed that the maximum temperature setting of the electric iron is 130^{o} C. Predict what will happen if the temperature is set above this maximum value, say 160^{o}C (Hint: Thermostat operating temperature for the “low” temperature setting is between 138^{o}C to 158^{o}C).

Appendix E:

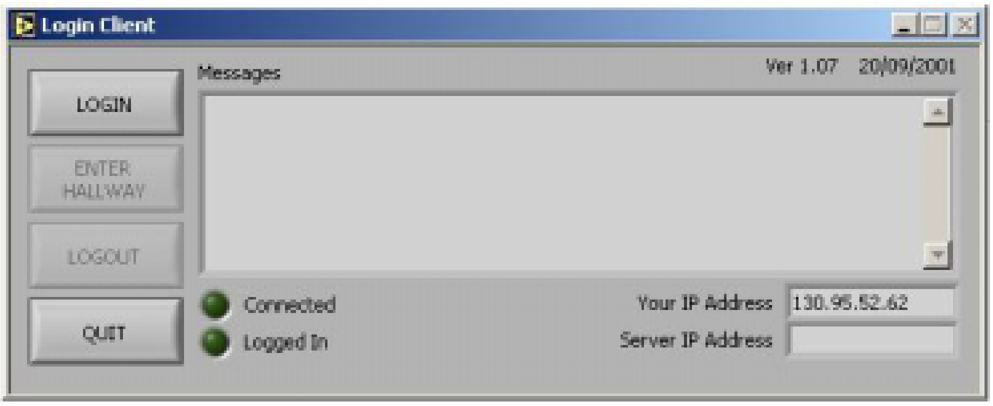
Setup Instruction

1 Set up Instruction

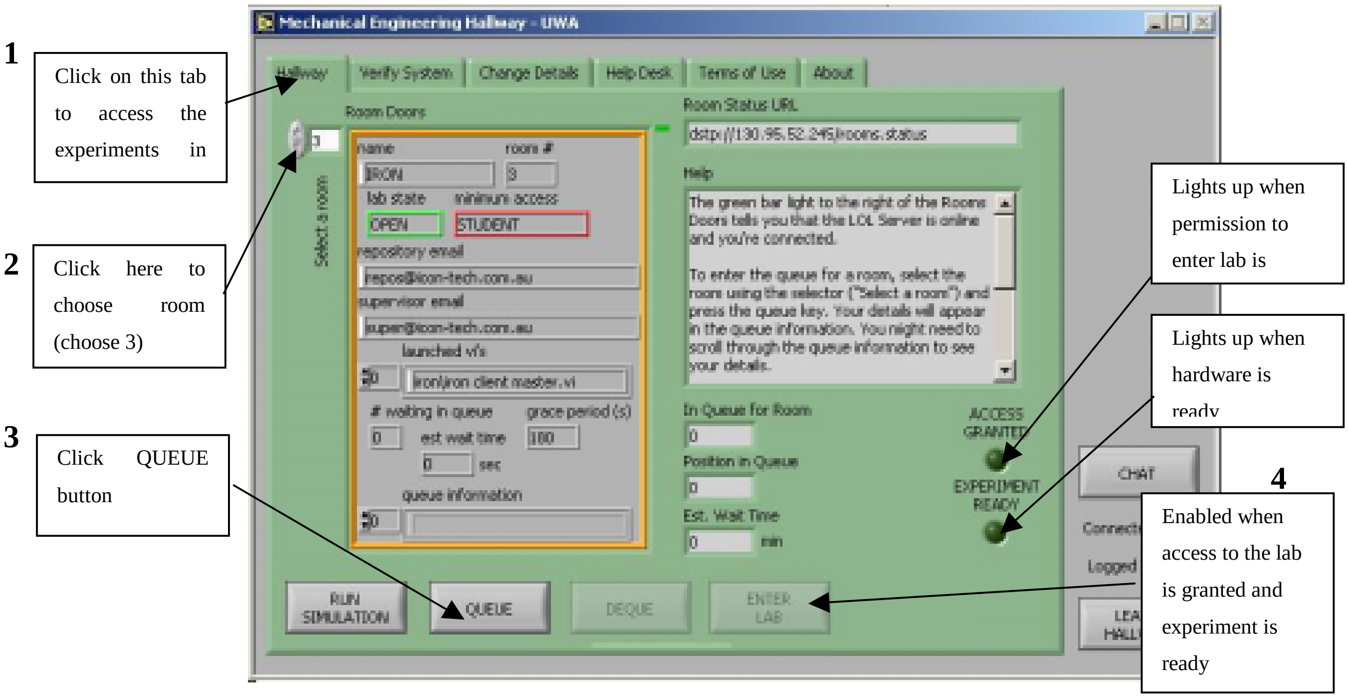
1. Go to the following site http://www.mech.uwa.edu.au/jpt/tele/Default.html.
2. Follow the instruction on the website.
3. Unzip lvrte602.zip and LOL-Client.zip.
4. Run the setup file “lvrt.msi”, you would need to enter the **password** “ni” into the program.
5. Once the LabVIEW Runtime engine and LOL-Client folder is copied, the program is ready to run.

2 Login

1. Go to the directory C:\Icons\LOL Client\
2. Click on “lol Client.exe” and the following panel should appear on your computer screen:



3. Click on the “LOGIN” button and the program would prompt you for a username and password.
4. Enter your username and password, then click the “OK” button. Once the correct username and password have been entered, the Logged In & Connected LED would light up and the “ENTER HALLWAY” & “LOGOUT” buttons are enabled.
5. Click on the “ENTER HALLWAY” button.
6. The Mechanical-Engineering-Hallway panel would appear on your computer screen showing the terms of use for the laboratory.

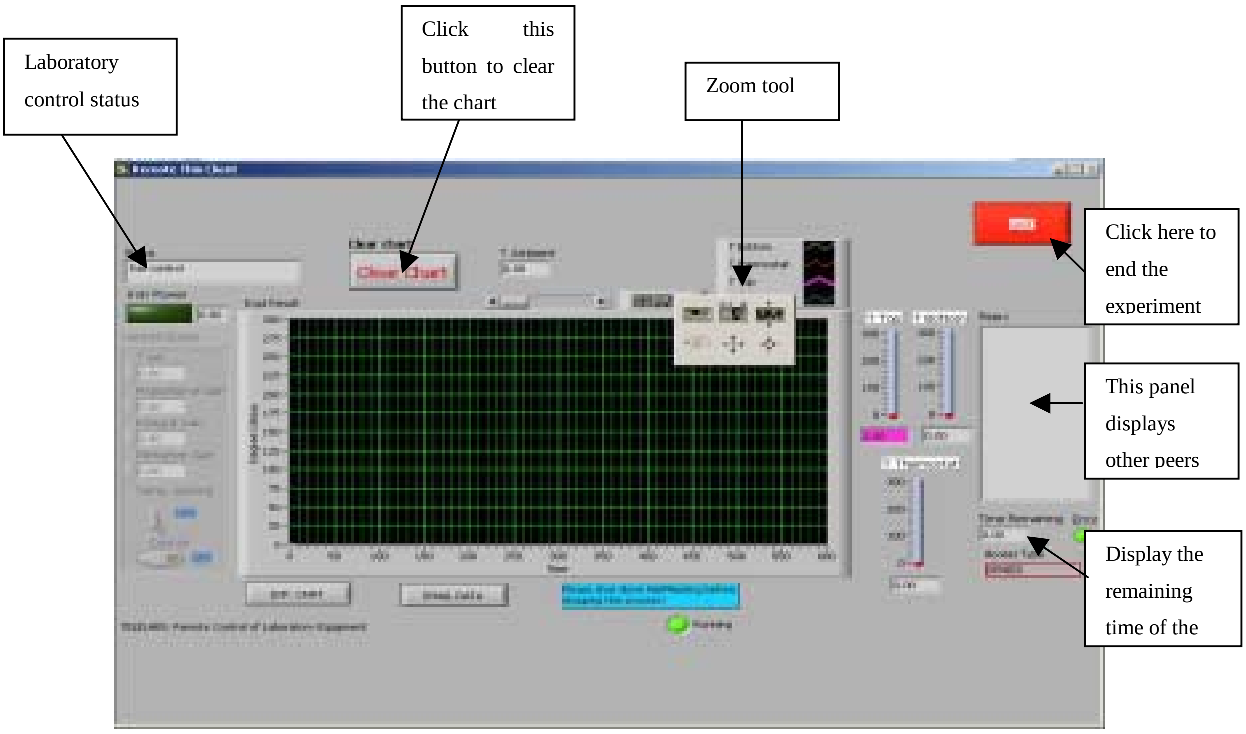


7. Read the term of use.
8. Click on the “Hallway” tab on the Mechanical-Engineering-Hallway panel.
9. Select the appropriate room by clicking on  or simply enter the room number into the box. The room number for the electric iron laboratory is 3.
10. Click on the “QUEUE” button and a panel would prompt you to decide how many minutes you want to use the experiment for (30 minutes is reasonable).
11. Click on “ENTER LAB” button to enter into the laboratory or “DEQUE” to cancel the lab reservation.
12. If “ENTER LAB” button have been pushed, you would see the Iron client master program appear on the screen.

Note:

1. *Students are in groups. When one group member accesses the laboratory. All other members of group already in queue go to operate the experiment at the same time.*
2. *If the estimated wait time is more than say, 20 minutes, you can logoff and reconnect in time to use the lab. If you lose your connection, your place in the queue is kept for you. However, if you don’t reconnect in time to use your slot, you may find you have gone back in the queue again.*
3. *By logging out of the system during the queuing process without pressing the “DEQUE” button, you can still keep your place in the queue.*

3 Iron Client Master



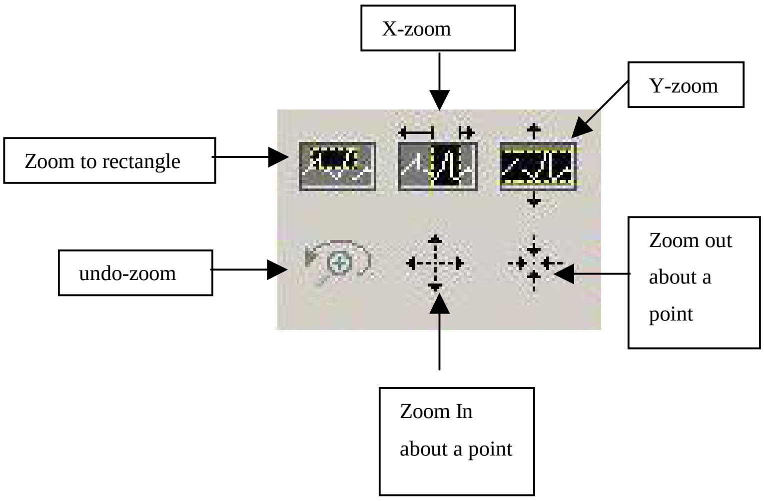
Iron Client Master

1. When the Iron Client master program starts, you should notice that it is in the "*You can control it*" status indicating that you can use the laboratory. Otherwise, the Control Cluster would be disabled, indicating that the laboratory is being controlled locally.
2. The temperature of the electric iron at the top & bottom of the plate and thermostat are displayed in the thermometers on the right side of the iron client master program.
3. The temperature graph displays the temperatures of the iron at various parts over time in second.
4. To set the temperature of the electric iron, enter a numeric value into the field "*T set*". The range for the temperature is set to 20^{o}C to 130^{o}C to avoid interference from the thermostat in the electric iron.
5. Enter a value between 0 – 50 for the proportional gain.

6. Enter a value between 0 – 5 for the integral gain. Due to the integral windup problem, the output power of the integral gain is limited to half of the total power. It can be observed from the power level on the right side of the power indicator light.
7. Enter any value for the derivative gain.
8. Enter a combination of PID values, such that the system has the fastest rise time and settling time, and smallest overshoot.
9. The temperature knob of the iron can be set by clicking on “*Temp. Setting*” switch.
10. The iron can also be cooled down by clicking on “*Cool air*” button.
11. Click on the “EMAIL DATA” button to email the result to your specified email address.
12. Alternatively, you can save the screen image by pressing “Print Screen” on your keyboard and paste the image onto a clipboard (eg: Microsoft Paint or Photo Editor).
13. After you have completed the experiment, please set the following:
    - *T set* to 20^{o}C
    - The *PID* controller to 0
    - *Temp. Setting* to low setting
    - *Cool air* to off
14. Click “QUIT” button to quit the iron client master.
15. To leave the system, do the following: Click the “LEAVE HALLWAY” button on the Mechanical Engineering Hallway, “LOGOUT” and “QUIT” buttons on the Login Client panel.

Tips in doing the laboratory:

1. *Always email the data of your result by clicking “email data” before clearing the chart.*
2. *Zoom tool becomes very handy when trying to read the data on the chart. Here is the some basic explanation for the zoom tool:*



* Zoom to Rectangle	Click a point on the display you want to be the corner of the zoom area and drag the tool until the rectangle covers the zoom area.
* X-zoom	Use this option to zoom in on an area of the graph along the x-axis.
* Y-zoom	Use this option to zoom in on an area of the graph along the y-axis.
* Zoom Out about Point	With this option, click a point you want to zoom out from.
* Zoom In about Point	With this option, click a point you want to zoom in on. Hold down the key to switch between Zoom In about Point and Zoom Out about Point.
* Undo Zoom	After you zoom in or out, use this option to return to the previous view.

Appendix F:

Student Feedback

Sent: Wednesday, 24 October 2001 12:34 PM

To: jamest@mech.uwa.edu.au; INVALID_ADDRESS@.SYNTAX-ERROR

Cc: mhodki@mech.uwa.edu.au; hro@tartarus.uwa.edu.au

Subject: PC307 Iron Lab Feedback

Dear all,

Firstly, the Iron lab that was put onto the web was beyond my expectation. At first, I did not expect to have the queue structure implemented as well as the chat room. Able to have a feature to email experimental data was unbelievable. It clearly shows what a great work Harjono has done!

Thus, I believe that Harjono should deserve a first class honours from the work he has done both to PC307 Lab and his final year thesis. PC307 would not be as interesting and exciting without his online lab. I am sure my other PC307 colleague will feel the same way.

However, it appears to me that the following files are missing during the download process: Iron client library.llb. I hope you can fix this before putting more frustrations (as it had happened to me a couple of days already) to my other PC307 colleague. Thanks to Harjono's help, I am able to get the lab going.

Some suggestions (I am not asking Harjono to do this, as I believe he has done a
lot of work already. Perhaps, next year's student?)

----

1) The data that I received from the email appears to me as a bunch of garbage. Some headings for each column would be appreciated. Also the data given was to 6 decimal places. This is too much! 1 decimal place should be sufficient.

2) Also in the Peers list, it apparently shows my name FOUR times although I have only log in ONCE. Perhaps a bug somewhere?

3) I would prefer an option to clear the chart and a button to stop the program. This is important as I may need to change the gain values without having to wait for the whole simulation to complete. For your information, the whole process takes ages! It appears to me that I have to log out of the system and log in again in order to halt the simulation to restart again. However, this has a disadvantage as this put me at the end of the queue again, and may have to wait again in order to proceed the experiment with a new gain value.

Other than that, the online experiment was well done and well built!

Lastly, can you tick my name off for completing the lab requirement for PC307. Thanks a million, Harjono.

Paul.