Louisiana State University

LSU Scholarly Repository

2004

Design and development of a process control temperature system for eastern oyster Crassostrea virginica research

Praveen Kolar
Louisiana State University and Agricultural and Mechanical College

Follow this and additional works at: https://repository.lsu.edu/gradschool_theses

Part of the Engineering Commons

**DESIGN AND DEVELOPMENT OF A PROCESS CONTROL TEMPERATURE
SYSTEM FOR EASTERN OYSTER *CRASSOSTREA VIRGINICA* RESEARCH**

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
In partial fulfillment of the
Requirements for the degree of
Master of Science in Biological and Agricultural Engineering

in

The Department of Biological and Agricultural Engineering

by

Praveen Kolar
B.Tech (Civil Eng.) S. V. University, 1991
M.Tech (Aqua Eng.) I. I. T, 1995
August 2004

Dedicated

To

My Parents

Kolar Venugopala Rao

And

Kolar Vidya

Acknowledgements

I express my sincere gratitude to Dr. Steven G. Hall for serving as my major professor and guiding me through out my graduate program. I thank my committee members Dr. Tiersch, Dr. La Peyre, Dr. Monlezun for their guidance and advice during research and analysis. Dr Tiersch guided technically and initiated my technical writing. Dr. La Peyre provided me with an opportunity to learn oyster biology and also gave me a lot of feedback. Thanks to Dr. Monlezun for statistical advice and timely suggestions. Special thanks to Dr. Supan for providing us with oysters for research and excellent lunches (whenever we went to the coast).

I sincerely thank Fernando (with whom I worked on this project) for his help in building our systems (even during weekends). I also thank graduate students at the farm, Patrice, Tyler, Roberto, Daisy, Alvaro, Jamie, Amogh, and Patro and graduate students at Biological and Agricultural Engineering (BAE), Jonathan, Navneet, Vishnu, David, Jeremy, Matt, and Sandeep for their help and tips. I extend thanks to Ms. Angela, Ms. Rhonda, Ms. Danielle, Mr. Tom Wagner, Mr. Tom McClure, Mr. Wayne Evans, the staff and faculty at BAE. Special thanks to Mr. Tom Bride who eagerly helped me whenever I needed. Thanks to staff at LSUARS, especially Ms. Jones. Dr. Romaire deserves special thanks for providing me with the laboratory space at LSUARS.

Finally, I thank the LSU Agricultural Center and National Sea Grant College Program for supporting me and our project.

Table of Contents

List of Tables

List of Figures`

Abstract

A process-control temperature system was designed and developed for facilitating somatic cell research on eastern oyster *Crassostrea virginica*. The system consisted of sixteen independent tanks capable of individual temperature regimes controlled by a central computer. Food grade 250-L tanks were used to hold 60 oysters per tank. Each tank was equipped with heating and cooling mechanisms. Immersion heaters were used for heating and a pump and a stainless steel heat exchangers immersed in chilled water (~ 7°C) were used for cooling. Type-T thermocouples along with a CIOEXP32 multiplexer board (Measurement Computing Inc.) were used to measure water temperatures. Control was achieved via a pair of analog and digital (A/D) boards that actuated heaters and pumps through solid state relays.

The performance of the system was evaluated in two experiments for studies on somatic cell proliferation in eastern oysters. In the first experiment, the system was programmed to maintain constant temperatures of 10, 15, and 20°C for a 20-day period. The system was able to maintain temperatures within ± 0.2°C. In the second experiment, diurnal fluctuations of 20 ± 1°C, 20 ± 5°C, 15 ± 1°C, 15 ± 5°C, and 15 ± 10°C were imposed. The system was able to simulate water temperatures within ± 0.2°C for regimes 20 ± 1°C, 20 ± 5°C, 15 ± 1°C.

A computer program *"Realwire"* for measuring and simulating real-time water temperatures was developed using freely available public domain software called Python and GNUwget. The software was able to measure and simulate real-time water temperatures.

Chapter 1

Foreword

The goal of this research was to develop process-control temperature systems to facilitate somatic cell research on the eastern oyster *Crassostrea virginica*. Such systems are expected to provide flexibility in planning experiments and control over environmental factors. Specific objectives within the scope of this thesis were to: 1) to design and develop a series of independent and automated time-temperature control systems, 2) to use the system for studies on somatic cell proliferation in eastern oysters, and 3) to develop real-time temperature-control software to simulate coastal water temperatures in the laboratory.

This research was a collaborative effort between the Louisiana State University Department of Biological and Agricultural Engineering (BAE) and the Aquaculture Research Station (ARS), of the LSU Agricultural Center. Portions of this project were presented at conferences and meetings (Table 1.1). The thesis was organized into six chapters. Chapter Two provides background on process-control technique and a history of automation in aquaculture and research, and a brief review on temperature effects on biology of oyster. Chapter Three covers the design and construction of the process control temperature system that was developed for eastern oysters. From an engineering point of view, this allowed for determination of the possible temperature ranges under which this system could operate. Chapter Four explains one possible application of the system in studies on somatic cell proliferation in oysters. From a biological point of view, this allowed for understanding of oyster responses under a variety of temperature regimes. Chapter Five deals with development of real-time temperature control software

called **_Realwire_** for use in oyster research. Conclusions are covered in chapter Six.

Detailed explanation of the data acquisition systems (DAS) boards, and their installation,

configuration, and calibration is provided in Appendix A. The software programs

developed for this project are provided in Appendix B. Standard operating procedures for

the system are provided in Appendix C. All chapters in this thesis are prepared in the

format of *Journal of Aquacultural Engineering*. Chapters Three Four were prepared for

submission to peer-reviewed journals.

Table 1.1 Conference presentations and abstracts within scope of this thesis.

Author	Title	Conference	Location
Kolar, P. and S. Hall., 2004.	Use of an automated temperature control system for studies on cell proliferation rates in eastern oyster *Crassostrea virginica.*	25th Annual Meeting of Louisiana Chapter of American Fisheries Society.	Baton Rouge, LA
Kolar, P. and S. Hall., 2004.	Design and testing of an automated Temperature control system for conditioning aquatic species.	Institute of Biological Engineering	Fayetteville, AR
Kolar, P. and S. Hall., 2003.	Instrumentation in aquaculture systems.	Annual meeting of Audubon Center for Endangered Species Resrearch	New Orleans, LA
Kolar, P. and S. Hall., 2003.	Process control oyster Recirculation systems	Louisiana Chapter of American Fisheries Society	Baton Rouge, LA

Chapter 2

Introduction

Automation and aquaculture research are intimately tied. Developments in automation engineering had direct implications on aquaculture research. Research associated with aquaculture has graduated from a laborious and manned process of data collection to automated data acquisition systems, analysis and control.

Use of automation in aquaculture research is not new. Simple mechanical devices based on timers and sensors were used for feeding of fish (Table 2.1). Advancement in technology and cheaper electronics, have taken automation to a higher level. Automatic electronic probes and meters have replaced manual methods of measuring water quality parameters in research laboratories worldwide. But it is the advent of computers that revolutionized automation especially in aquaculture research. Computer programs coupled with mechanical and electronic sensors were developed to monitor and record multiple water quality parameters and biological data of aquaculture animals. These systems were also able to activate alarm systems that dialed a telephone number (Table 2.2). However, some of these applications were semi-automatic and required manned activation.

The current trend of automation in aquaculture research is towards use of "intelligent", computer-controlled , automated systems popularly known as "process control systems" (Lee, 1995). Process control is roughly defined as an automatic control of a process, in which a computer system is used to regulate the continuous operations or processes.

4

Table 2.1 Mechanical and electrically operated fish feeders (dry, wet and live feeds) were used for feeding animals in laboratories (arranged chronologically).

Device description and species	Significant findings	References
Dry and moist fish feeder (Pacific Herring *Clupea harengus*)	Automatic and electrically driven.	Pozar, 1980
Dry-food fish feeder (General)	Automatic, electrically driven, and economical.	Falls, 1980
Live food feeder (Striped bass *Morone saxatilis*)	Continuous feeder and Economical	Nicholson, et al,1982
Demand feeders (Fingerling steelhead *Salmo gairderi*)	Automated and mechanical.	Tipping et al.,1986
Demand feeders (Blue tilapia *Tilapia aurea*)	Mechanical and inexpensive	Meriwether, 1986
Dry feed dispenser (General, all species)	Automatic , electro-mechanical,	Charlon et al., 1986
Feeder for fry and fingerlings (Striped bass *Morone saxatilis*)	Automated and low-cost	Parker, 1989
Feed dispenser (General, all species)	Mechanical, economical and efficient.	Mal, 1996

Table 2.2 Computer based monitoring systems were proposed and developed (arranged chrologically) for measurement of water quality parameters, fish sizes, feed consumption.

Devices used	Significant findings	References
Computer based monitoring.	Use of data acquisition systems was proposed.	Piedrahita et al., 1987
Computer based alarm system ALFA-LOG.	Used for hatcheries	Sanno et al., 1987
Computer automation with sensors.	Data acquisition system for aquaculture was developed.	Losordo et al.,1988
Computer-based measurement of biological data of fish.	Provided accurate measurements of fish.	Armstrong et al., 1989
Computer vision	Non-invasive size measurement and counting of fish was developed.	Naiberg et al., 1993; Petrell et al., 1993
Computer-based data systems.	Decision support systems for aquaculture research were developed.	Bourke et al., 1993
Computer and AMACS software.	Measured dissolved oxygen (D.O), pH, and temperature.	Munasinghe et al., 1993
Computer image analysis	Techniques for fish sizing and monitoring.	Ruff et al., 1995
Computer based monitoring.	Methods to measure and control water quality parameters were developed.	Rusch et al., 1993
Computers with sensors.	Decision support system for hatchery was designed.	Schulstad, 1997
Computer vision	Non-invasive control of feed dispensation was developed.	Petrell et al., 1997
Computer vision.	Feed consumption was monitored.	Foster et al., 1999
Computer vision and optics.	Parasitic influence was detected.	Tillet et al., 1999
Computer vision	In-vivo fish sorting techniques were developed.	Zion et al., 2000
Computer based feeding systems.	Online feeding method for sea bream.	Papandroulakis et al., 2001

In other words, it is a technique of automatic measurements and actions within a process, to ensure that output conforms to required specifications. The following section provides basic understanding of a simple process control design.

2.1 Process Control Basics

Most common type of process control is based on "closed loop" systems (Figure 2.1). The steps involved in such loops are:

1. A variable such as temperature, pressure, or flow is measured by a sensor, such as thermocouple, strain gauge, or flow meter.

2. The measured value of the variable is transmitted to a controller such as, a computer.

3. The controller performs two functions: a) compares the measured value with the desired value (or set point), and b) reduces the difference (error or offset) between measured and desired values of the variable. After making a decision, the controller conveys the decision to an actuator such as heater or valves.

4. The actuators manipulate the variable (heating or opening valves), till the error is minimized.

Further, controllers are classified into three types depending upon the nature of application. They are:

1. Proportional control in which the controller measures the error (explained above) and produces an output that is directly proportional to error. For example, if the error (difference between desired and actual variable say, pressure) is large in a system measuring air pressure, the output signal will allow for complete opening of valve to increase pressure in the system.
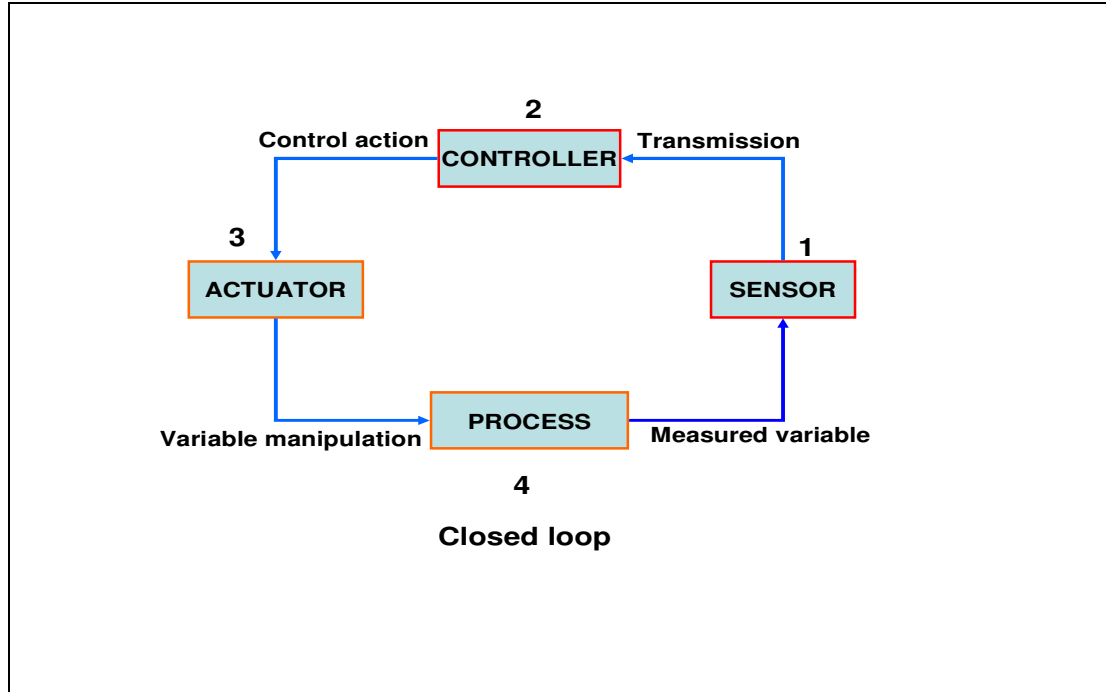
Figure 2.1 Schematic for a closed-loop system. The system automatically measures and controls variables and reduces the errors.

2. Integral control in which the controller provides an output that is related to the time integral of the error. In other words, output depends on size of the error and duration (time) of the error. For example, for the system (mentioned above) measuring pressure, output will depend on the amount of error and for how long was this error was active. If the pressure measured was less than desired pressure for a considerable amount of time, valve is opened fully to minimize the error or vice versa.

3. Derivative control, on the other hand, depends on the rate of change of error in the system. The response (output) of the controller is proportional to the rate

of change of the error. For the same example (above), if the error if increasing

with time, the system will allow the valve to open proportionally.

Depending on the nature of   application, one or more combinations are used. For

example, a controller which uses all three types of control is called proportional integral

derivative (PID) control. Typical applications of PID controllers include food processing

(milk pasteurization, ice creams, etc) and chemical based processing industries (Lee,

1995)

Process control systems are traditionally used in manufacturing and chemical

industries. They gained acceptance and popularity because of following reasons:

1) Process control systems were usually 100% automated at least for routine operations,

thereby reduced the need for manpower.

2) Because the process was continuously monitored, the resulting products from the

process were consistent and of superior quality.

3) Process control systems operated 24 hours a day and seven days a week, maximized

production, reduced overheads, and increased profit margins.

The use of process control systems for aquaculture research stems from the idea that

processes involved in an aquaculture system and any typical manufacturing system are

similar (Lee 1995). A comparison of steps and an analogy between aquaculture process

and any typical manufacturing process are provided (Table 2.3). Similar to any

manufacturing facility, inputs in aquaculture like temperature, oxygen, and pH may be

manipulated to obtain desired outputs.

Table 2.3 Steps involved in aquaculture are similar to any typical manufacturing process. Process control engineering may be extended to aquaculture, to "custom-produce" animals for research.

Steps	Typical manufacturing facility	Aquaculture
Inputs	Raw material, pressure, temperature etc	Fish, feed, salinity, temperature, pH , dissolved oxygen etc.
Process	Treatment of raw material, quality testing, packing etc.	Feeding, growth, respiration, spawning etc.
Outputs	Finished products	"Ready" animals for research

However attractive process control appears to be, these systems cannot replace human intervention, especially when dealing with aquatic animals. Process control only helps to achieve better control, and to obtain better efficiency with greater consistency (Young, 1954). To be able to apply process control engineering to aquaculture, one must have an understanding of biology of aquatic animals.

Biology of fish is dependent on factors such as water quality, feed availability, and climatic conditions, especially temperature. Fish are also called "ectotherms", because the body temperatures are determined by the ambient temperature. Water temperature plays a role in biological and physiological functions such as growth, feeding, health, and metabolism (Table 2.4 and Table 2.5). Also, in natural waters, temperature can control the amount of feed available to aquatic animals. For example, warmer waters promote production of phytoplankton and zooplankton biomasses, which are consumed by fish and other higher order aquatic animals. Research has shown that higher temperatures reduce the solubility of oxygen in water, stressing the animals. Water quality parameters

such as ammonia (NH_3) concentration can become toxic depending on water temperature. For example, higher temperatures can increase the proportion of un-ionized ammonia (NH_3), which can affect the health of fish (Timmons, 2001). Because this project focused on development of systems for eastern oyster, understanding of oyster biology is necessary.

2.2 Oyster Biology

The eastern oyster *Crassostrea virginica* belongs to Phylum Mollusca and Class Bivalvia. These oysters are abundant in shallow and semi-enclosed water bodies. Adult oysters can tolerate a wide range of temperatures from -2°C to 36°C (Shumway 1996). Oysters survived a drop in water temperatures as high as 22°C (Loosanoff and Engle 1940). Oysters are found along the coast of the Gulf of Mexico, where water temperatures can vary greatly (Figure 2.2). In shallow waters, oysters can be frozen solid in winter and can be thawed and survive when water is available (Shumway 1996). In the life cycle of oyster, temperature is the most important factor that affects virtually every aspect of oyster biology including survival, feeding, and growth, gonadal development, and spawning, larval settlement, sex and disease outbreaks (Shumway, 1996).

The rate of change of temperature has a greater effect on the survival of oysters, than the actual temperature itself (Fingerman and Fairbanks, 1957; Shumway, 1996). Survival of oysters under buried conditions is also dependent on ambient temperature. For example, oysters survived (under buried condition) only for 2 days in summer when temperatures were greater than 25°C, whereas in winter, survival was extended over 5 weeks, when temperature dropped below 5°C (Downington, 1968). This was probably due to reduced metabolic rates at low temperatures.

11

Average Daily Water Temperature
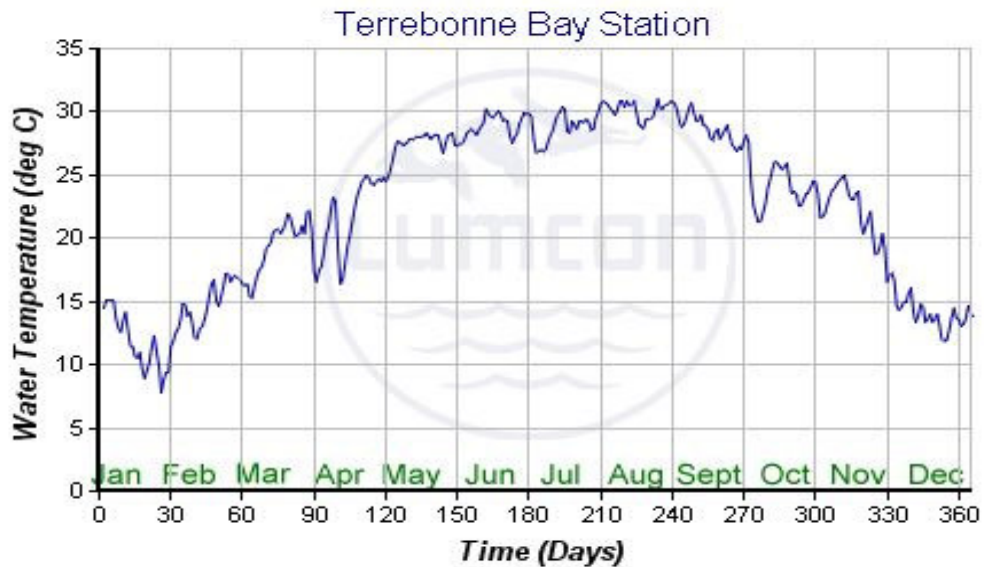
Figure 2.2 Water temperatures ranged from 8°C (January-February) to 31°C (July-August) at Terrebonne bay in 2003.

Oysters are active suspension feeders and consume particulate matter from water that is pumped through gills (Newell and Langdon, 1996). Pumping rate is affected by temperature (Shumway, 1996). Within temperature range of 8-28°C, pumping rate increased with an increase in temperature. For temperatures above 34°C and below 2°C, pumping rates were reduced (Shumway, 1996).

As a general rule, growth rate of larvae and adult oysters is affected by temperature (Shumway, 1996). Growth is rapid in warm waters of Gulf of Mexico, where marketable size (> 9 cm) is attained in about two years. On the other hand, in northern waters like Long Island Sound, the same growth takes about 4-5 years.

C. virginica is a protandric species, which means that they initially function as males and may change to females depending on environmental conditions like water quality,

temperature, etc (Thompson et al., 1996). However, oysters in unfavorable environment (for example, poor water quality) may not develop as females because functioning of female demands more energy for gonadal development and resisting unfavorable environments would limit energy for gametogenesis.

Gonadal development in oysters is correlated with water temperature (Shumway, 1996). Gametes start developing as water temperatures rise in spring and mature just before spawning (Thompson et al., 1996). Oysters are broadcast spawners. In nature, females and males release eggs and sperms into the water simultaneously, allowing for external fertilization (Thompson et al., 1996). This process of release of eggs and sperms is triggered by a stimulus and studies have indicated that under natural conditions this stimulus is associated with rise in water temperatures (Thompson et al., 1996; Shumway, 1996). Spawning is generally believed to occur when water temperature increases above 20°C.

Survival of sperm and eggs of oyster are temperature dependent. Sperm survival was higher at lower temperature (60 h at 20°C) when compared to higher temperatures (9 h at 40°C). Eggs also exhibited more tolerance to lower temperatures (6 h at 20°C) and less tolerance to higher temperatures (5 min at 40°C) (Shumway, 1996). Successful fertilization and larval development occurs between 15 and 30°C. The time between fertilization and larval development varies depending on water temperature (25 h at 15°C to 3h at 30°C) (Shumway, 1996). Although, there are other factors such as food supply, oxygen supply, larval settlement is predominantly affected by water temperature and most favorable temperature for settlement was between 19 and 24°C (Shumway, 1996).

In general, high temperature promote disease outbreak (Shumway, 1996). Higher temperature along with salinity was found to promote *Perkinsus marinus* infections in oyster (Ford, et al., 1996). Oysters that were infected with *P. marinus*, experienced 100 % mortality when held for four weeks at 28°C. However, when held at 15°C, infection and mortalities were negligible (Shumway, 1996). Incidence of Multinucleated Spore Unknown (MSX) disease was found to be low in low temperature conditions (Hofmann, et al., 2001).

The dependence of oyster on water temperature may be utilized to design and develop facilities for research. Such systems should:

1. Automatically control water temperature, based on a feedback and feed forward mechanism,

2. Be flexible to study or rear multiple aquatic species,

3. Consist of independent tanks to insure minimize water contamination and chance of disease spreading, allow multiple experiments, improve controllability, and

4. Maintain optimum water quality throughout the period of study or research.

This project envisioned the above constraints and a system was designed consisting of 18 independent recirculating tanks capable of automatic temperature control for oyster cell research. A personal computer (PC), type-T thermocouples, combination of analog and digital boards (A/D boards) were used as hardware, and Microsoft Visual C+ + was used as programming software for data acquisition and control. During initial testing of the system, the control over water temperatures was within ± 0.2ºC for temperatures between 10 and 25 ºC. After initial testing, this system was used for preliminary experiments on the eastern oyster *Crassostrea virginica* to evaluate temperature effects

on somatic cell proliferation rates. The following chapters (Chapters Three and Four)

describe the design, construction, testing, and application of the system for oyster

research.

Table 2.4 Water temperature played a role in biology of aquatic animals (references for each effect ordered chronologically).

Common name (Scientific name) and description	Temperature effects on	Significant findings	References
Orconectid crayfishes (*Orconectes virilis*) (*Orconectes immunis*)	Growth	Gains in weight and length increased significantly with increase in water temperature from 10ºC to 25C.	Wetzel and Brown, 1993
Leader prawns (*Penaeus monodon*)	Growth	Optimum growth occurred between 27º C and 32ºC.	Dearing, 1995
Juvenile sea bass (*Centropristis striata*)	Growth	Growth rates were significantly higher at 20, 25 and 30ºC than 15C.	Cotton et al., 2003
Large mouth bass (*Micropterus salmonides)*	Growth	Growth at 32ºC was significantly greater than 26ºC.	Tidwell et al., 2003
Juvenile sea horses (*Hippocampus whitei*)	Growth	Optimum growth rates were observed between 17 and 26ºC	Wong et al., 2003
White river crawfish (*Procambarus acutus acutus*)	Feed consumption	Feed consumption increased linearly over a temperature range of 5 to 25ºC.	Seals et al., 1997
Sea urchin (*Strongylocentrotos Franciscanus*)	Feed consumption	Feed consumption was significantly higher at 16.1C than 12.9C.	McBride et al., 1997
Pacific white shrimp (*Litopenaeus vannamei*)	Disease outbreak	Survival of shrimp infected with white spot syndrome virus (WSSV) was significantly higher when temperatures were increased form 25 ± 8ºC to 32 ± 3ºC.	Vidal et al., 2001
Red claw crayfish (*Cherax quaricarinatus*)	Spawning	Optimum spawning temperatures were around 30ºC.	Yeh and Rouse, 1995

Table2.5 Ambient water temperature played a role in biology of oyster.

Species	Temperature effects on	Significant findings	References
Flat oyster (*Ostrea edulis*)	Survival	Oxygen consumption increased	Haure et al.,1998
Pacific oyster (*Crassostrea gigas*)	Survival	Increased temperatures resulted in mass mortalities	Cheney et al., 2001
Calafia mother-of-pearl oyster (*Pinctada mazatlanica*)	Survival	Consumption of oxygen was temperature-dependent with optimum range of 23 and 28 ℃.	Saucedo et al., 2003
Eastern oyster (*Crassostrea virginica*)	Growth	Significantly higher growth rates were observed under higher temperatures regimes.	Chee et al., 1990
Pearl oyster (*Pinctada fucata*)	Growth	Shell growth increased with increase in water temperature	Tomaru et al., 2002
Pacific oyster (*Crassostrea gigas*)	Growth	Maximum growth was observed in summer	Gangnery et al., 2002
Calafia mother-of-pearl oyster (*Pinctada mazatlancia*)	Reproduction	Increased gametogenesis with increased temperature	Saucedo et al., 2001;
Pacific oyster (*Crassostrea gigas*)	Reproduction	Early gametogenesis With increased temperature	Saucedo et al., 2002

(table cont'd)

Pectinacean bivalve (*Spondylus tenebrosus*)	Reproduction	Gametogenesis increased with temperature.	Parnell, 2002
Chilean oyster (*Ostrea chilensis*)	Reproduction	Increased gametogenesis could be achieved at lower temperatures	Jeffs et al., 2002
Pacific oyster (*Crassostrea gigas*)	Reproduction	Gametogenesis was found to be correlated with temperature.	Ren et al., 2003
Pacific oyster (*Crassostrea gigas*)	Spawning	Spawning occurred when water temperature reached 23-25 deg C	Kang et al., 2003
Eastern oyster (*Crassostrea virginica*)	Disease	Increased risk of disease with higher temperature.	Hofmann et al.,

2.3 References

Armstrong, M. C., Nagtegaal, D. A., and Boutillier, J. A.,1989. Microcomputer-based measuring device for collecting biological data. The Progressive Fish Culturist. 51, 47-51.

Bourke, G., Stagnitti, F., Mitchell, B.,1993. A decision support system for aquaculture research and management. Aquacultural Engineering. 12, 111-123.

Boyle, W.A., Asgeirsson, A., Pigott, G. M., 1993. Advances in the development of a computer vision fish biomass measurement procedure for use in aquaculture. In: Techniques for Modern Aquaculture, Ed. J. K. Wang, ASAE, MI, 382-392.

Charlon, N., and Bergot, P., 1986. An improved automatic dry food dispenser for fish larvae. The Progressive Fish Culturist. 48,156-158.

Chee, Y. L., Wang, J. K., 1990. The effects of feed water flow rate on the growth of aquacultured *Crossostrea virginica* in Hawaii. Aquacultural Engineering. 9, 411-427.

Cheney, D., Elston, R., MacDonald, B., Kinnan, K., Suhrbier., 2001. The roles of environmental stressors and culture methods on summer mortality of Pacific oyster *Crassostrea gigas*. Journal of Shellfish Research. 20, 1195.

Connor, W. A. O., and Lawler, F. L., 2003. Salinity and temperature tolerances of embryos and juveniles of the pearl oyster *Pinctada imbricate Roding*. Aquaculture. 229, 493-506.

Cotton, F. C., Walker, R. L., Recicar, T. C., 2003. Effects of temperatute and salinity on growth of juvenile Black sea bass, with implications for aquaculture. North American Journal of Auaculture. 65, 330-338

Ciereszko, R. E., Dabrowski, K., Ciereszko., 1997. Effects of temperature and photoperiods on reproduction of female yellow perch *Perca flavescens*: Plasma concentrations of steroid hormones, spontaneous and induced ovulation, and quality of eggs. Journal of the World Aquaculture Society. 28, 344-356.

Chu Y. N., Chen, W.C., Chen, Y.H., 1993. A computerized system for integrated measurements of length and width of aquatic organisms. In: Techniques for Modern Aquaculture, Ed. J. K. Wang, ASAE, MI, 362-371.

Deering, M. J., Fielder, D. R., Hewitt, D. R., 1995. Effects of temperature on growth and protein assimilation in juvenile leader prawn *Penaeus monodon*. Journal of the World Aquaculture Society. 26, 465-468

Falls, W. W., 1980. Economical, automatic, dry-food fish feeder. The Progressive Fish-Culturist. April, 240.

Ford, S. E., Borrero, F. J., 2001. Epizootiology and pathology of juvenile oyster disease in eastern oyster *Crassostrea virginica*. Journal of Invertebrate Pathology. 78, 141-154.

Foster. M., Petrell, R., Ito, M. R., Ward, R., 1993. Detection and counting of uneaten food pellets in a sea cage. In: Techniques for Modern Aquaculture, Ed. J. K. Wang, ASAE, MI, 393-402.

Frost A.R., Saunders, K.G., Lee, S.R., 1996. The development of a remotely operated vehicle (ROV) for aquaculture. Aquacultural Engineering. 15, 6, 461-483.

Gangnery, A., Chabirand, J., Lagarde, F., Le Gall, P., and others. 2002. growth model of the Pacific oyster *Crassostrea gigas*, cultured in Thau lagoon. Aquaculture . 215, 267-290.

Grimsen. S., Jaques, R. N., Enest, V., Balchen, J. G., 1987. Aspects of automation in a lobster farming plant. In: Automation and Data Processing in Aquaculture, Ed. J. G Balchen. Pergamon Press. 221-224.

Hansen E., 1987. Computer aided control and monitoring of aquaculture plants. In: Automation and Data Processing in Aquaculture, Ed. J. G. Balchen. Pergamon Press. 187-192.

Haure, J., Penisson, C., Bougrier, S., Baud., J. P., 1998. Influence of temperature on clearance and oxygen consumption rates of the flat oyster *Ostrea eduli*: determination of allometric coefficients. Aquaculture . 169, 211-224.

Hofmann, E., Ford, S., Powell, E., Klinck, J., 2001. Modeling studies of the effect of climate variability on MSX disease in eastern oyster *Crassostrea virginica* populations. Hydrobiologia. 460, 195-212

Hurley, D, H., Walker, R, L., 1997. Effects of temperature and salinity upon larval growth, survival and development in hatchery reared Southern Atlantic surfclams *Spisula solidissima similis*. Journal of the World Aquaculture Society. 28, 407-411.

Jeffs, A. G., Dunphy, B. J., Wells, R. M.G., 2002. Experimental effects of water temperature on the gametogenic development of broodstock in the oyster Chilean oyster *Ostrea chilensis*. Journal of Shellfish Research. 21, 743-747.

Jorgensen L., 1987. An automated system for incubation of pelagic fish eggs. In: Automation and Data Processing in Aquaculture, Ed. J. G. Balchen. Pergamon Press. 211-213.

Kang, S. G., Choi, K. S., Bulgakov. A. A., Kim, Y., Kim. S. Y., 2003. Journal of Experimental Marine Biology and Ecology. 282, 1-21.

Lee G. P., 1993. Computer automation for recirculating aquaculture systems. In: Techniques for Modern Aquaculture, Ed. J. K. Wang, ASAE, MI, 61-70.

Lee. G. P., 1995. A review of automated control system in aquaculture and design criteria for their implementation. Aquacultural Engineering. 14, 3, 205-227.

Lee G. P., 2000. Process control and artificial intelligence software for aquaculture. Aquacultural Engineering. 23, 1-3, 13-36.

Losardo. T., Piedrahita, R. H., Ebeling, J. M., 1988. An automated water quality data aquisition system for use in aquaculture ponds. Aquacultural Engineering. 7, 265-278.

Lyon P. R., Arthurs, H. G., Hancock, D., Mullins, J., 1993. AMACS: A user friendly windows based aquaculture monitoring and controlling software. In:Techniques for Modern Aaculture, Ed. J. K. Wang, ASAE, MI, 81-88.

Mal, B. C., 1996. Performance of Hawaii-type automatic fish feed dispenser. Aquacultural Engineering. 15, 2, 81-90.

Martinez, G., Perez, H., 2003. Effect of different temperature regimes on reproductive conditioning in the scallop *Agropecten purpuratus*. Aquaculture. 228, 153-167.

McCarter, N. H., 1990. Inexpensive sensors for a hatchery alarm system. The Progressive Fish Culturist. 52, 136-139.

Meriwether, F. H., 1986. An inexpensive demand feeder for cage-reared tilapia. The Progressive Fish Culturist. 48,226-228.

Naiberg, A., Petrell, R. J., Savage, C. R., Neufeld, T. P., 1993. A non-invasive fish size assessment method for tanks and sea cages using stereo video. In: Techniques for Modern Aquaculture, Ed. J. K. Wang, ASAE, MI, 372-381.

Nicholson, L. C.,1980. A low cost live food feeder. The Progressive fish culturist. 245-248.

Oiested. V., Pedersen, T., Folkvord, A., Bjordal, A., Keenseth, P.G., 1987. Automatic feeding and harvesting of juvenile Atlantic cod (G*adus morhua L.*) in a pond. In: Automation and Data Processing in Aquaculture, Ed. J. G Balchen. Pergamon Press. 199-203.

Papandroulakis. N., Dimitris, P., Pascal, D., 2001. An automatic feeding system for intensive hatcheries. Aquacultural Engineering. 26, 3-26.

Parker, N. C., 1989. Low-cost automated feeder for fry and fingerlings. The Progressive Fish Culturist. 51, 42-46.

Parnell, P. E., 2002. Larval development, precompetent period, and a natural spawning event of the Pectinacean bivalve *Spondylus tenebrosus.* Veliger. 45, 58-64.

Pascual, C., Valera, E., Re-Regis, C., Gaxiola, G., Sanchez, A., Ramos, L., Soto, L, A., Rosas, C., 1998. Effect of water temperature on reproductive tract condition of Penaeus setiferus adult males. Journal of the World aquaculture Society. 29, 477-484.

Petrell R .J., Neufeld, T. P., Savage, C. R., 1993. A video method for non-invasively counting fish in cages. In: Techniques for Modern Aquaculture, Ed. J. K. Wang, ASAE, MI, 352-361.

Petrell. R. J and Ang, K .P., 1997. Control of feed dispensation in sea cages using underwater video monitoring: Effects of growth and food Conversion. Aquacultural Engineering. 16, 45-62.

Piedrahita. R. H., Ebeling, J. M., Losordo, T. S., 1987. Use of data acquisition systems in aquaculture. In: Automation and Data Processing in Aquaculture, Ed. J. G Balchen. Pergamon Press. 259-262.

Pozar, D. F., 1979. Automatic fish food dispenser for use with Oregon moist and dry food pellets. The Progressive Fish Culturist. July, 45-48.

Prilutzky, A., Birkan. V., Appelbaum, S., 1996. Automatic air pressure control in systems for intensive breeeding of aquatic animals. Aquacultural Engineering. 15, 3, 209-214.

Rana, K. J., 1997. Global overview of production trends. In: Food and Agriculture Organization (FAO) circular 886 FIRI/C886 (Rev 1).

Ren, J. S., Marsden, I. D., Ross, A. H., Schiel, D. R., 2003. Seasonal variation in the reproductive activity and biochemical composition of the Pacific oyster *Crassostrea gigas* from the Marlborough Sounds, Newzealand. 37, 171-182.

Ruff B. P., Marchant, J. A., Frost, A. R., 1995. Fish sizing and monitoring using a stereo image analysis system applied to fish farming. Aquacultural Engineering. 14, 5-173

Rusch, K. A., Malone, R. F., 1993. Bench-scale evaluation of micro-computer automated algal trubidostat. Journal of the World Aquaculture Society. 24, 379-389.

Rusch. K. A., Malone, A.F., 1993. A Micro computer control and monitoring strategy applied to aquaculture. In: Techniques for Modern Aquaculture, Ed. J. K. Wang, ASAE, MI, 53-60.

Saucedo, P. E., Rodriguez, P., Aldana, C., Monsalvo, P., Reynoso, T., Villarreal, T., Monteforte, M., 2001. Gonadic conditioning of the calafia mother-of-pearl oyster *Pinctada Mazatlanica* under two temperature regimes. Aquaculture. 195. 103-119.

Saucedo, P. E., Ocampo, L., Monteforte, M., Bervera, H., 2004. Effect of temperature on oxygen consumption and ammonia excretion in the Calafia mother-of-pearl oyster, *Pinctada Mazatlanica*. Aquaculture. 229, 377-387.

Sanno. J. O., 1987. ALFA-LOG-A computerized registering, control and alarm system for aquaculture. In: Automation and Data Processing in Aquaculture, Ed. J. G Balchen. Pergamon Press. 205-210.

Schulstad. G., 1997. Design of a computerized decision support system for hatchery production management. Aquacultural Engineering. 16, 1-2, 7-25.

Seals, C., Eversole, A, G., Tomasso, J. R., Petrosky, B. R., 1997. Effects of temperature on feeding activity of the White river crayfish *Procambarus acutus acutus*. Journal of the World Aquaculture Society. 28, 133-141.

Small, B, C., Bates, T, D., 2001. effect of low-temperature incubation on Channel catfish *Ictalurus punctatus* eggs on development, survival and growth. Journal of the World Aquaculture Society. 32, 189-194

Thomsen. R. D., Muench, K. A., 1987. A computerized data management design approach for aquaculture. In: Automation and Data Processing in Aquaculture, Ed. J. G Balchen. Pergamon Press. 255-257.

Tillett R. D., C. R Bull and J. A. Lines, 1999. An optical method for the detection of sea lice, *Lepeophtheirus salmonis*. Aquacultural Engineering. 21, 3, 133-48.

Tidwell, J. H., Coyle, S. H., Bright, L. A., VanArnum, A., Yasharian, D., 2003. Effect of water temperature on growth, survival, and biochemical composition of large mouth bass *Micropterus salmoides*. Journal of the World Aquaculture Society. 34, 175-183.

Timmermans, T., Sistler, F. E., Lawson, T.B., 1995. Predicting crawfish molting with machine vision. Journal of the World Aquaculture Society. 26, 234-239.

Timmons. M. B., Ebeling, J. M., Wheaton, F. W., Summerfelt, S. T., Vinci, B. J., 2001. Recirculating Aquaculture Systems, NRAC Publication, Itchaca, NY.

Tipping, J. M., Rathvon, R. L., and Moore, S. T., 1986. Use of demand feeders on large steelhead rearing ponds. The progressive Fish Culturist. 48, 303-304.

Tomaru, Y., Kumatabara, Y., Kawabata, Z., Nakano, S., 2002. Effect of water temperature and chlorophyll abundance on shell growth of the Japanese Pearl oyster *Pinctada fucata*. Aquaculture Research. 33, 109-116.

Twibell, R, G., Wilson, K, A., Brown, P, B., 1998. Effect of acclimation temperature and temperature changes on molting and survival of ablated and non-ablated crayfish *Orconectes virilis*. Journal of the World Aquaculture Society. 29, 23-30

Wetzel, J. E., Brown, P. B., 1993. Growth and survival of juvenile *Orconectes virilis* and *Orconectes immunis* at different temperatures. Journal of the World Aquaculture Society. 24, 339-343.

Wong, J. M., Benzie, J. A. H., 2003. The effects of temperature, *Artemia* enrichment, stocking density and light on the growth of juvenile seahorses, *Hippocampus whitei* from Australia. Aquaculture. 228, 107-121.

Young., 1954. Process Control, Instruments Publishing Company, Pittsburg, PA

Zhang, D., Lin, J., Creswell, R, L., 1998. Effects of food and temperature on survival and development in the Peppermint shrimp *Lysmata wurdemanni*. Journal of the World Aquaculture Society. 29, 471-476.

Zion. B, A., A Shklyar, and I. Karplus, 2000. In-vivo fish sorting by computer vision. Aquacultural Engineering. 22, 3, 165-179.

Chapter 3

Design, Development and Testing of a Variable-Temperature Control System for Studies on Eastern Oysters *Crassostrea virginica*

3.1 Introduction

The eastern oyster *Crassostrea virginica* is an important commercial fishery in the United States. But, the oyster production has declined in general throughout the United States due to disease, pollution and overexploitation (MacKenzie, 1996). This decline has initiated a research on eastern oyster especially in the areas of disease, genetics and reproduction (Buchanan et al., 1998). However, research on oyster in natural waters is limited, because of the time and money involved in transporting of oysters to the laboratory and acclimatizing them to the laboratory conditions. Also, the control over the water quality is not possible in natural waters. This is important especially for studies on effects of water quality on oysters. Development of indoor controlled systems is expected to reduce these problems and facilitate research on oysters. However, the first step in development of controlled systems for oysters involves understanding the biology of oysters and factors affecting biology of oysters.

The eastern oyster belongs to Phylum Mollusca and Class Bivalvia. These oysters are abundant in shallow and semi-enclosed water bodies across the United States. Adult oysters can tolerate a wide range of temperatures from -2°C to 36°C (Shumway 1996). However in the life cycle of oyster, temperature is the most important factor that affects virtually every aspect of oyster biology including survival, feeding, growth, gonadal development and spawning, larval settlement, sex, and disease outbreaks (Shumway, 1996).

Development of temperature control systems for oyster research is not new. Temperature-control recirculation systems were used for holding and conditioning of eastern oyster for research on gametogenesis (Buchanan et al. 1998). Temperature controlled systems were designed and developed for research on oyster diseases due to apicomplexan parasite *Perkinsus karlssoni* (MacMillan et al., 1994). Although these systems were used for holding oysters and studies on oyster diseases, improvements to the design are still possible. For example, contamination of water, which caused infection and disease in oysters, was not properly addressed by earlier systems. The earlier systems had multiple tanks with a single biological filter. This posed a disease threat due to mixing of water. This can be minimized by designing the independent tanks with independent biological filters. Similarly, 1) a series of independent tanks can allow statistically sound experimental designs and 2) independent tanks also allow temperature control and manipulation allowing for research under multiple and variable temperature regimes. Hence, modified control systems are needed which can accommodate these constraints. Such systems should have the following features:

1. Ability to replicate experiments: This requires a series of tanks to allow replication of experiments supporting standard statistically designed experiments.

2. Minimizing disease outbreak: This feature requires the tanks to be independent of each other to minimize mixing of water between tanks. If there is an occurrence of disease, or failure of tank system, the tank can be excluded, without affecting the overall research.

3. Maintain optimum water quality: Independent tanks may be ideal from disease-control perspective. However, this feature imposes each tank to have its own water filter and such filter was to be a part of the each tank.

4. Allow independent and variable temperature control: Temperature control over each tank is required. Since tanks are independent, separate heating and cooling for each tank is to be provided. The system should be able to hold the tanks at required temperatures and also impose variable temperature regimes.

The goal of this study was to develop temperature-controlled holding systems for research on eastern oysters. The specific objectives were to: 1) design and develop a temperature-control system capable of automatic temperature regulation, and 2) test the system for a variety of temperature fluctuations. The temperature control system developed was capable of controlling the water temperature to within ± 0.5°C and allowed for analysis of time-temperature responses in eastern oyster.

3.2 Materials and Methods

The following sections explain the construction, hardware customization, and software details for development of the system. A detailed description of installation, calibration and working of the electronics are given in Appendix A.

3.2.1 Design and Construction of the System

Sixteen cylindro-conical food-grade tanks each of 250 L capacity (Aquatic Eco-Systems, Inc, Apopka, FL) were used as holding tanks. All tanks were insulated with encapsulated fiberglass insulation (~10 cm) (Johns Manville, Denver, CO) to prevent heat loss. All tanks were completely independent from each other. General schematic of

tanks is shown in Figure 3.1. Each tank had separate heating and cooling mechanisms and had the following components (Figure 3.2):

1. An airlift, consisting of a 5-cm diameter and 80-cm height PVC tube, with a 90° bend at the top. Two 1.9-cm elbows were glued to the bottom of the airlift pipe to form a U-shaped air injection point. The air was injected via a 1.25-cm plastic tube connected to the air supply. The airlift performed two functions: aeration of water, and circulation of water in each tank.

2. Heating was provided by 300-W electric glass immersion heaters in each tank (Model # 300, Commodity Axis, San Gabriel, CA).

3. Chilling was provided when needed by circulating water through heat exchangers. A submersible water pump (Model # 306, Commodity Axis, San Gabriel, CA), recirculated saline water in the tank through the heat exchanger @ 2 lpm. The heat exchangers were constructed from stainless steel trays (54 x 33 x 6 cm) and lids sealed with silicone sealant and held in place with stainless steel screws. The heat exchangers were immersed in fresh water at 4°C in a chill tank of dimensions 2.1 x 0.55 x 0.5 m (Frigid Units, Toledo, OH). The airlift ensured uniform cooling of water in the tanks, by continuous mixing.

4. A set of two biological filters was placed at the bottom of each tank. The filter media consisted of plastic bio balls (5 cm dia), with a void space of approximately 94% and geometric surface area 31 ft^2/ft^3 (Keeton Industries, Inc., Wellington, CO). The filter media was placed in perforated plastic sacs.
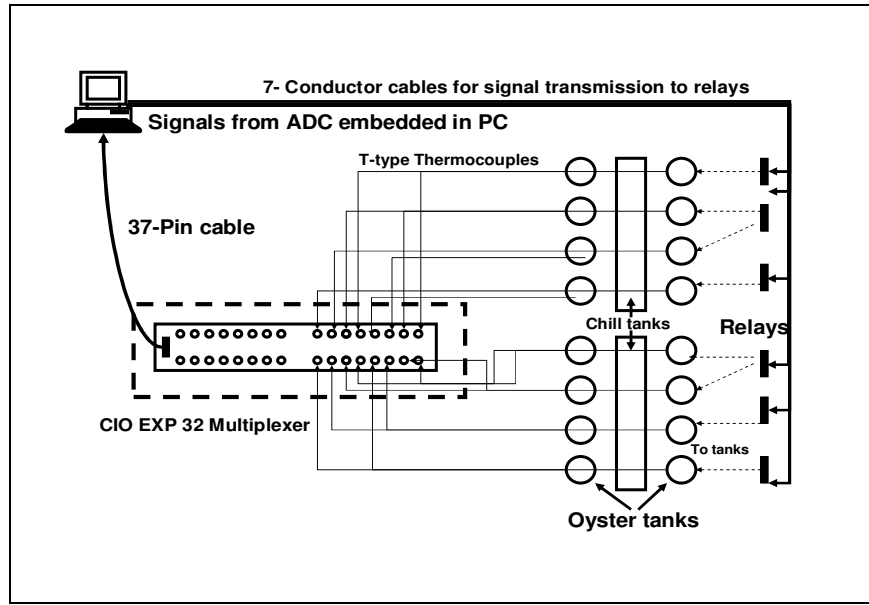
Figure 3.1 General schematic of the system developed at the Louisiana State University Aquaculture Research Station. Thermocouples acted as temperature sensors and all thermocouples were connected to a 32-channel multiplexer. A personal computer processed the signals and relayed the output signals to the solid state relays that actuated the pumps and heaters.
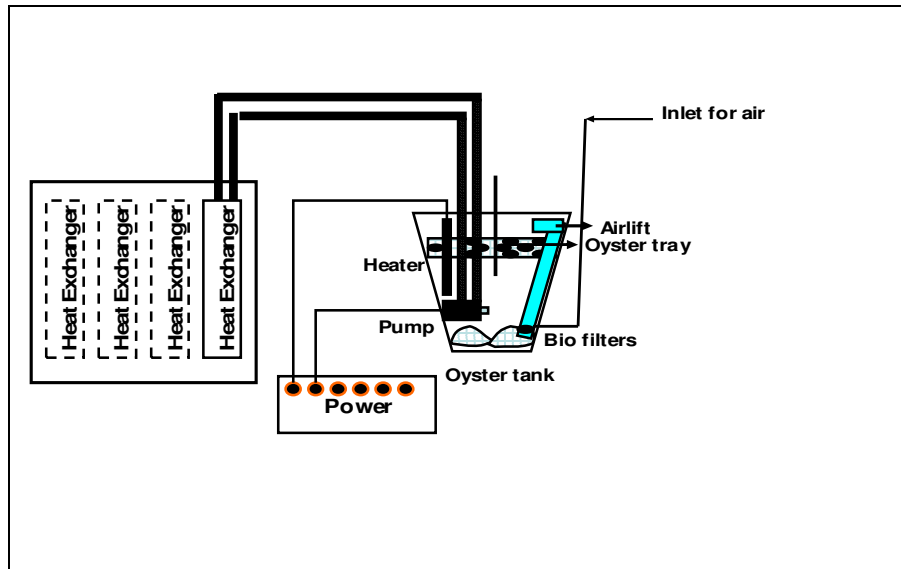
Figure 3.2 Tank components included a 5.0-cm airlift to aerate and circulate water in the tank, a 300-W immersion heater and 30 GPH water pump were used for temperature control. Biological filters were placed at the tank bottom to treat wastes generated by oysters.

Eight tanks were connected to one rectangular chill tank (0.57 m^3) (Figure.3.3) Water in each chill tank was cooled by a pair of electrically operated chillers (120 V AC and 2000 W approximately). The heat exchangers remained submerged in the chilled water and submersible pump recirculated water in the oyster tank through the heat exchanger.
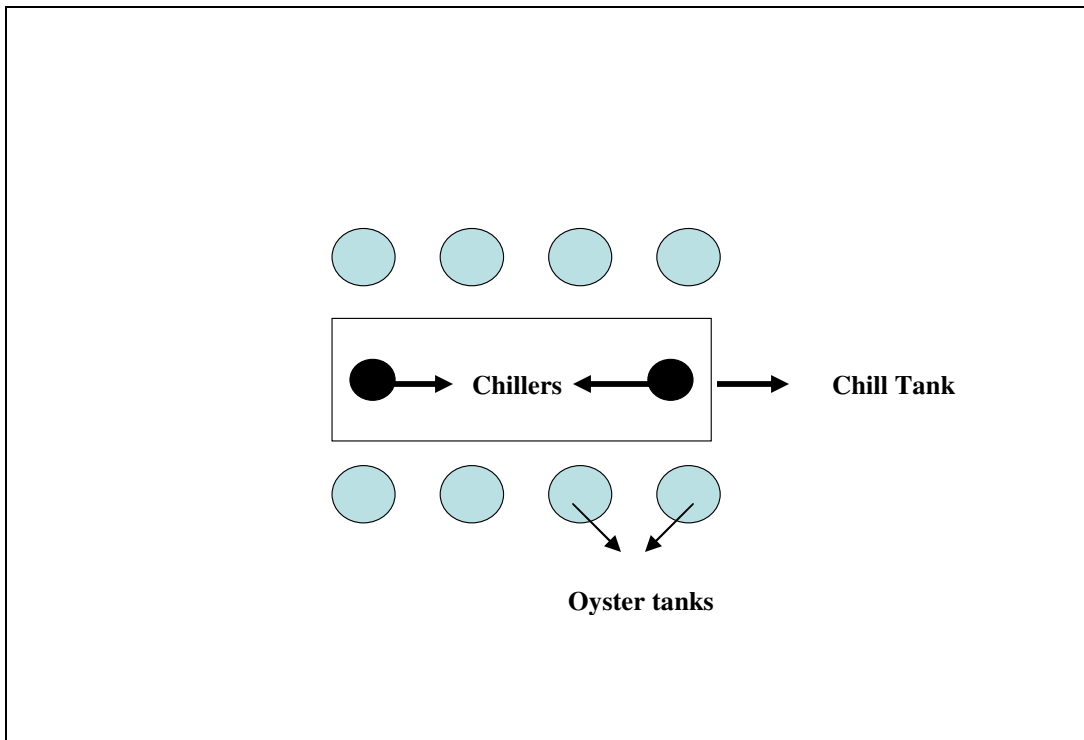
Figure 3.3 Schematic of the chill tank with chiller. The chillers are connected to 120 V AC and cool the water to 5 °C.

3.2.2 Hardware Design and Customization

The control hardware of the system included a 32-channel CIO EXP-32 multiplexer board, two 8-channel PCIM-DAS 1602/16 analog-to-digital converter boards (ADCs) (Measurement Computing Inc., Middleboro, MA.), T-type thermocouples (Omega Engineering Inc., Stamford, CT) and a set of 15 A 4-28 VDC solid-state relays (P/N 611489, Eastman Kodak Co).

Control hardware connections and sequence of operations are shown in Figure 3.4 The hardware and software of the system were customized to process analog and digital data. The junction ends of the thermocouples were soldered with lead-free electric solder (Radio Shack Inc.) and were made water resistant using silicone sealant. The measuring ends were connected to each of the first 16 analog input channels of a CIO-EXP 32 multiplexer board. The temperatures were read as differential voltages and the multiplexer amplified the voltage signals by a factor of 100. A personal computer (PC 486), with embedded ADC (Figure 3.2) processed the signals. The multiplexer was connected to the ADC via a CEXP2DAS16-10 special 37-conductor cable (Measurement Computing Inc., Middleboro, MA).

Based on the control software, 5 V DC signals were transmitted by the ADC through a BP 40-37 (Measurement Computing Inc.) ribbon cable that brought 40-pin on-board connections out to 37-pin male connector on computer back plate. TwoC37 FF-2 (Measurement Computing Inc.) ribbon cables were used for transmitting output signals. One end of C37 FF-2 cable was connected to male pin of BP 40-37 and other ends were soldered to 8-pin standard male DIN connector (Marlin P. Jones & Associates, Inc.,).

Seven-conductor cables were used to connect C37 FF-2 cables. Both the ends of the 7-conductor cables were soldered with 8-pin standard female DIN connectors (Marlin P. Jones & Associates, Inc.,). One end of the 7-conductor cable was connected to the relays and other end was connected to C37 FF-2 cables as shown in the wiring diagram (Figure 3.5). The pin numbers, port numbers and corresponding controls are shown in Table 3.1. The output signals (5V DC) closed the 110V AC circuitry and actuated pumps and heaters in the tanks as dictated by software.

PC with PCIM DAS 1602/16 ADC

CIO EXP 32
Multiplexer

T-type Thermocouple

Software | Algorithm

Relays

Oyster tanks
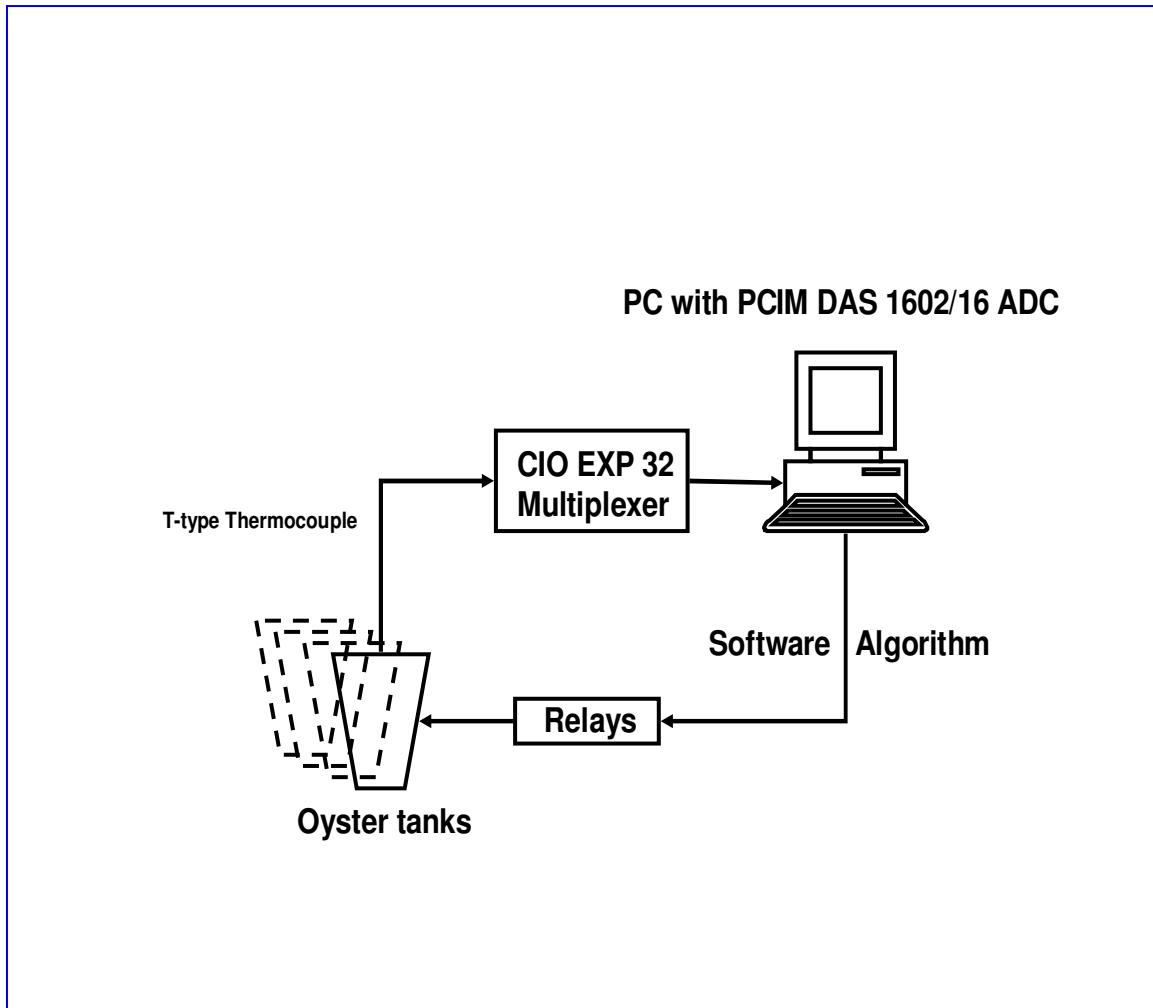
Figure 3.4 Type-T thermocouples from each tank were connected to a 32-channel multiplexer board. The inputs from the multiplexer were processed by the PC embedded with two 8-channel ADCs. The software allowed the ADC to send out a 5 V DC signal to the 3-32V DC solid-state relays to actuate the pumps and heaters. The signals were transmitted to the relays via a 7-conducter cable.

3.2.3 Software Design and Logic

The software for the control system was developed in Visual C++ (Version 6.0,

Microsoft Corp, WA) executed on a Windows NT based platform. The control was

executed every 3 min. The program measured the temperature every 3 sec and average

calculated over 3 min for each tank was computed.
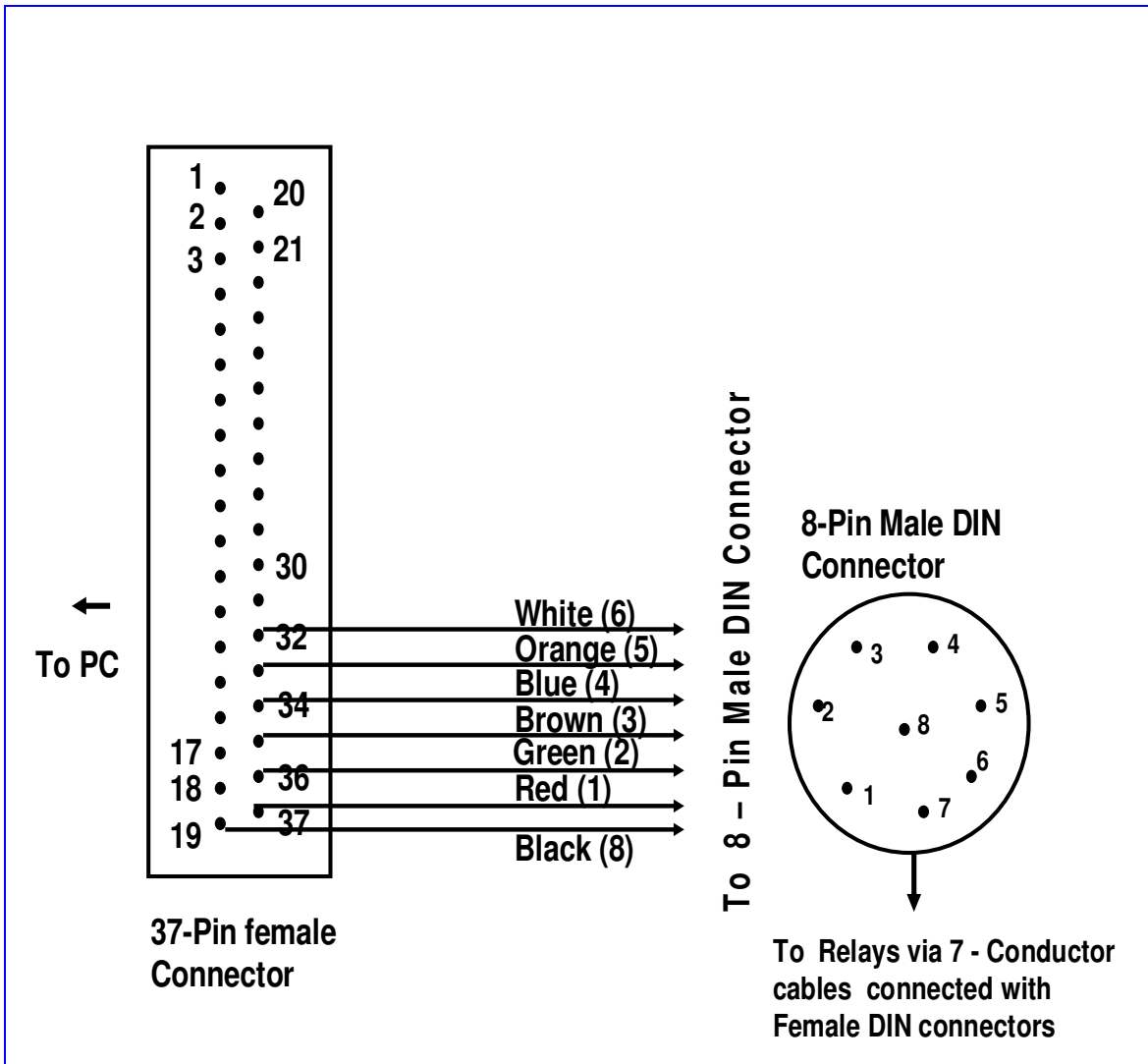
Figure 3.5 Wiring diagram of the connections from PC to the relays. A 37-pin female cable transmitted the signals to the relays through 7-conductor cables that were connected with 8-pin DIN female connectors. The order of the colors and corresponding pin number may be noted.

The computed average was compared with the desired temperature. For every execution

cycle, one of the following conditions was satisfied.

1) If the actual temperature was above the required temperature, the pump was switched

   on to cool the water.

Table 3.1 Connections between 40-pin PCIM boards and CEXP2DAS16-10 special 37-conductor cable connectors.

Board	Pin #on board (40 pins)	Pin # on 37 female connector	Port	Color code for 8-conductor cable from the relays	Controls
1	36	37	A0	A1 (Red)	Pump 1
1	34	36	A1	A2 (Green)	Heater 1
1	32	35	A2	A3 (Brown)	Pump 2
1	30	34	A3	A4 (Blue)	Heater 2
1	28	33	A4	A5 (Orange)	Pump 3
1	26	32	A5	A6 (White)	Heater 3
1		19	Ground	Ground (Black)	
1	19	10	B0	B1 (Red)	Pump 4
1	17	9	B1	B2 (Green)	Heater 4
1	15	8	B2	B3 (Brown)	Pump 5
1	13	7	B3	B4 (Blue)	Heater 5
1	11	6	B4	B5 (Orange)	Pump 6
1	9	5	B5	B6 (White)	Heater 6
1		11	Ground	Ground (Black)	
1	20	29	C0	C1 (Red)	Pump 7
1	18	28	C1	C2 (Green)	Heater 7
1	16	27	C2	C3 (Brown)	Pump 8
1	14	26	C3	C4 (Blue)	Heater 8
1		13	Ground	Ground (Black)	
2	37	37	A0	D1 (Red)	Pump 9
2	36	36	A1	D2 (Green)	Heater9
2	35	35	A2	D3 (Brown)	Pump 10
2	34	34	A3	D4 (Blue)	Heater 10
2	33	33	A4	D5 (Orange)	Pump 11
2	32	32	A5	D6 (White)	Heater 11
2	19	19	Ground	Ground (Black)	
2	10	10	B0	E1 (Red)	Pump 12
2	9	9	B1	E2 (Green)	Heater 12
2	8	8	B2	E3 (Brown)	Pump 13
2	7	7	B3	E4 (Blue)	Heater 13
2	6	6	B4	E5 (Orange)	Pump14
2	5	5	B5	E6 (White)	Heater 14
2	11	11	Ground	Ground (Black)	
2	29	29	C0	F1 (Red)	Pump 15
2	28	28	C1	F2 (Green)	Heater 15
2	27	27	C2	F3 (Brown)	Pump 16
2	26	26	C3	F4 (Blue)	Heater 16
2	21	21	Ground	Ground (Black)	

2) If the actual temperature was below the required temperature, the heater was switched on to heat the water.

3) If the actual temperature was equal to required temperature, heater and pump were switched off.

Depending on the actual temperature of water in each tank, the water was either cooled or heated, until desired temperatures were reached (Figure 3.6). Copies of the software programs used in this project are attached in Appendix B. The system was programmed to hold constant temperatures or follow sinusoidal temperature variations and could be programmed to follow complex time-temperature histories if needed.
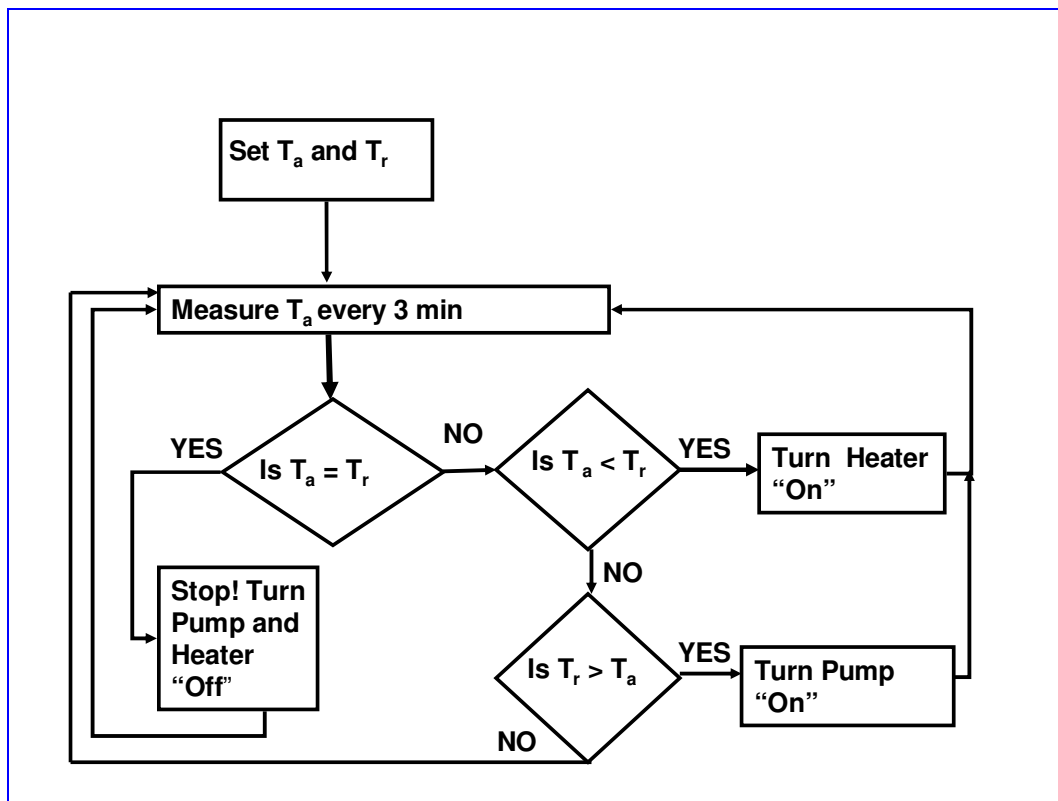
Figure 3.6 Programming logic for the software consisted of three scenarios. The water was either cooled or heated. If the measured temperature was same as the required temperature, pumps and heaters were inactivated.

3.3 Testing of the System:

All thermocouples were calibrated prior to testing and corrective coefficients were incorporated into the program. Two experiments were conducted to test the system for temperature control. In the first experiment, the system was tested for a 20-day period to hold constant temperatures in the tanks. Three temperatures, 10°C (Four replicates), 15°C (Six replicates), and 20°C (Four replicates) were selected.

In the second experiment, diurnal temperatures were simulated for an 8-day period in the tanks. The average temperatures were fixed at 10°C, 15°C, and 20°C. A diurnal fluctuation of ± 5°C was imposed on each average temperature (two replications each). Also, to evaluate the physical limits of the system an additional fluctuation of ± 10 °C was imposed over 15°C. Water quality parameters, ammonia, nitrite, and nitrate were measured using HACH test kits after one, three, and seven days of temperature fluctuations for 10± 5°C ,15 ± 5°C, 15 ± 10°C, 20 ± 5°C.

3.4 Results

Figure 3.7 shows the layout of the tanks with supporting electronics.

Actual daily average temperatures in the tanks for a 20-day period maintained at 10°C are plotted as shown in Figure 3.8. Because the system monitors the water temperatures every three minutes, each day will consist of 480 temperature readings and over a 20-day period, the number of observations per tank were 9600 (480 x 20) and data management becomes complicated. For simplicity, an average temperature over a 24-hour period was computed for each tank and those daily averages were used in data analysis.

Figure 3.7 Photograph of the system developed at the LSU Aquaculture Research Station. All tanks were insulated using encapsulated fiberglass insulation to prevent heat loss.

The actual daily average temperatures (solid lines) in all four tanks were within ±0.5°C from the expected temperatures (dashed lines). Around 10/31/2004, the temperatures in tanks 1 and 2 experienced a steep increase.

Daily average temperatures in tanks (replicates 1 through 6) for a 20-day period at 15°C are shown in Figure 3.9. The average daily temperatures were within ±0.5°C from the expected temperatures (dashed lines).

A 20-day temperature simulation in tanks (replicates1 through 4) at 20°C is shown (Figure 3.10). The average daily temperatures were within ±0.5°C from the expected temperatures (dashed lines).
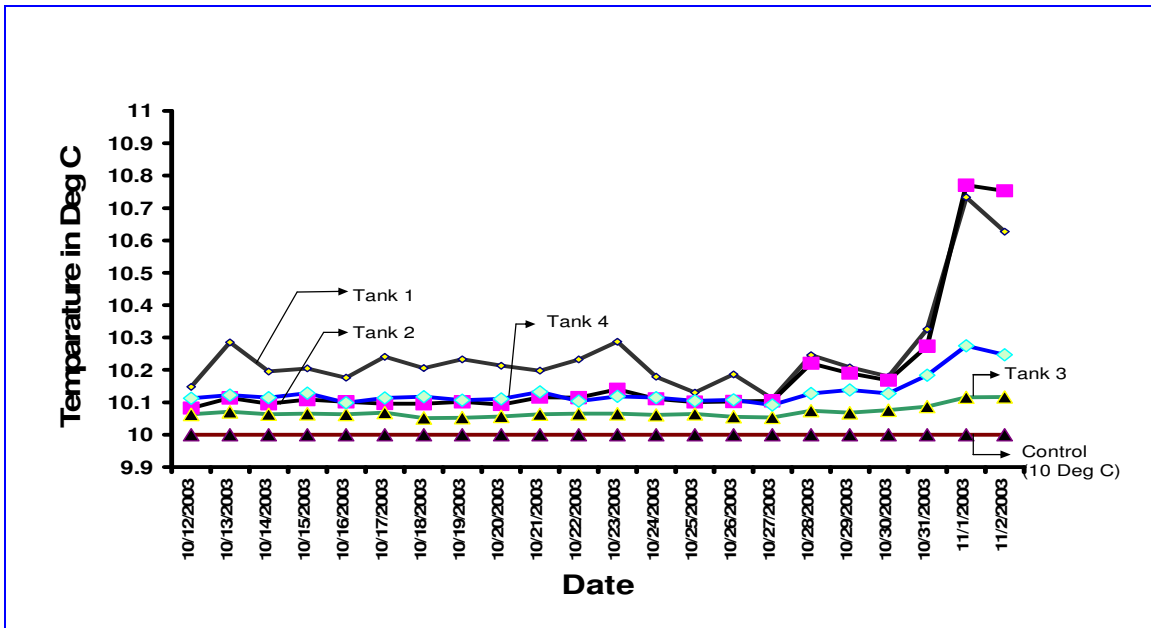
Figure 3.8 Tanks were programmed to maintain constant temperatures of 10°C. The system was able to maintain the temperatures within ± 0.5°C. Around 10/31/ 2003, the power bars were tripped off for two tanks (Tank 1 and Tank 2) due to electrical overloading.
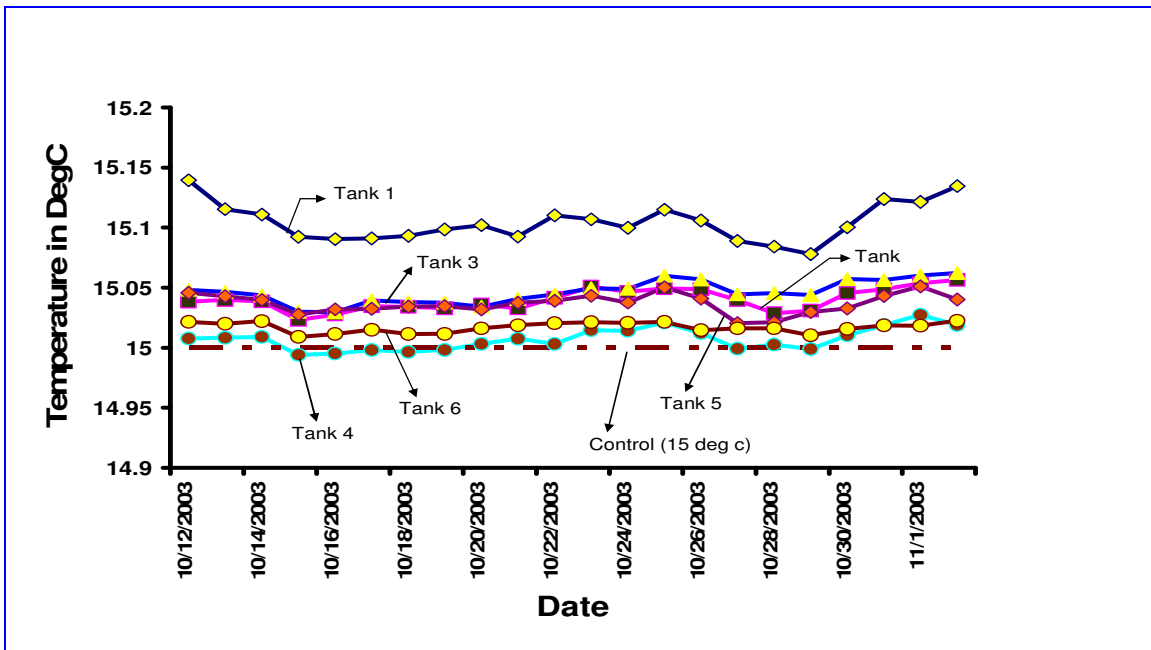
Figure 3.9 Tanks were programmed to maintain constant temperatures of 15°C. The tanks were able to maintain the temperatures within ± 0.5°C.
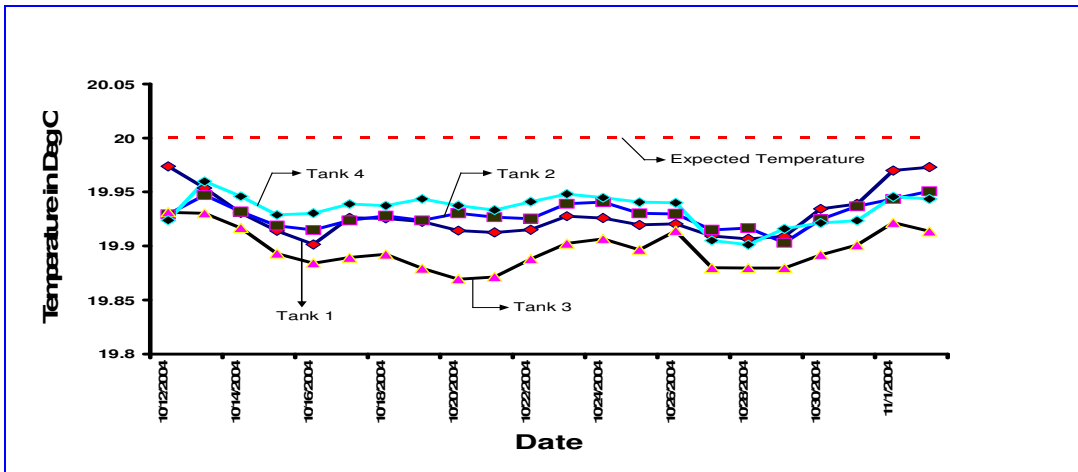
Figure 3.10 Tanks were programmed to maintain constant temperatures of 20°C. The tanks were able to maintain the temperatures within ± 0.5°C.

Diurnal simulations for a typical 8-day period are shown in Figure 3.11. The average temperature was set at 20°C and a variation of ± 5°C was imposed. The actual temperatures were within 0.5°C from the expected temperature. A sample 8-day diurnal simulation for 10 ± 5°C is shown (Figure 3.12). Figure 3.13 shows diurnal simulation for the temperature regime 15 ± 5°C and Figure 3.14 for the temperature regime 15 ± 10°C. Ammonia was found to be less that 0.5 ppm in all four regimes, 10± 5°C, 15 ± 5°C, 15 ± 10°C, 20 ± 5°C (Table 3.2).

Table 3.2 Despite imposing fluctuations ammonia remained below 0.5 ppm

Temperature	10 ± 5			15 ± 5			15 ± 10			20 ± 5		
Days	1	3	7	1	3	7	1	3	7	1	3	7
Ammonia (ppm)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Nitrate (ppm)	0	0	0	0	0	0	0	0	0	0	0	0
Nitrate (ppm)	-	80	100	-	80	100	-	80	100	-	80	100
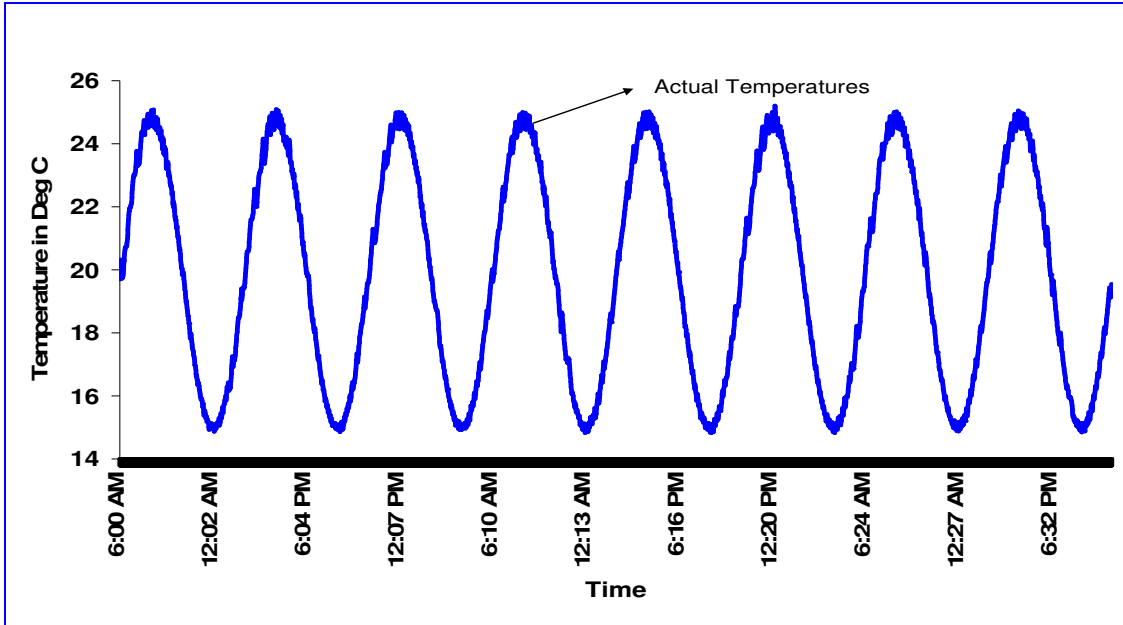
Figure 3.11 Diurnal temperatures were simulated in the tanks for an 8-day period. The average temperatures were fixed at 20 ° C. Daily variations of ± 5° C were imposed such that temperatures typically peaked around noon and lowest temperatures occurred around midnight.
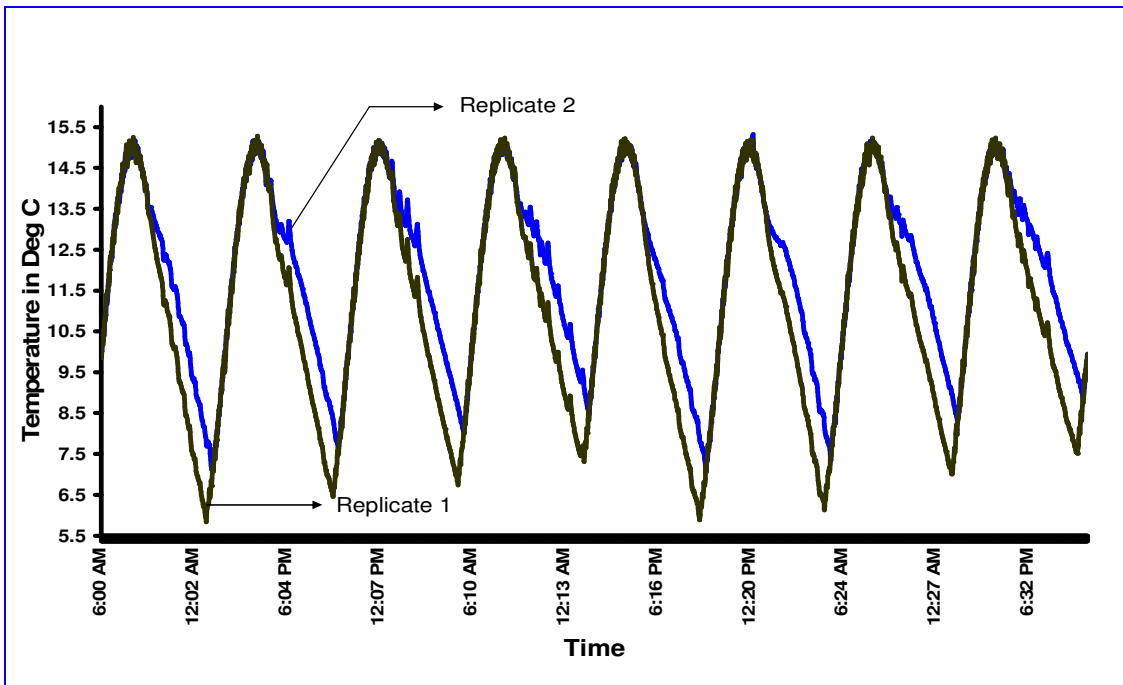
Figure 3.12 A variation of ± 5°C was imposed over an average temperature of 10 °C. The system had difficulty in cooling the water to 5 °C at midnight.
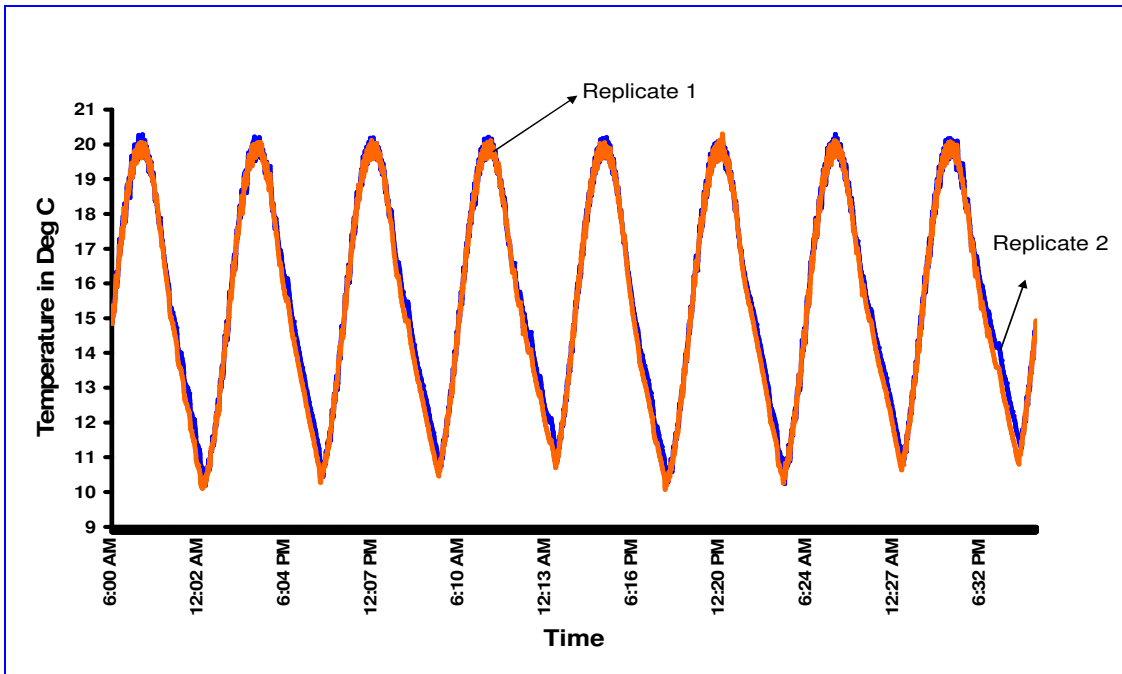
Figure 3.13 A variation of ±5°C was imposed over an average temperature of 15 °C. The system was able to maintain the expected temperatures within ±0.5°C.
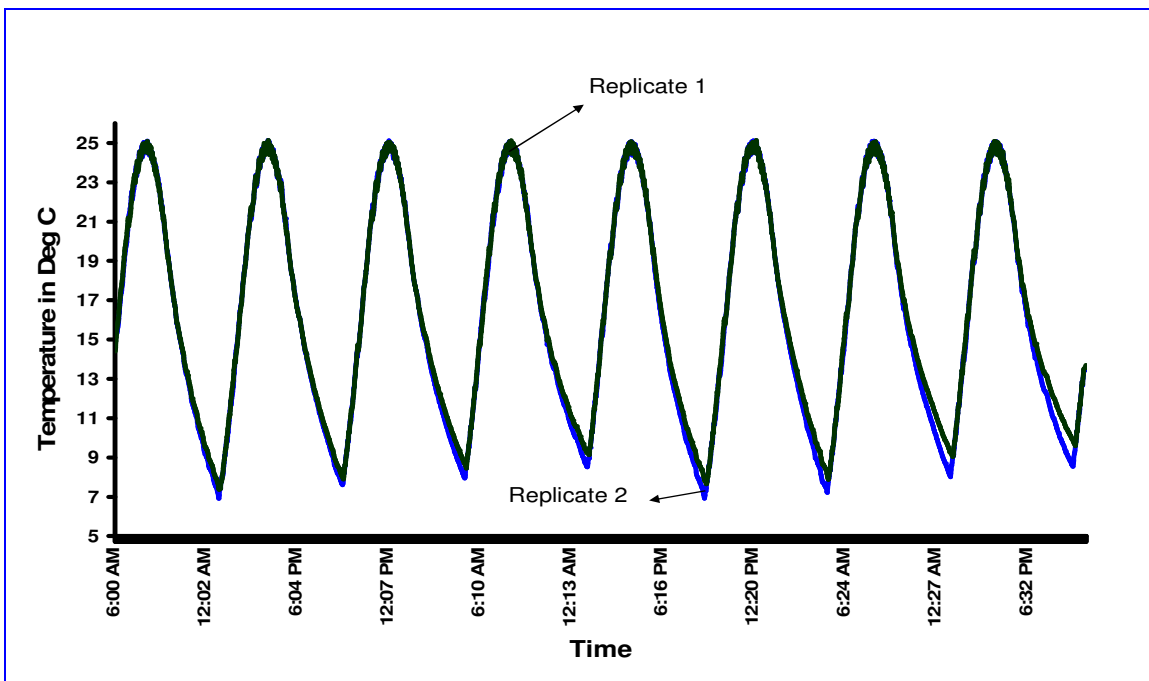
Figure 3.14 Diurnal simulations 15 ± 10°C were simulated for an 8-day period to test the physical limits of the system. Though the tanks reached highest temperatures during noon, they had difficulty in reaching the coolest point (5°C) at midnight

41

A detailed analysis of the temperature dynamics for a typical 24-hour period was carried out using Microsoft Excel. Figure 3.15 shows diurnal temperature simulation of 20 ± 1°C for a 24-hr period. The mean of differences between the expected and actual water temperatures was 0.135°C with a standard deviation 0.033°C, which means that on an average the actual temperatures are lower than expected temperatures. The system requires heat to maintain the required temperatures. Temperature variations for a 24-hr period, from the expected temperatures for each case are shown in figures 3.16 to 3.22. Table 3.2 shows the mean differences and variances for each case.
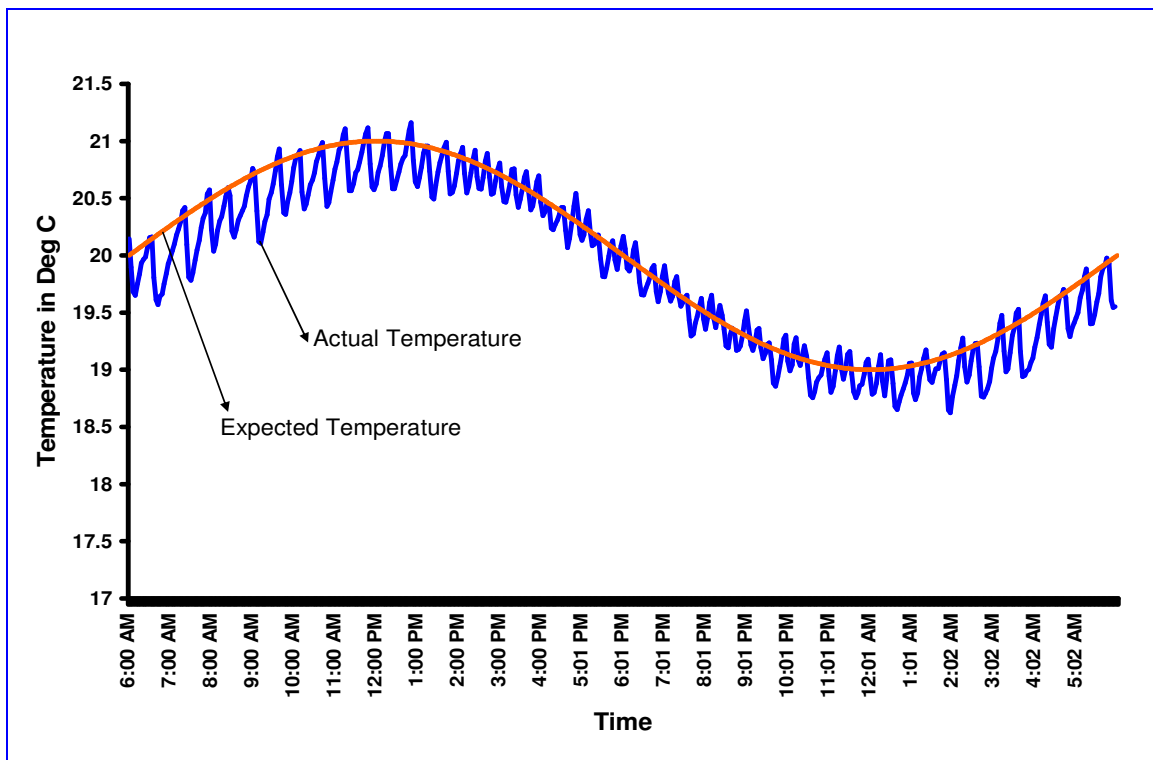
Figure 3.15 Temperatures of 20 ± 1°C was simulated in a tank for a 24-hr period to study temperature dynamics. For the given 24-hr period, the actual average temperature was below the expected temperature indicating the requirement of the heat into the system.
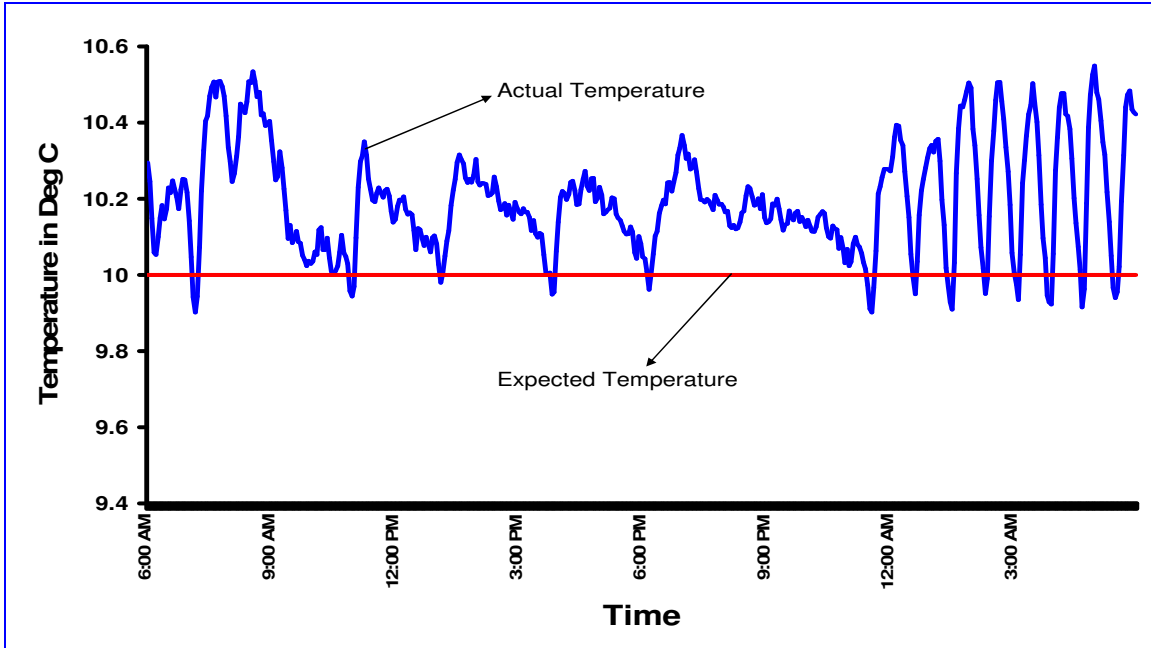
Figure 3.16 Temperatures of 10°C was simulated in a tank for a 24-hr period to study temperature dynamics. For the given 24-hr period, the actual average temperature was above the expected temperature indicating the requirement of the cooling into the system.
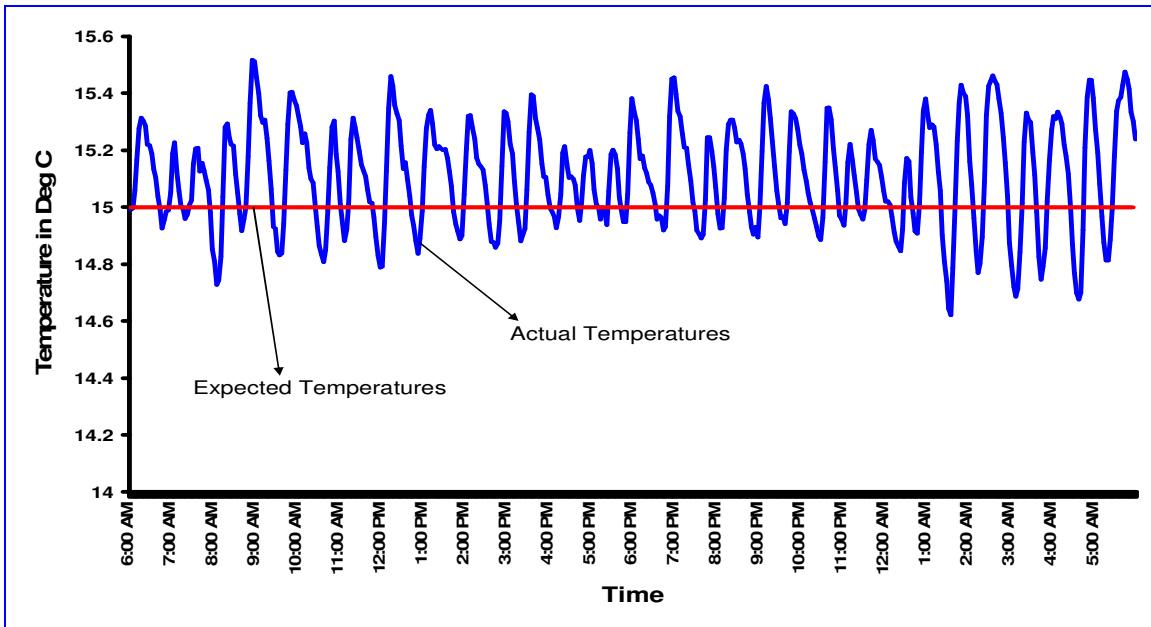
Figure 3.17 Temperature of 15°C was simulated in a tank for a 24-hr period to study temperature dynamics. For the given 24-hr period, the actual average temperature was above the expected temperature indicating the requirement of the cooling into the system.
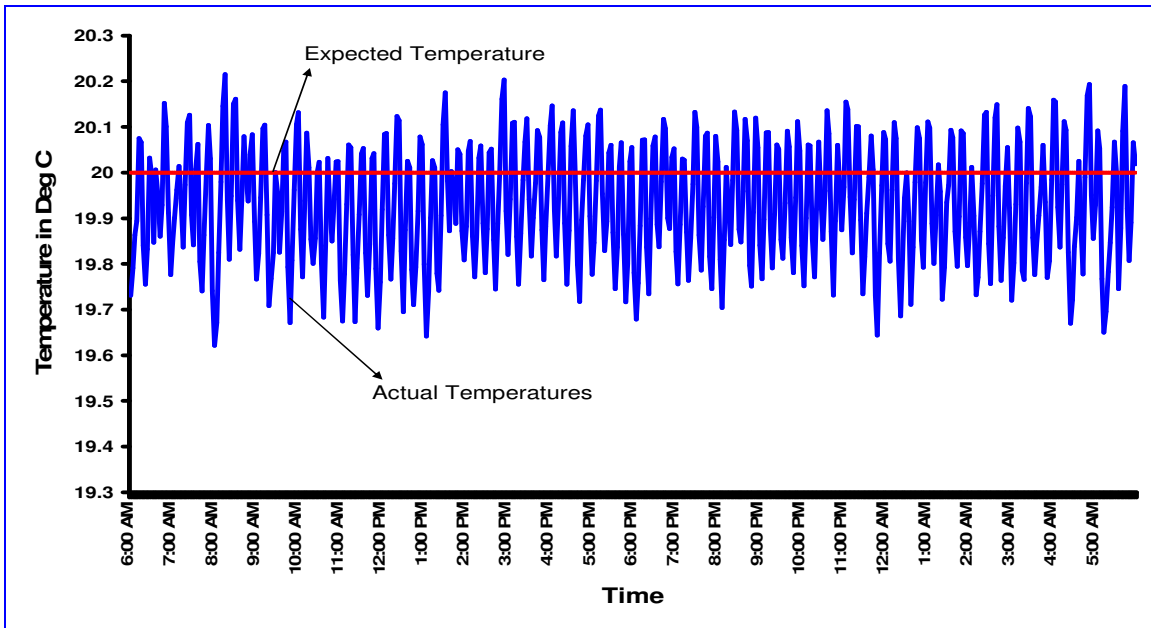
Figure 3.18 Temperatures of 20 °C was simulated in a tank for a 24-hr period to study temperature dynamics. For the given 24-hr period, the actual average temperature was below the expected temperature indicating the requirement of the heat into the system.
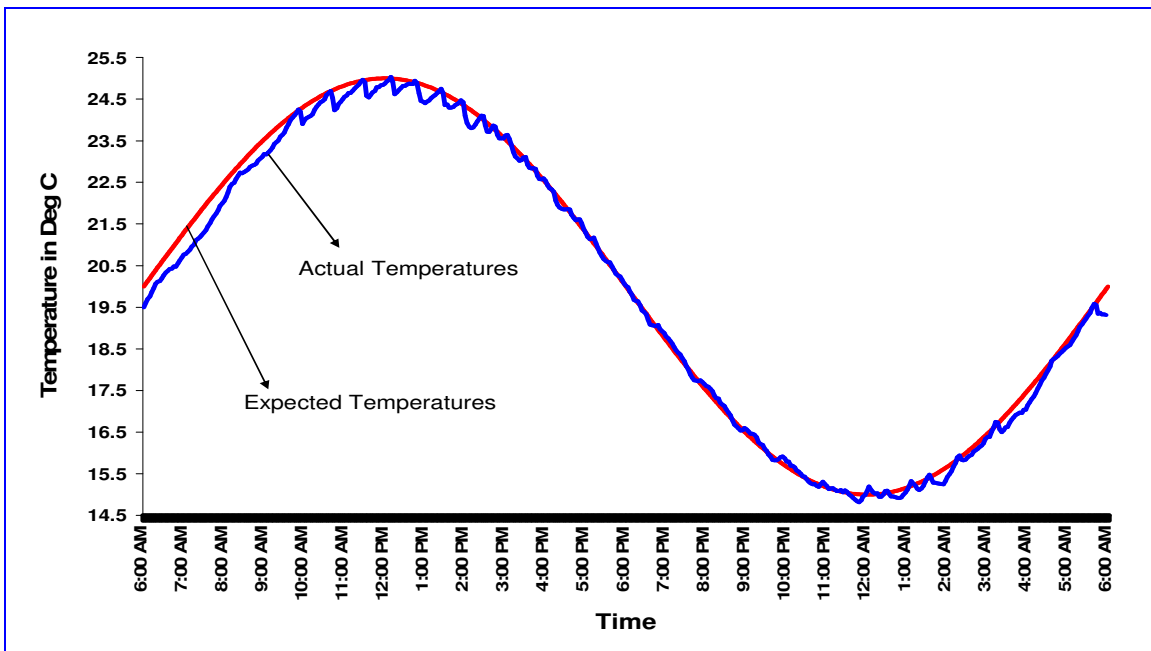
Figure 3.19 Temperatures of 20 ± 5°C was simulated in a tank for a 24-hr period to study temperature dynamics. For the given 24-hr period, the actual average temperature was below the expected temperature indicating the requirement of the heat into the system.
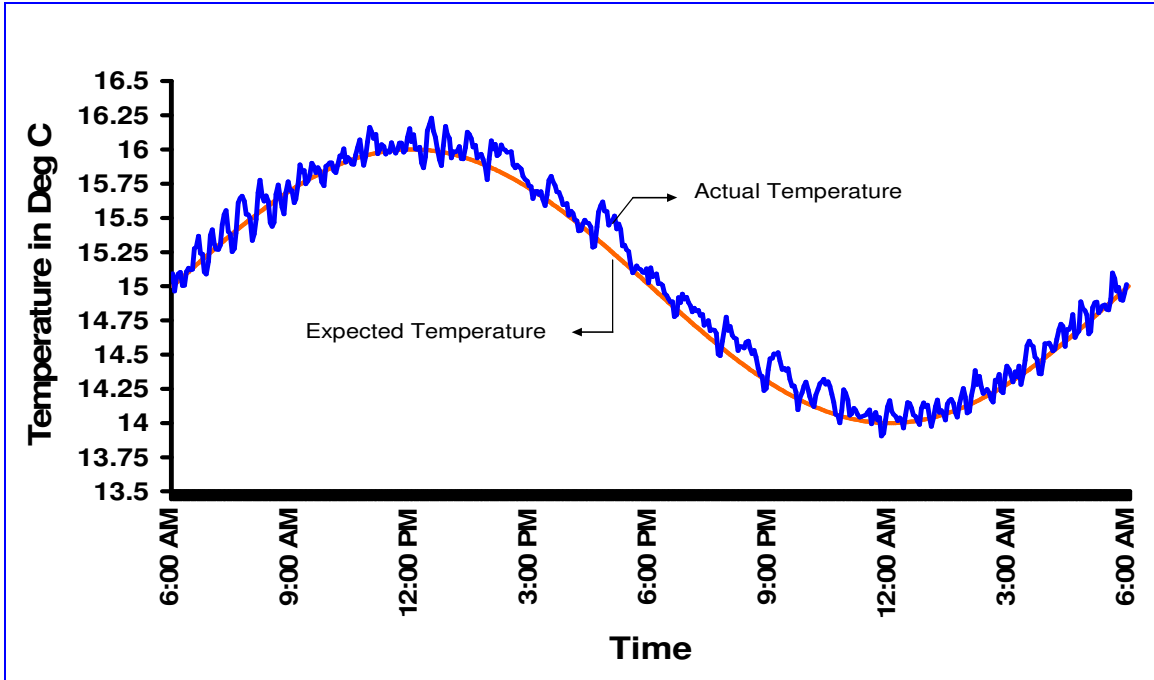
Figure 3.20 Temperatures of 15 ± 1°C was simulated in a tank for a 24-hr period to study temperature dynamics. For the given 24-hr period, the actual average temperature was above the expected temperature indicating the requirement of the cooling into the system.
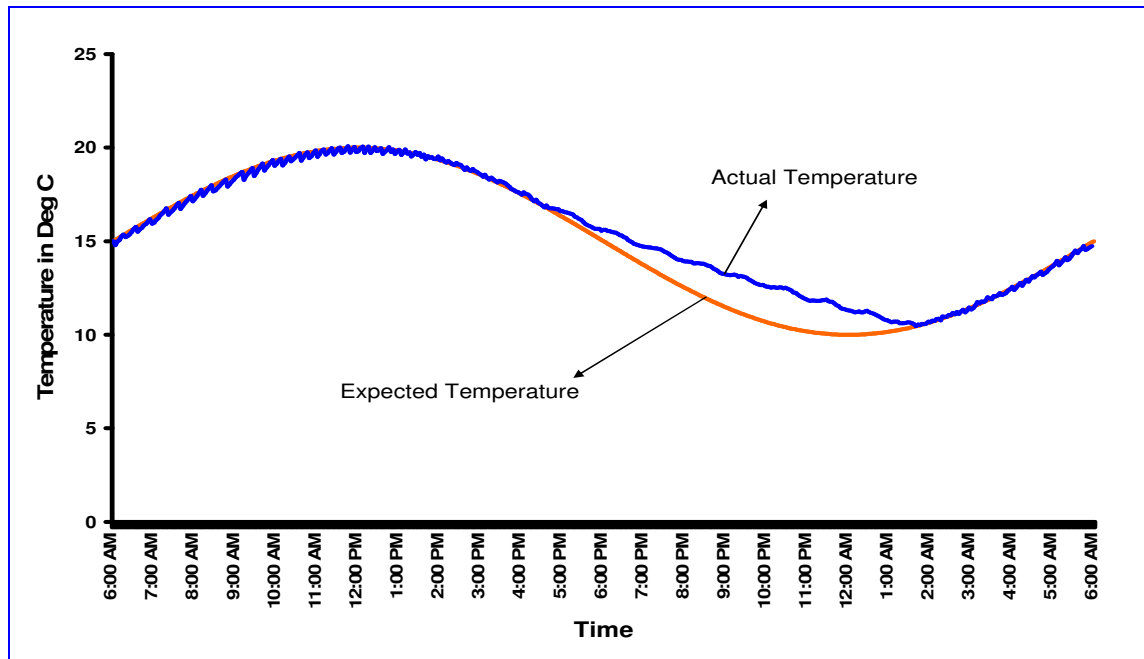
Figure 3.21 Temperatures of 15 ± 5°C was simulated in a tank for a 24-hr period to study temperature dynamics. For the given 24-hr period, the actual average temperature was above the expected temperature indicating the requirement of the cooling into the system

45
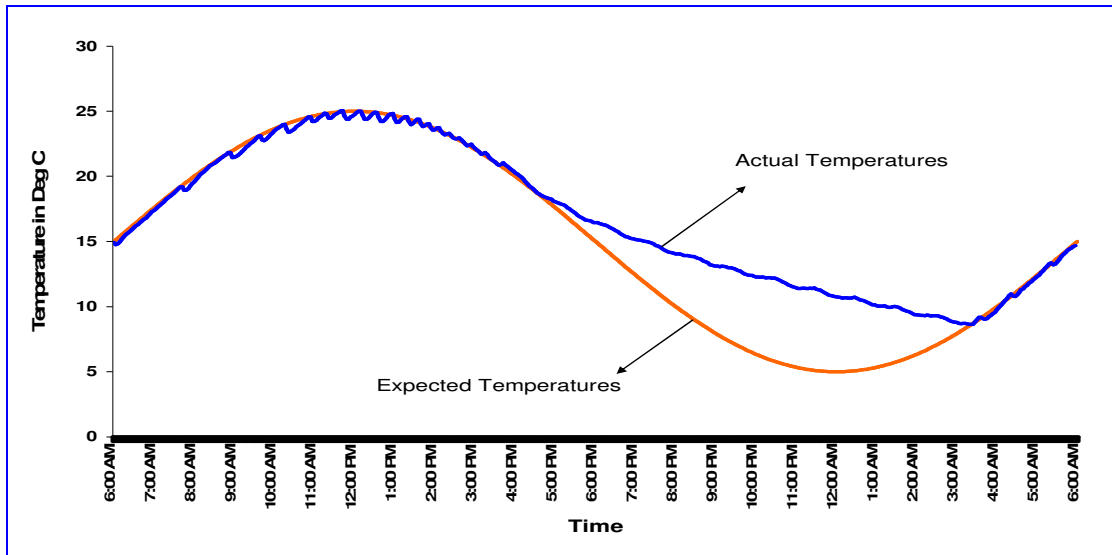
Figure 3.22 Temperatures of 15 ± 10°C was simulated in a tank for a 24-hr period to study temperature dynamics. For the give n 24-h period, the actual average temperature was above the expected temperature indicating the requirement of the cooling into the system.

Table 3.3 Means and Variances of the temperature differences between expected and actual temperatures for a 24-h period. A positive mean represents that actual temperatures are less than expected temperatures, requiring more heat into the system.

Description of the Temperature regime	Temperature (°C)	Mean of the Difference (°C)	Variance of the Difference (°C)	Remarks Required Energy
Constant	10	-0.19	0.02	Cooling
Constant	15	-0.10	0.03	Cooling
Constant	20	0.068	0.01	None
Diurnal	20 ± 1	0.13	0.03	Heating
Diurnal	20 ± 5	0.02	0.04	Heating
Diurnal	15 ± 1	-0.06	0.01	Cooling
Diurnal	15 ± 5	-0.42	0.53	Cooling
Diurnal	15 ± 10	-1.63	5.69	Cooling

3.5 Discussion

The effect of water temperature on biology of oyster was the basis for the development of this temperature control system. The design constraints listed in the previous sections were addressed by this system. The system and the software were easy to use and allowed for easy temperature manipulation. A copy of standard operating procedures (SOP) is included in Appendix C. With the present configuration, the system could be scaled up to 64 independent tanks. Because all tanks were independent, there was no mixing of water between the tanks which minimized potential for pathogen transfer. Also, each tank had built-in features for biological filtration, heating, and cooling. The filter media was effective in keeping the ammonia levels below 0.5 ppm despite each tank holding 30 oysters and the variable temperature fluctuations that were imposed. This was possible due to continuous aeration of water in the tank that ensures aerobic conditions in the filter media. The design of the cooling mechanism for this system was unique. The heat exchangers were custom-designed by affixing food grade stainless steel serving trays with covers. Despite continuous circulation of water through the heat exchangers, no corrosion and leaching was observed.

The design provided the control over the water temperature and the flexibility necessary for studying oysters in laboratory. Because the system could be used for a temperature range of 10-25°C, experiments on oysters could be planned all year round. The feedback received from the results on oyster responses may be utilized to modify the system dynamics. For example, the program execution time (presently set to 3 min intervals) may be decreased to 1 min interval to reduce sudden cooling or heating.

The system also allowed for programming variability with which different seasonal temperatures could be simulated in different tanks. This technology is useful to study the species that have specific temperature requirements.

Each of the tanks could be programmed to different temperature regimes. Because each tank was independent, different tanks can have different salinities. The system also offers the potential for real-time control of water temperature.

3.6 Concerns

Two of the four tanks (1 and 2) which were programmed to maintain 10°C, had difficulty in maintaining water temperatures at 10°C around 10/31/2004. The power bars were tripped due to electrical loads in excess of 60 amperes and temperatures increased to 10.8°C. This problem was solved by drawing power from two separate sources.

Though the system was efficient in holding and simulating diurnal water temperatures, it had difficulty in cooling the water below 7°C. As evident from the regime 10 ± 5°C and 15 ±10°C, the water did not reach 5°C at midnight as programmed. Because water in all tanks was constantly in circulation, the load on the chiller increased. Despite the chiller's peak performance, it reached a point beyond which further cooling was not possible. However, an additional chiller in each chill tank can facilitate faster cooling of water in chill tanks and to the desired temperature.

3.7 Conclusions

A process control temperature system consisting of 16 independent self-filtering tanks was developed at the LSUARS for assisting the researchers to study oysters in laboratory. The design of the system allowed for independent and variable control of temperature in

each tank. The system was able to hold and simulate diurnal temperatures between 10

and 25°C and allowed researchers to focus on oyster reproduction year round.

3.8 References

Buchanan, J.T., Roppolo, G.S., Supan, J.E., Tiersch, T.R., 1998. Conditioning of eastern oysters in a closed, recirculating system. Journal of Shellfish Research. 4, 1183-1189.

Fingerman. M., Fairbanks, L.D., 1957. Heat death and associated weight loss of the oyster (*Crassostrea virginica*). Tulane studies in Zoology and Botany. 5, 55-68.

Ford, S. E., Tripp, M. R., 1996. The eastern oyster *Crassostrea virginica*. Maryland Sea grant College, University of Maryland System, College Park. 581-660 pp.

Hofmann, E., Ford, S., Powell, E., Klinck, J., 2001. Modeling studies of the effect of climate variability on MSX disease in eastern oyster *Crassostrea virginica* populations. Hydrobiologia. 460, 195-212

Loosanoff, V.L., Davis, H.C., 1963. Rearing of bivalve mollusks. Advanced Marine Biology. 1, 1-136.

MacKenzie, C. L., 1996. The eastern oyster *Crassostrea virginica*. Maryland Sea grant College, University of Maryland System, College Park. 707-721 pp.

MacMillan, R. J., Cawthorn, R.J., Whyte, S. K., Lyon, P. R., 1994. Design and maintenance of a closed artificial seawater system for long-term holding of bivalve shellfish. Aquacultural Engineering. 13. 241-250.

Newell, R. I. E., Langdon, C. J., 1996. The eastern oyster *Crassostrea virginica*. Maryland Sea grant College, University of Maryland System, College Park. 185-229 pp

Nelson, T.C., 1928. On the distribution of critical temperatures for spawning and for ciliary activity in bivalve mollusca. Science. 67, 220-221.

Shumway, S. E., 1996. The eastern oyster *Crassostrea virginica*. Maryland Sea grant College, University of Maryland System, College Park. 467-513 pp.

Thompson , R. J., Newell, R. I. E., Kennedy, V. S., Mann, R., 1996. reproductive processes and early development. In: The eastern oyster *Crassostrea virginica.* Maryland Sea grant College, University of Maryland System, College Park. 335-370 pp.

Chapter 4

Use of Temperature Control System for Studies on Eastern Oyster *Crassostrea virginica*

4.1 Introduction

This part of the project was a symbiotic interaction between biology and engineering. Research interest from a biological point of view was to evaluate the temperature effects on oyster somatic cell proliferation. However, from the engineering point of view, our interest was to evaluate the temperature control system for studies on oyster somatic cell proliferation rates. The feedback obtained from the responses of oysters to different temperatures and temperature regimes were used to modify and improve the design of the process control system.

The goal of this study was to evaluate the use of the process control temperature system for studies on cell proliferation rates in the eastern oyster *Crassostrea virginica.* Specific objectives were to use the temperature control system to: 1) maintain constant temperature regimes of 10, 15, and 20 °C to study oyster somatic cell proliferation rates, and 2) simulate variable-temperature regimes of 10 ± 1°C, 10 ± 5°C, 15 ± 1°C, 15 ± 5°C, 15 ± 10°C, 20 ± 1°C, and 20 ± 5°C for studies on oyster somatic cell proliferation rates.

4.2. Materials and Methods

4.2.1 Preparation of Tanks

A total of sixteen tanks were used for this study (Figure 4.1). Each tank contained 200-L artificial sea water (Kent sea salt) at a concentration of 20 ppt. Circular trays (60 cm dia) fabricated from plastic coated hardware cloth (LOWE'S) were used as holding surface for oysters. Each tray was reinforced with 1.25-cm PVC tubes to resist the load of 60 oysters per tank. The trays were inserted in the tanks (Figure 4.2) until the trays were tightly

secured in the tank (about 20 cm from the water surface). On each tray two square holes 5 cm x 5 cm were made at opposite points to accommodate two 300-W immersion heaters.

4.2.2 Experimental Plan

Two separate experiments were planned and conducted. The purpose of both experiments was to evaluate the suitability of temperature control system from an engineering and design perspective. The first experiment was carried out during August-October 2003 and second experiment during March-May 2004 (in progress as of May 2004).
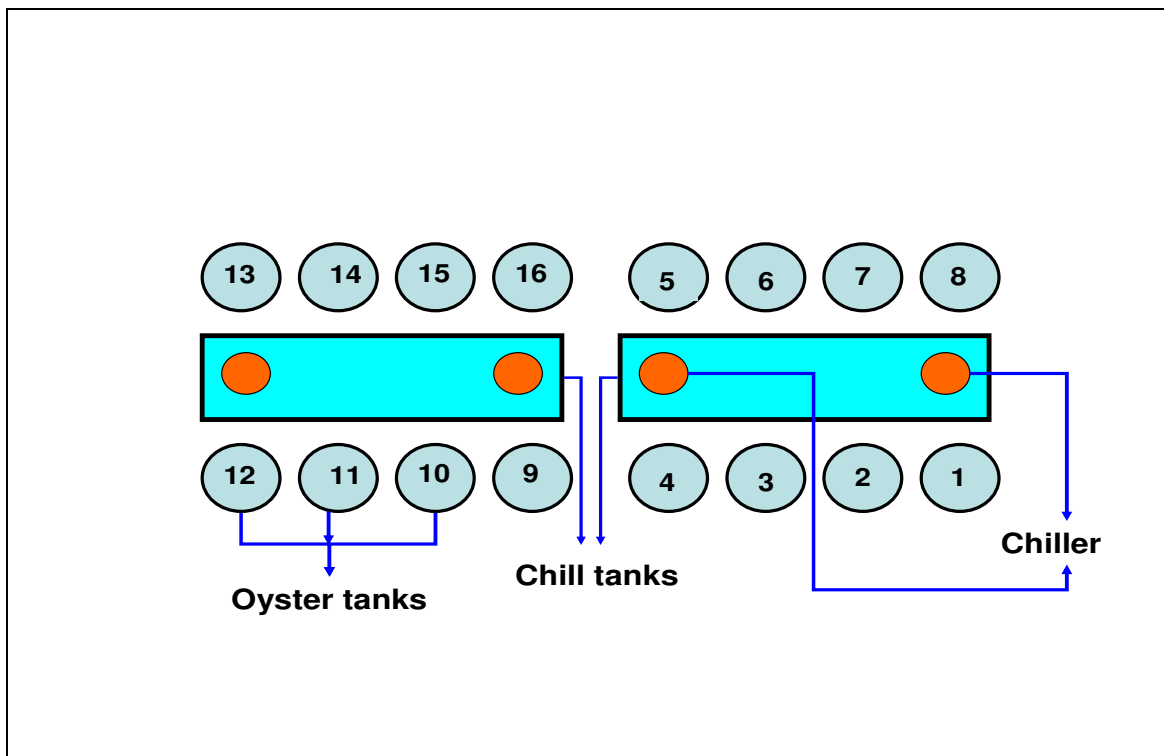
Figure 4.1 Layout of tanks at Aquacultural Engineering laboratory, at LSUARS. Tank 12 was used as a control (28°C). Eight oyster tanks were connected to stainless steel heat exchangers in one of the two chill tanks and water in each chill tank was cooled by a pair of 2000-W chillers.
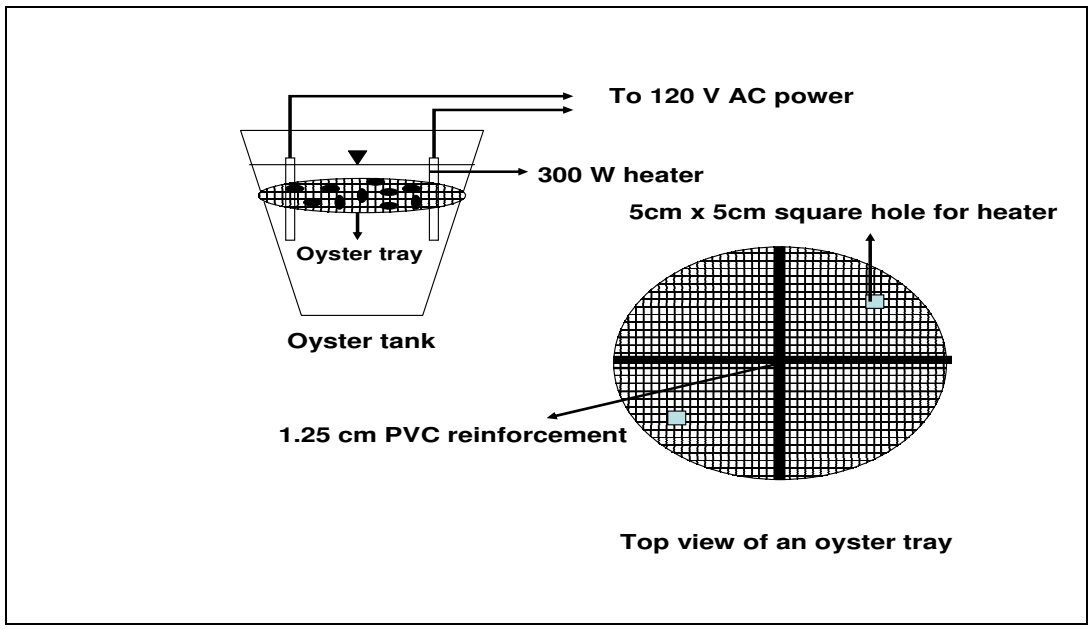
Figure 4.2 Schematic of oyster tray used in this study. The tray was reinforced with 1.25 cm PVC tube and secured with cable ties. The reinforced tray was able to support the load of 60 oysters. Two square holes (5 cm x 5 cm) were made to receive immersion heaters.

4.2.3 Experiment 1: Maintaining Constant Temperatures

Approximately 1-year-old oysters were procured from coastal waters of Grand Isle, LA in August 2003. The animals were transported to the Aquacultural Engineering Laboratory at the Louisiana State University Aquaculture Research Station (LSUARS). The outer shells of oysters were washed to clean the dirt and transferred on to the trays in the tanks at the rate of 60 oysters per tank. After 72 h, the animals were fed 15 mL of Instant Algae Shellfish Diet (Reed Mariculture Inc., Campbell, CA) twice daily.

Three temperature regimes, 10, 15, and 20°C, were selected to hold the oysters. Because the water temperature in the laboratory and Grand Isle waters were both around 28°C (when the animals were procured), 28°C was set as the control temperature for this experiment and initially all 15 tanks were set to 28°C for a 7-d acclimation period. The 10°C regime was replicated in four tanks (Tanks 5, 14, 15, and 16), 15°C in six tanks

(Tanks 2, 3, 4, 9, 10, and 11), and 20°C in four tanks (Tanks 1, 6, 7, and 8). The control temperature 28°C was simulated in tank 12. Then, water temperatures in all tanks except for tank 12, were dropped at the rate of about 1°C per day till all tanks reached 20°C. After reaching 20°C, water temperatures in tanks 1, 6, 7 and 8 were held constant at 20°C, and the water temperatures in the rest of the tanks were continuously dropped at the rate of about 1°C per day. Again, after reaching 15°C, water temperatures in tanks 2, 3, 4, 9, 10, and 11 were held constant at 15°C, and temperatures in remaining tanks were allowed to drop until tanks 5, 14, 15, and 16 reached 10°C. During this time, four tanks were at 20°C, six tanks were at 15°C, four tanks were at 10°C (Figure 4.3), and the control tank at 28°C. The system was allowed to maintain these temperatures in tanks for another approximately 10-day period before obtaining samples of oysters for cell proliferation research. The interest from an engineering perspective was in the optimum functioning of system under temperature regimes (10, 15, and 20 °C), while the biological interest was the proliferation of oyster hemocytes, heart, and labial palps cells under these temperatures regimes.
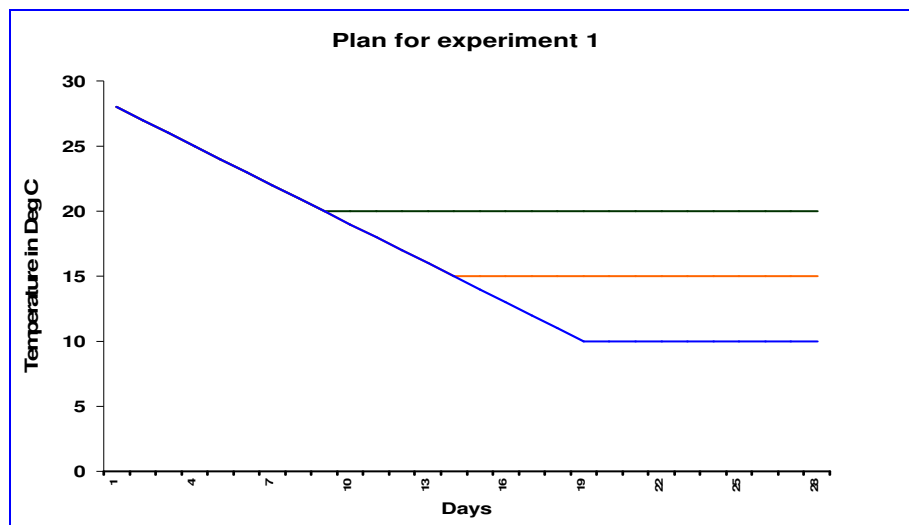
Figure 4.3 Water temperatures were dropped @1°C/day till tanks reached their desired temperatures (10, 15 and 20°C)

Because the computer measured the temperatures every 3 min, a 24-hr period consisted of 480 temperature measurements per tank. With sixteen tanks under observation, data became enormous to handle. Hence for simplicity, mean temperatures over a 24-hr period were computed so that each tank had just one temperature observation per day.

The data were analyzed using Statistical Analysis System (SAS Inc., Version 9.0, NC) and Microsoft Excel (Microsoft Inc., WA). One-way analysis of variance (1-way ANOVA) was used to test the differences between tank replications of same temperature regimes. Further, a grand average (average of all tanks for each regime) was computed. One-sample T-test was used to test the differences between grand averages and their expected temperatures.

4.2.4 Experiment 2: Imposing Temperature Variations

Procurement of animals was same as experiment 1. However, second experiment had 30 oysters per tank and acclimation period was increased to a two-week period (1-week period for experiment 1). The ambient water temperature in laboratory was between 20ºC and 21ºC. The water temperatures were dropped at the rate of 1ºC/day till tanks 1, 6, 7 and 8 were at 20ºC, tanks 2, 3, 4, 9, 10, and 11 were at 15ºC, and tanks 5, 12, 14, 15, and 16 were at 10ºC (as explained in experiment 1). After the water temperatures reached the desired temperatures (10, 15 or 20ºC as the case may be), the tanks were allowed to maintain the temperatures for approximately one week. After two week, variability was imposed on each of the temperature regimes. Table 4.1 describes the imposed variability in tanks.

Table 4.1 A variety of fluctuations were imposed on tanks. Each regime was replicated twice for statistical soundness.

Tanks	Average Temperature (°C)	Variability (°C)
1 and 7	20	±1
6 and 8	20	±5
9 and 11	15	±1
3 and 10	15	±5
2 and 4	15	±10

4.3 Results

4.3.1 Experiment 1: Maintaining Constant Temperatures

1. **Regime 1 (20ºC):** After dropping the water temperature to 20ºC in tanks (explained in methods section), Tanks 1, 6, 7, and 8 were programmed to hold water at 20ºC. The average temperatures in tanks were within ± 0.2ºC (Figure 4.4). No apparent differences between the temperatures in Tanks 1, 6, 7, and 8 were found ($P=0.27$). Throughout the experimental period, all tanks (1, 6, 7, and 8) were found to maintain the water temperatures below 20°C ($P=0.0012$), indicating an overall heat deficit in the system. An extra heater may be needed to compensate the heat deficit in the system. However, water temperatures in tanks 7 and 8 increased to 20.2ºC around 09/07 due to electrical disturbances (Figure 4.5). These concerns are discussed separately in later sections.
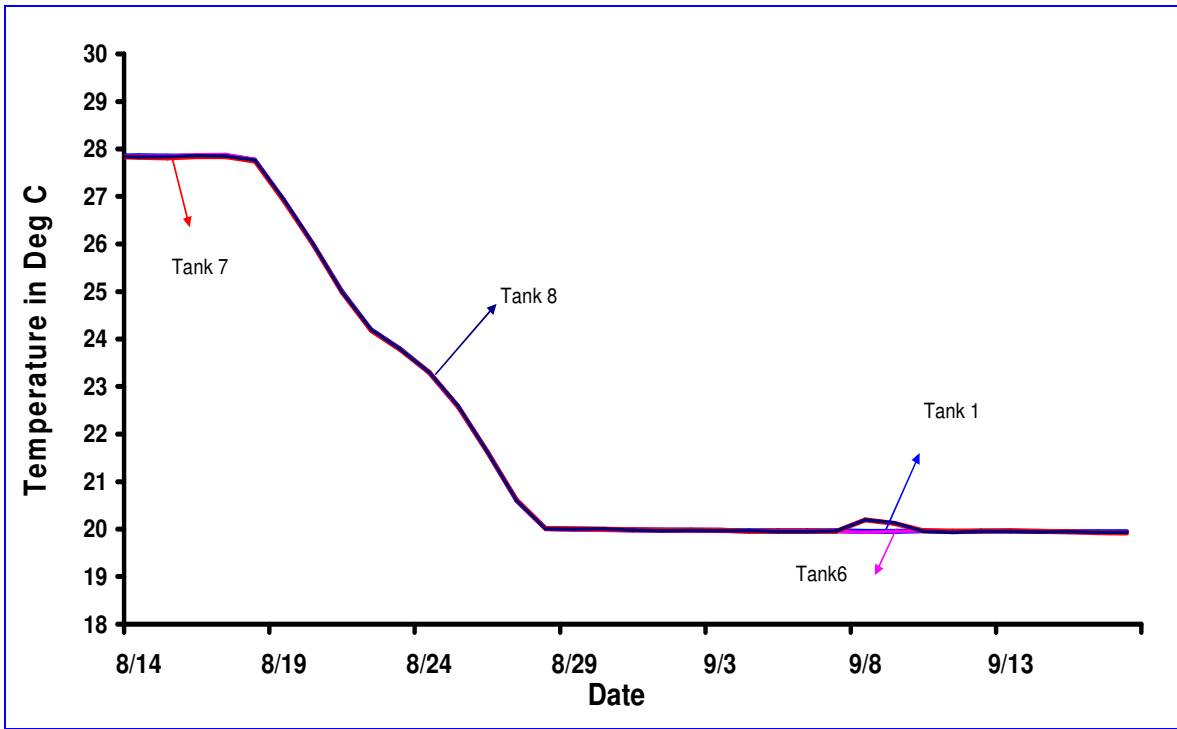
Figure 4.4 Temperatures in tanks 1, 6, 7 and 8 were dropped from 28°C to 20 °C and were held at 20 °C for a 2-week period for studies on oyster somatic cell proliferation.
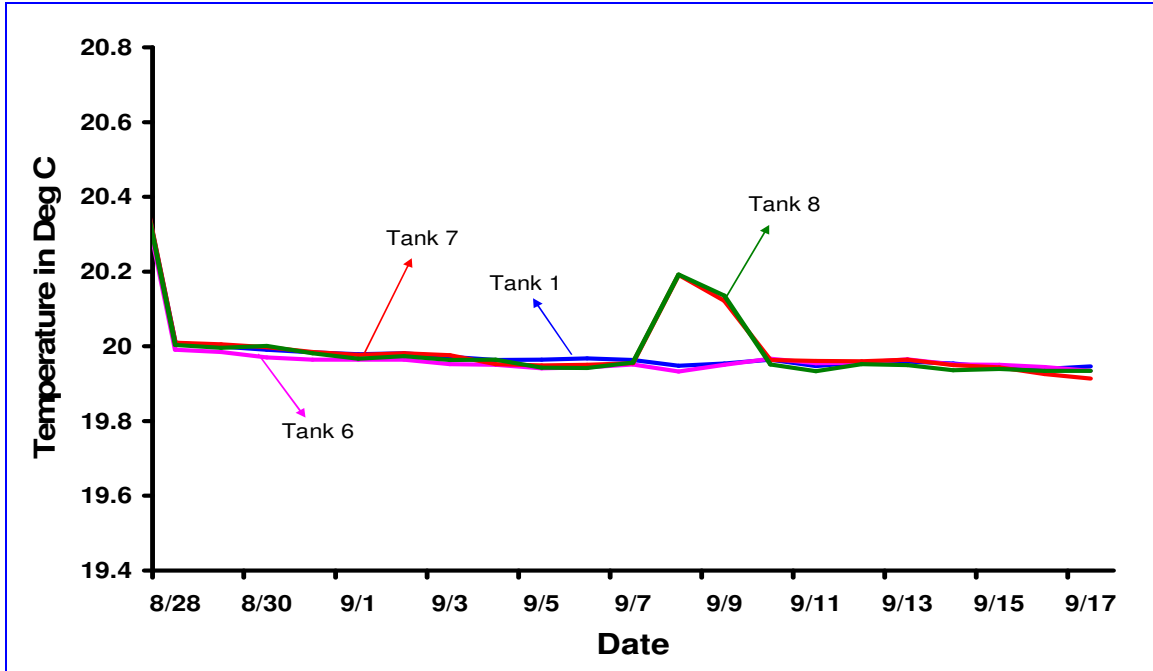
Figure 4.5 Water temperature in tanks 7 and 8 was increased to 20.2ºC due to an electrical disturbance.

2. **Regime 2 (15ºC):** Tanks, 2, 3, 4, 9, 10, 11 were allowed to reach 15ºC and programmed to maintain at 15ºC. Around 8/21, electrical outlet connected to tank 11, was tripped and water temperature increased to 29ºC (Figure 4.6). Corrective measures were immediately initiated by replacing power bars. Again, similar problems were encountered for tanks 9, 10, and 11 between 9/4 and 9/6 (Figure 4.7). However, after 9/7, the system did not experience electrical problems for 15ºC regime and all six tanks were at 15ºC ($\pm$ 0.2ºC). No apparent differences between the temperatures in tanks 2, 3, 4, 9, 10, and 11 were found (P=0.20). Throughout the experimental period, all tanks (2, 3, 4, 9, 10, and 11) were found to maintain the water temperatures above 15°C (P= 0.01), indicating an overall cooling deficit in the system. An extra chiller may be needed to compensate the heat deficit in the system.
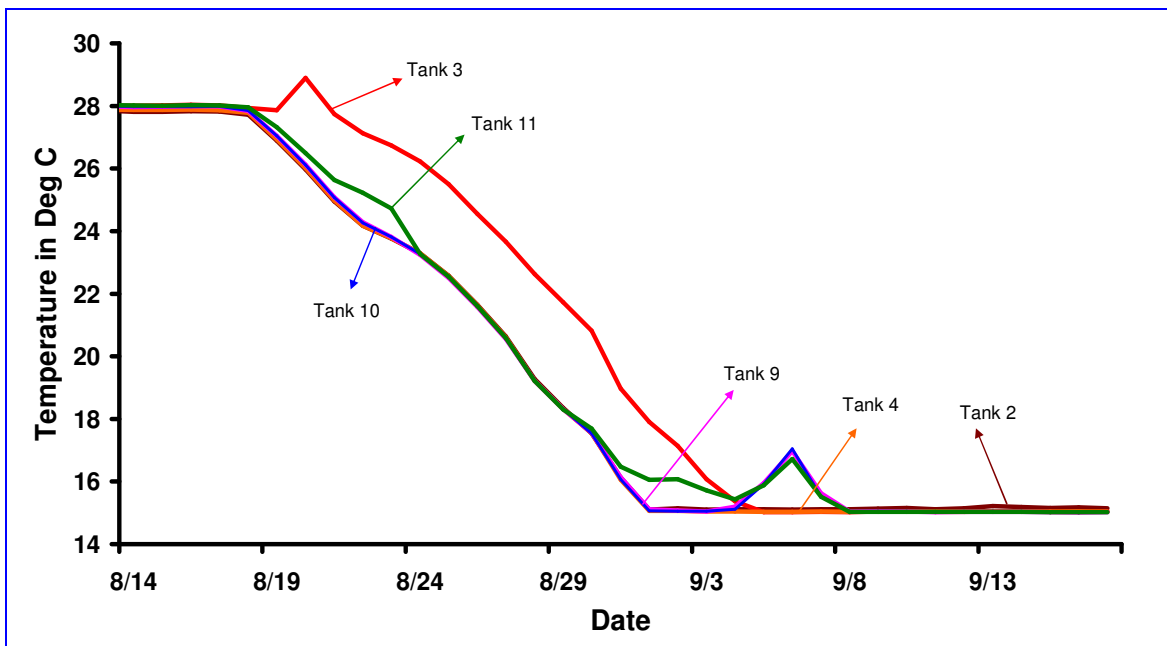
Figure 4.6 Temperatures in tanks 2, 3, 4, 9, 10, and 11 were dropped from 28°C to 15 °C and were held at 15 °C for a 2-week period for studies on oyster somatic cell proliferation. The sudden increase in temperature tank 11 around 8/21 was due to electrical problem and so was for tanks 9, 10 and 11 around 9/4.
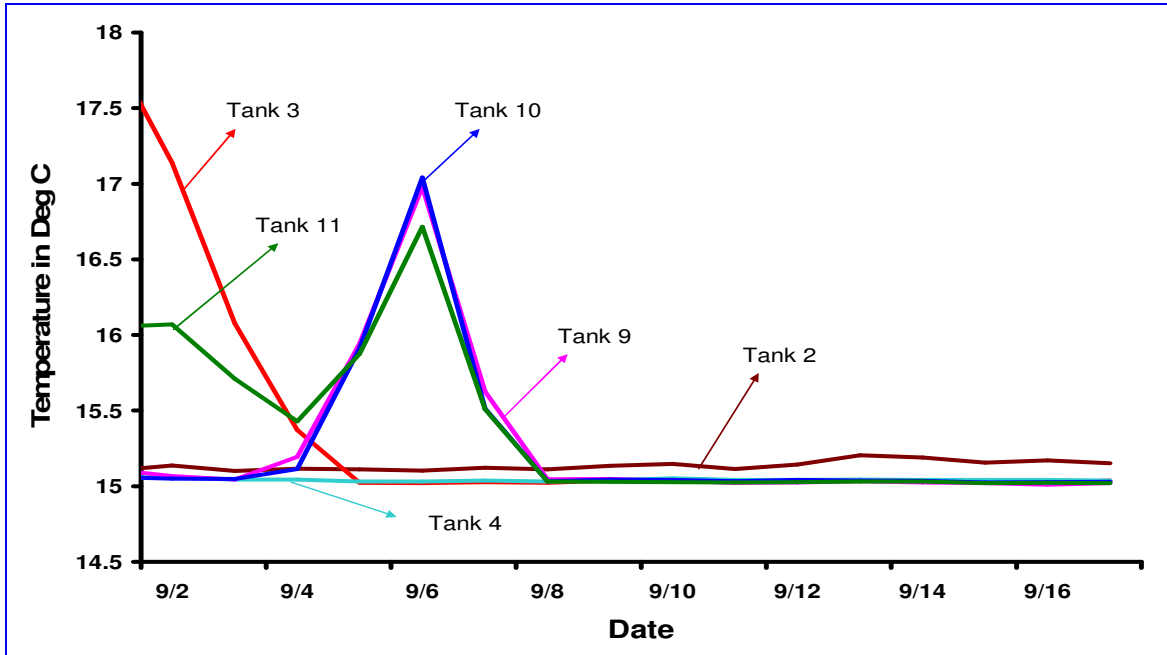
Figure 4.7 Tanks 9, 10, and 11, were connected to the same power bar experienced electrical problems. The power bar was replaced to solve the problem.

3. **Regime 3 (10 ºC):** Although, four replicates (tanks) were used to maintain 10ºC, data from three tanks (5, 15, and 16) only were used in data analysis (Figure 4.8). The temperatures in tank 16 were not available between dates 9/11to 9/13. Tank 15 experienced a steep increase in water temperature indicating that control of the system over tank 15 was temporarily lost for the period 8/23-8/24. However, proper electrical connections resolved the problem. Tank 14 also experienced similar electrical problems around 9/3. But, after resolving the electrical and mechanical issues all three tanks were around 10º ± 0.25ºC (Figure 4.9). Differences between the temperatures in tanks 5, 14 and 15 were found ($P=0.02$). However, throughout the experimental period, all tanks (5, 15, and 15) were found to maintain the water temperatures above 10°C ($P=0.015$), indicating an overall cooling deficit in the system. An extra chiller may be needed to compensate the heat deficit in the system.
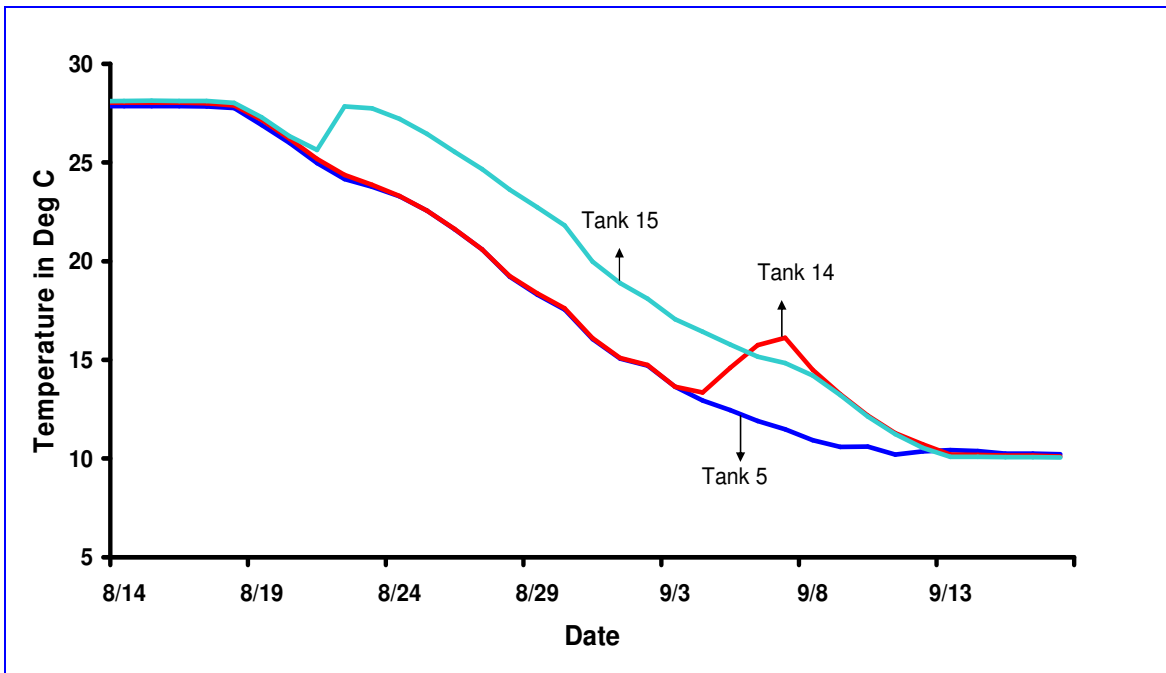
Figure 4.8 Temperatures in tanks 5, 14, and 15 were dropped from 28°C to 10 °C and were held at 10 °C for a 2-week period for studies on oyster somatic cell proliferation. Tank 14 experienced an electrical outage that increased the water temperatures by over 2°C.
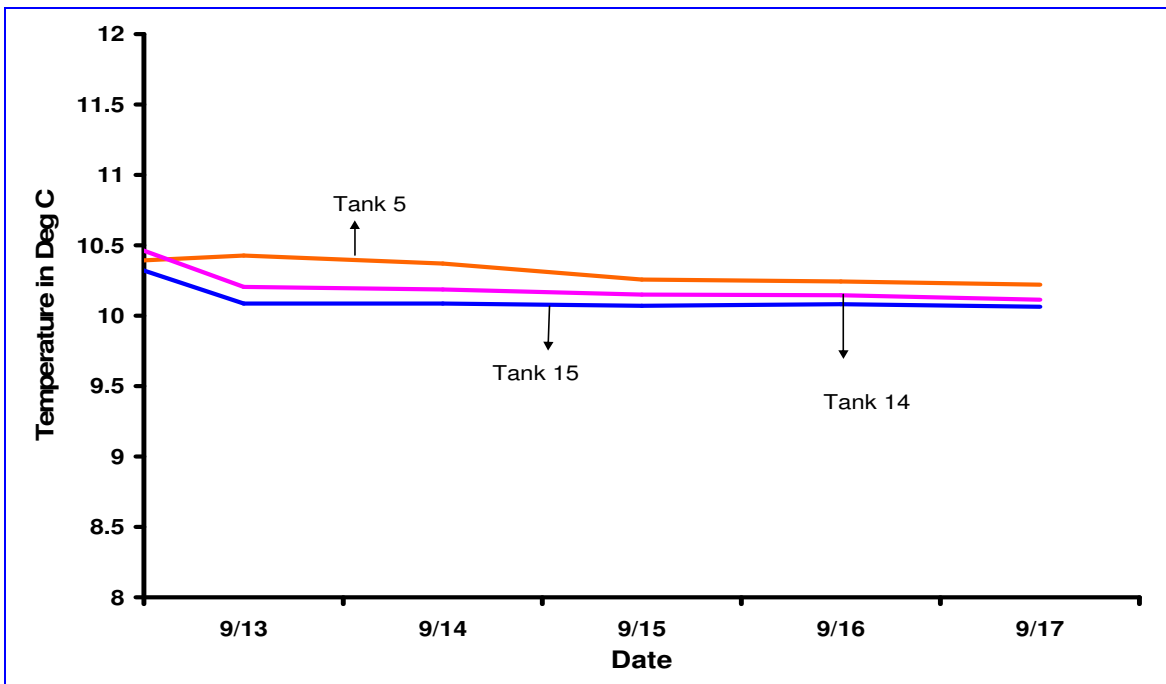
Figure 4.9 Apart from minor electrical and mechanical problems, the system was able to maintain the water temperature in tanks 5, 14, and 15 around 10 °C.

4.3.2 Experiment 2: Imposing Temperature Variations

1. **Regime 1 (20 ± 1ºC):** Water temperatures in tanks 1 and 7 were found to be within ± 0.5ºC from the expected temperatures. Because the water was held at 20ºC for 20 days before imposing simulations, plotting of data became difficult (approximately 19000 observations). Hence only part of the data was plotted (Figure 4.10). For an easy understanding, a 24-hour temperature data was plotted (Figure 4.11). As evident from the graph, the system was controlling water temperatures as programmed. The mean of the difference between expected temperatures and actual temperatures was 0.11ºC and variance of the difference was 0.015ºC.
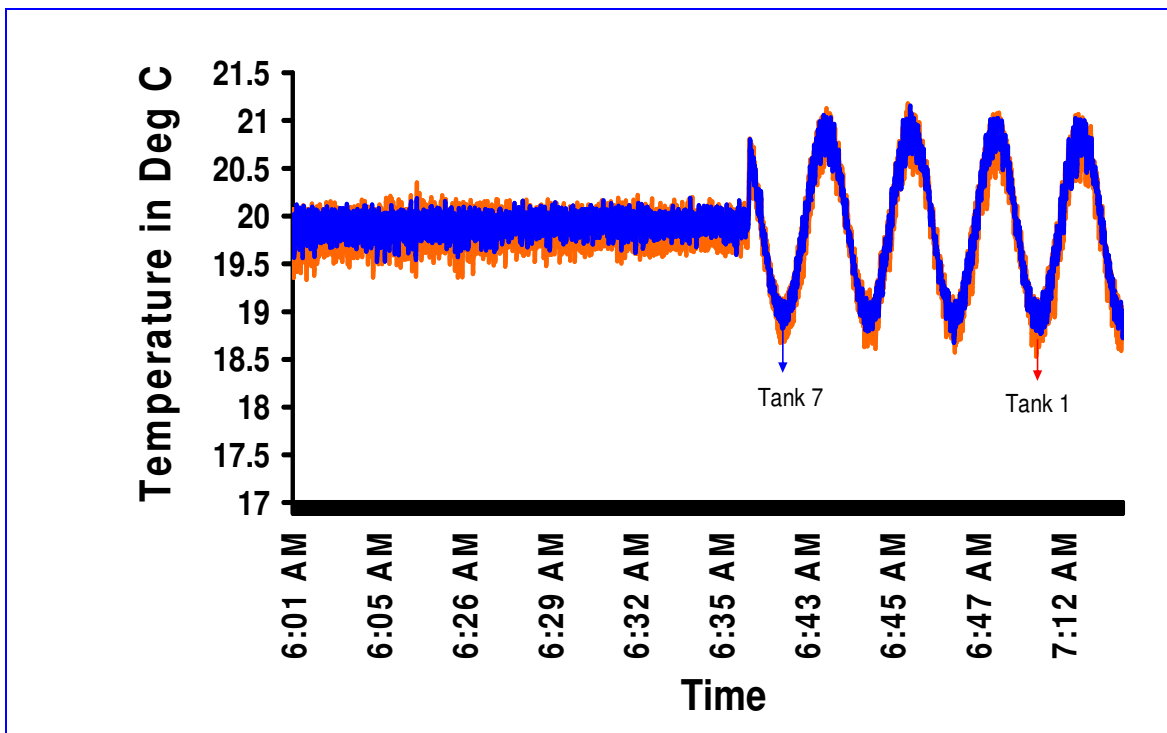
Figure 4.10 Water temperatures in tanks 1 and 7 for a 11-day-day period. Initial temperatures were maintained around 20 °C for 20 days before imposing a variation of ± 1°C.
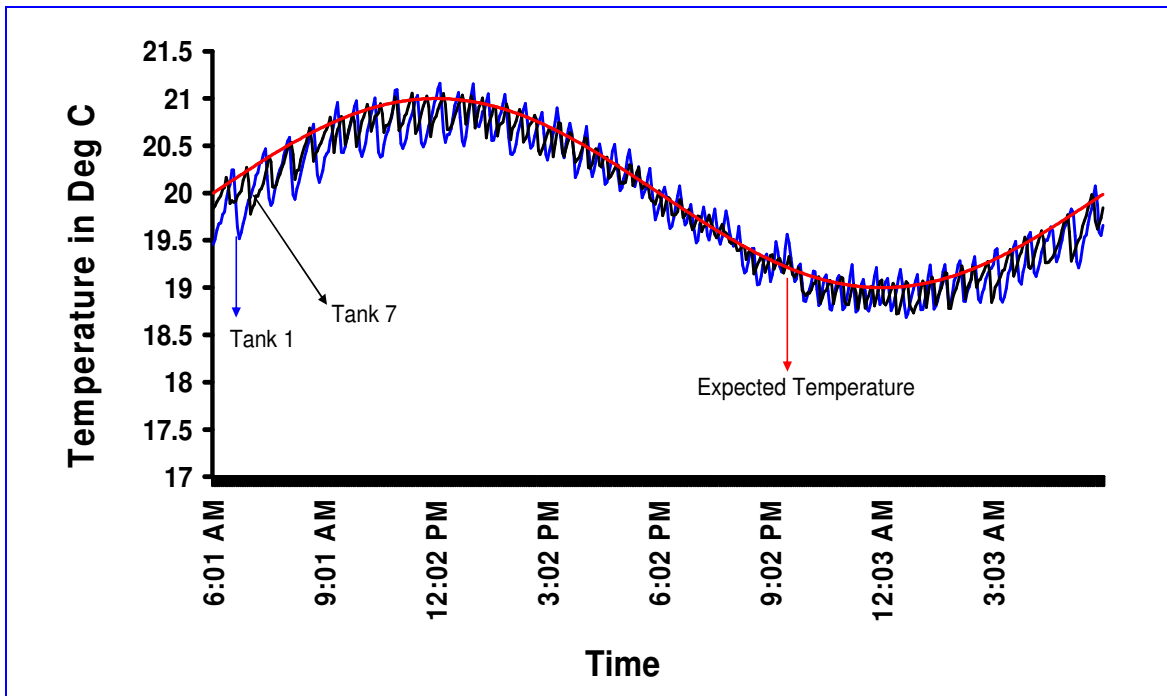
Figure 4.11 Mean differences between expected and actual water temperatures in tanks 1 and 7 were 0.11°C and variance of differences was 0.015°C.

2. **Regime 2 (20 ± 5ºC):** The system was able to maintain temperatures at the level of 20°C and simulate variations of ± 5°C around 20 °C (Figure 4.12). The mean and variance of the 24-hr difference between expected and actual water temperatures were 0.19°C and 0.008°C (Figure 4.13). Means and variances of the differences between expected and actual temperatures for all regimes for a 24-hr period are provided (Table 4.2).

3. **Regime 3 (15 ± 1°C):** The system had no difficulty to simulate this regime. However in tank nine, the temperatures did not drop to 14°C at midnight as programmed. During the heating phase, water temperatures in tanks 9 and 11 reached around 16°C.
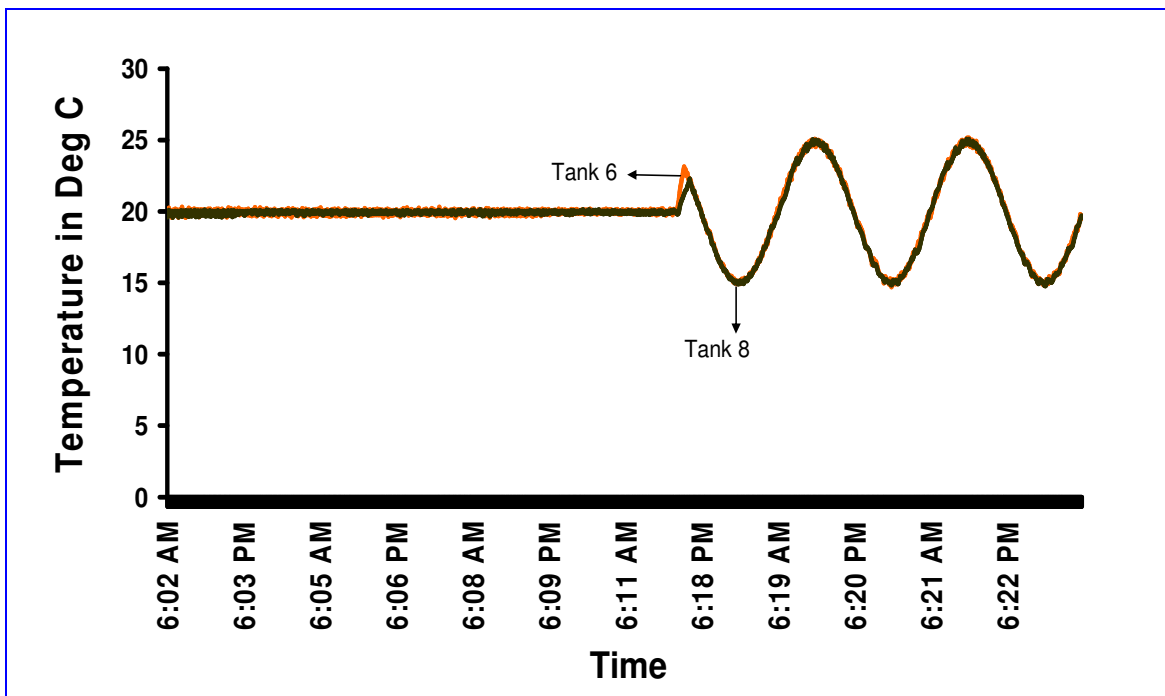
Figure 4.12 On an average, the system was able to simulate water temperatures of 20 ± 5°C within 0.2°C throughout experimental period.

Table 4.2 Mean of differences between expected and actual temperatures for a 24-hr period. Positive mean denotes that heat if required into the system to maintain water at expected temperatures. Regimes 15 ± 5°C and 15 ± 10°C needed to be cooled further to be maintained at expected temperatures

Regime	Mean of difference between expected and actual water temperatures (°C)	Variance of differences between expected and actual water temperatures (°C)	Remarks
20 ± 1°C	0.11	0.015	Heat required
20 ± 5°C	0.19	0.088	Heat required
15 ± 1°C	-0.03	0.004	Cooling required
15 ± 5°C	-0.43	0.556	Cooling required
15 ± 10°C	-1.78	6.61	Cooling required
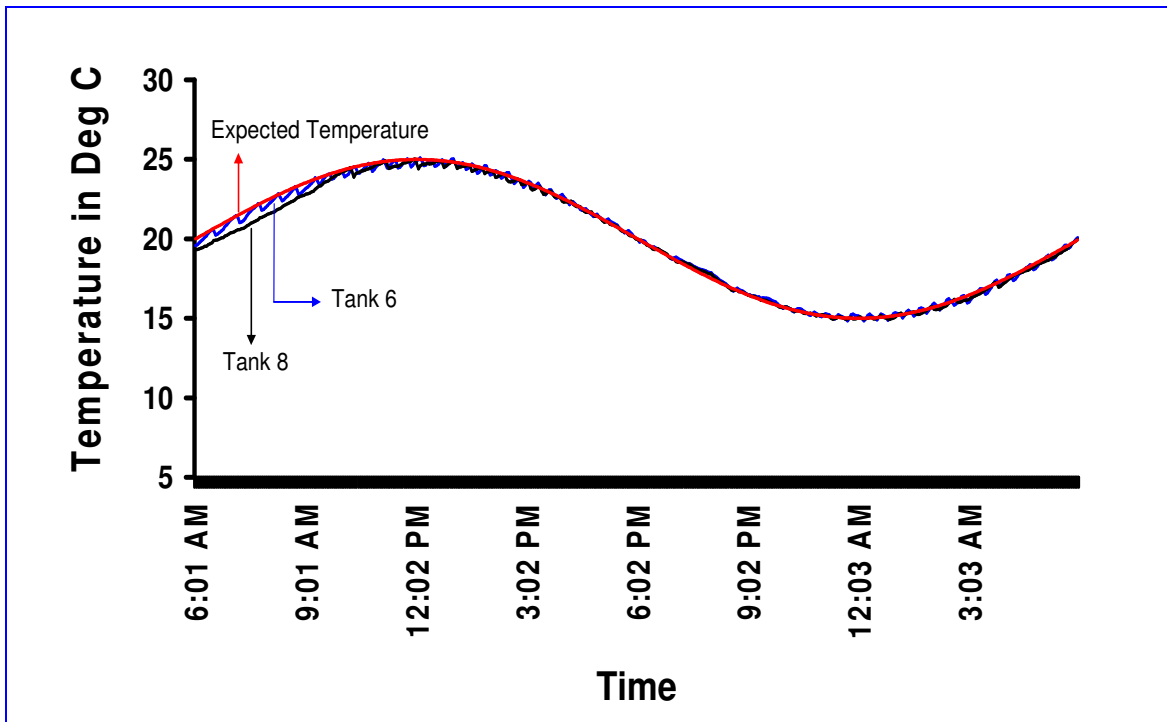
Figure 4.13 Mean differences between expected and actual water temperatures in tanks 1 and 7 were 0.19°C and variance of differences was 0.088°C.

4. **Regime 4 (15 ± 5°C):** Of two tanks (tanks 3 and 10) were imposed with this variation; both tanks had difficulty in cooling. However system was able to provide heat, and temperatures were near their expected temperatures.

5. **Regime 5 (15 ± 10°C):** Neither of tanks (2 and 4) was able to reach 5°C at midnight. Also, heating patterns in tanks 2 and 4 were different. Although tank 2 was able to reach 25°C by noon, tank 4 had difficulty in heating.
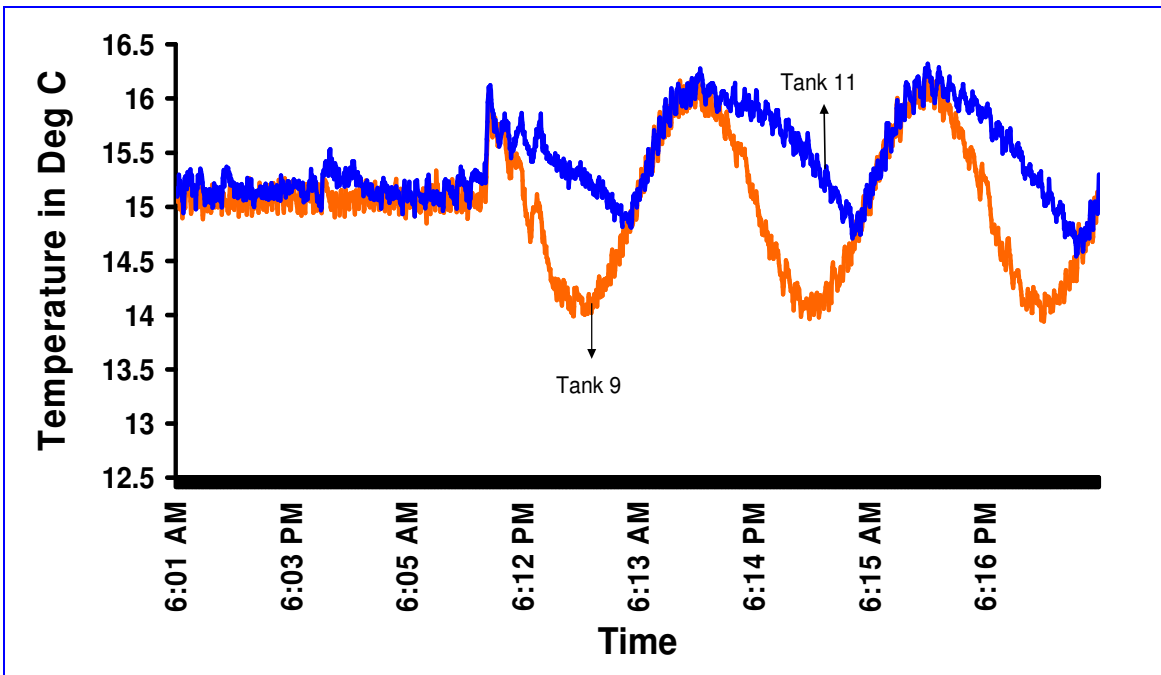
Figure 4.14 After dropping water temperatures from around 20°C to 15°C, system maintained water at 15°C before simulations were imposed.
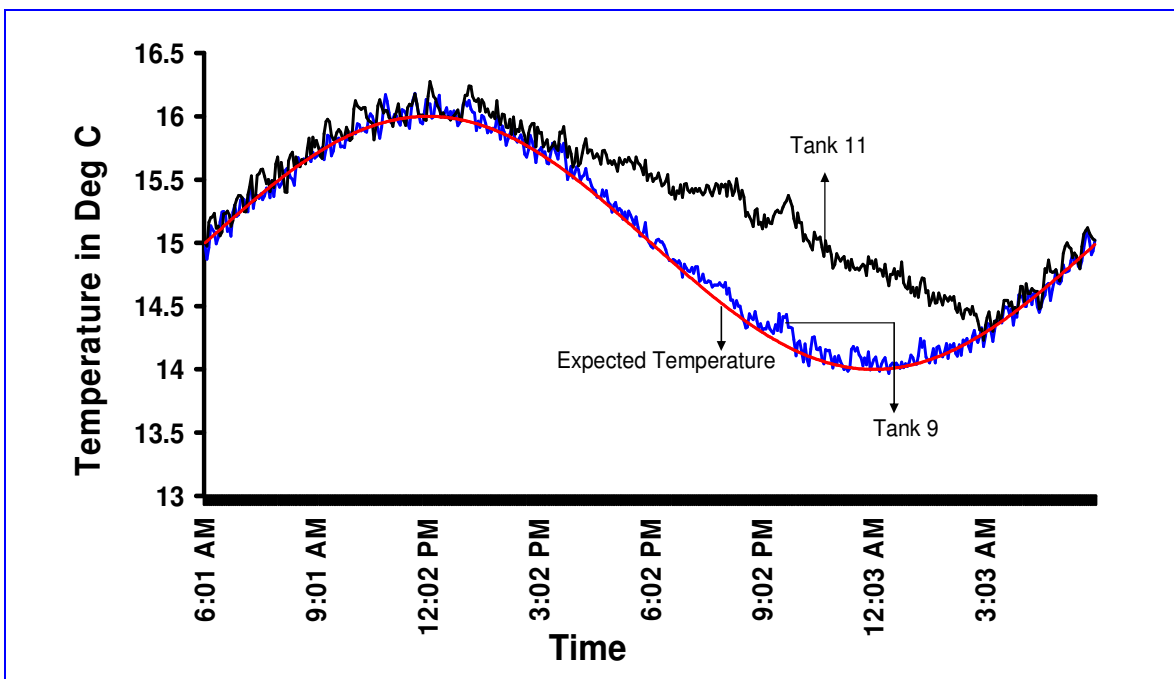
Figure 4.15 The system was able to simulate water temperatures within 0.2°C. Water in tank 11 had problems with cooling.
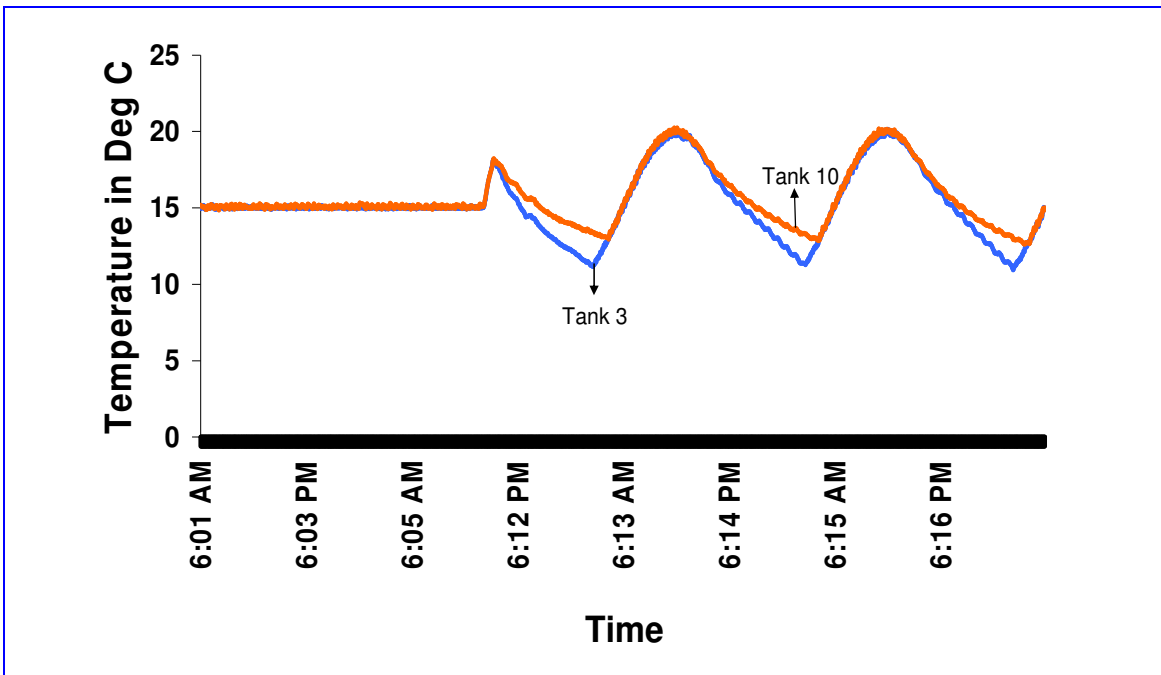
Figure 4.16 The system was able to heat the water efficiently, but could not cool the water as desired.
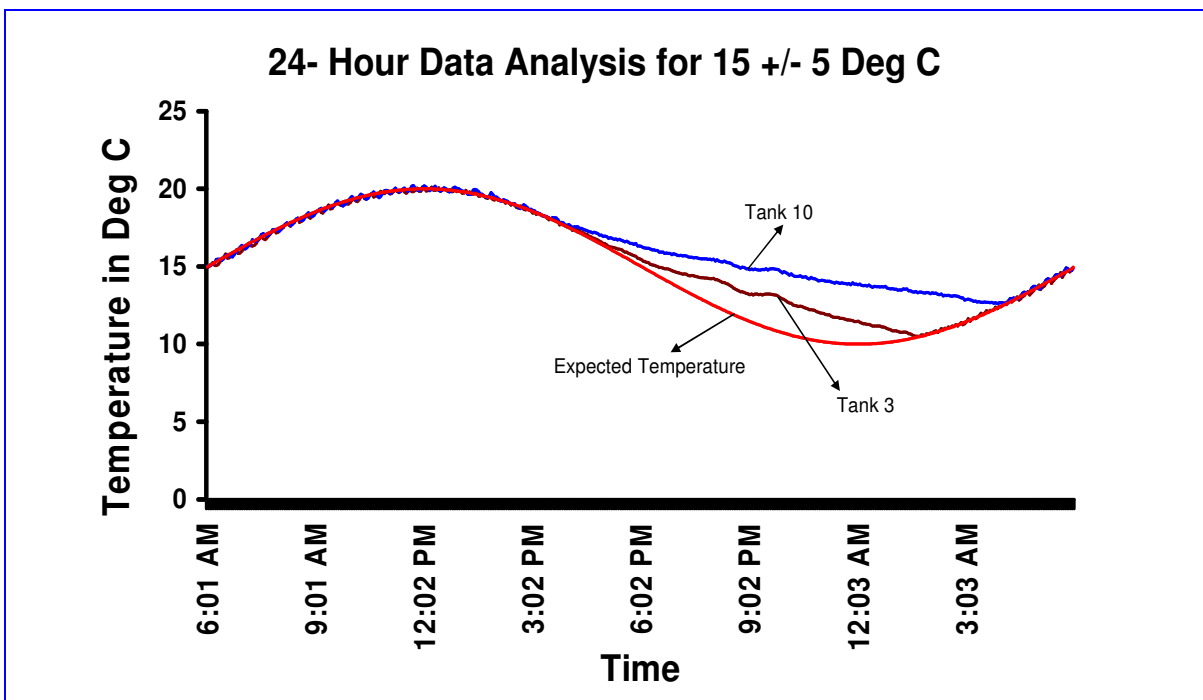
Figure 4.17 Both tanks experienced difficulty in cooling. However, both reached 20°C at noon as expected.
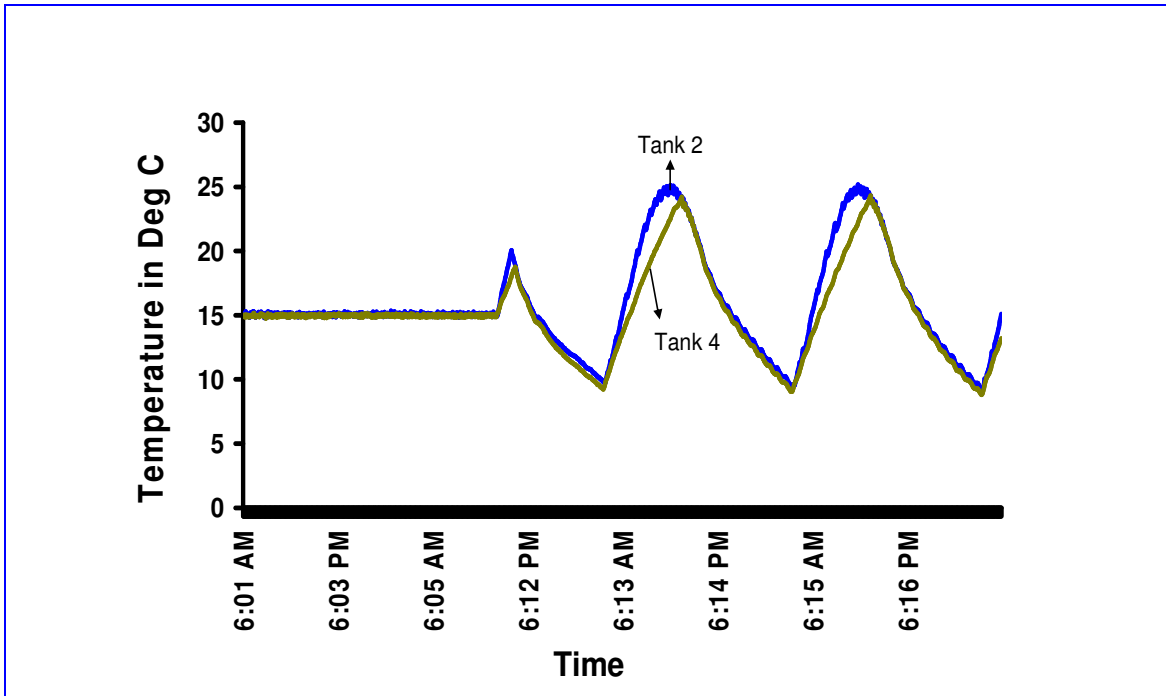
Figure 4.18 Tanks were consistently maintained around 15°C before fluctuations.
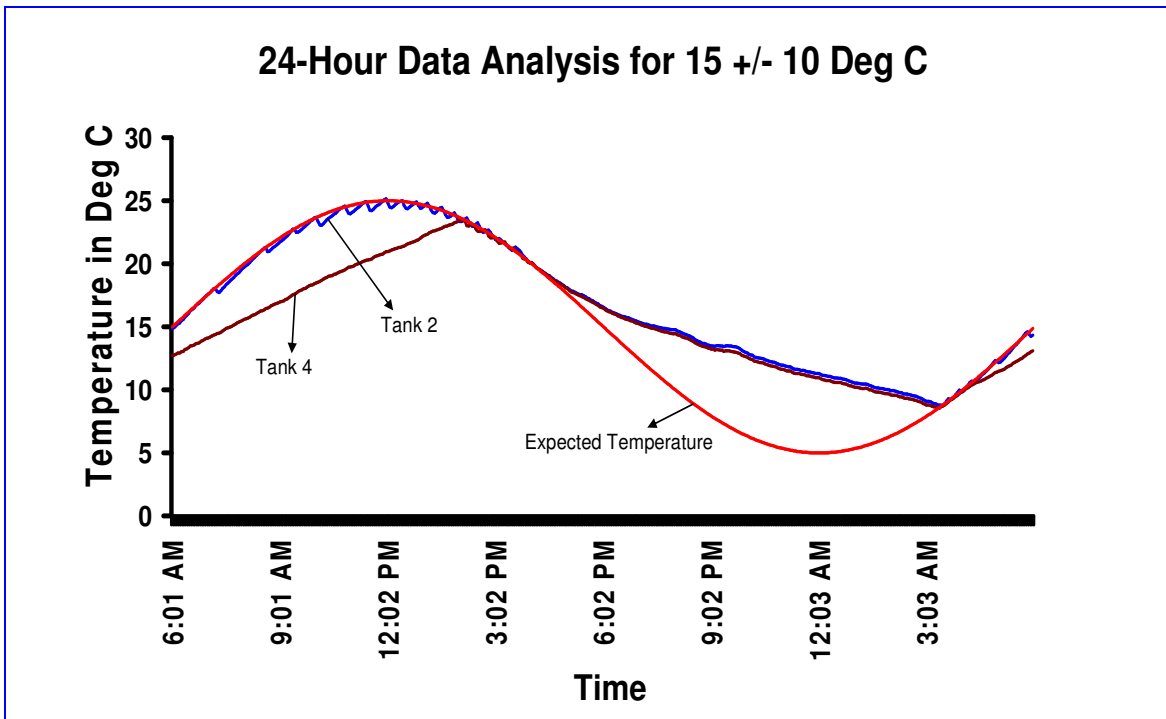
Figure 4.19 Tanks did not reach 5°C at midnight. Although tank 2 reached 25°C at noon as expected, tank 4 had difficulty in heating.

4.4 Concerns, Trouble Shooting, and Limitations

4.4.1 Experiment 1: Maintaining Constant Temperatures

1. **Regime 1 (20ºC):** As expected, average temperatures in tanks 1, 6, 7, and 8 were close to each other. There was an electrical glitch in the power bars associated with tanks 7 and 8. The solid state relays for tanks 7 and 8 had a loose contact which was resolved by rewiring the power bar. Except for the occasional electrical glitch in tanks 7 and 8 between 09/07 and 09/11, average temperatures of all four tanks were with ± 0.2 ºC which may be acceptable. The process control system was successful in maintaining water temperatures around 20ºC.

2. **Regime 2 (15ºC):** Although, water temperatures in six tanks were dropped @ 1ºC/day, tank 11 experienced a sudden increase in water temperature due to ill-functioning of electrical outlet. Replacing the power bar resolved this problem. However, for tanks 9, 10, and 11, the problem was of electro-mechanical type. Because the power bar and chiller were connected to the same power source, the distribution of power was uneven. At times it was found that both chiller and power bars were both without power. Also, the switch of the power bar was corroded and became dysfunctional. We presume that salt accumulation on the power bar was responsible for this corrosion. The power bars were disconnected and rewired by bypassing the reset button and allowing direct flow of electricity in the power bar. Hence forth, the water temperatures in tanks were within 0.2ºC from 15ºC.

3. **Regime 3 (10ºC):** Apart from minor electrical and mechanical issues, the tanks were within 0.25ºC from the assigned temperature. However, the ambient air temperature played a role in temperature and system dynamics in this regime.

Because four 2000-W chillers were continuously on (24 x 7), the chillers generated heat, which prevented cooling of water efficiently. Although the room was air-conditioned by an 1800-W window air conditioner, the room temperatures increased to 30-32ºC. An alternate and efficient method of energy utilization may be considered to make the system efficient in terms of cooling. Alternately, methods to use the energy generated by the chillers to heat the tanks in lieu of heaters may be developed.

4.4.2 Experiment 2: Imposing Temperature Variations

1. **Regime 1 (20 ±1°C):** The system was found to be effective in simulating the regime. Because the variation was only ±1°C, system was able to provide cooling and heating efficiently.

2. **Regime 2 (20 ± 5°C):** This was probably the optimum regime for this system. Although a variation of ± 5°C was imposed, both tanks were able to maintain water temperatures close to the expected (Figure 4.12). Because system functioned efficiently for this regime, experiments with other animals that are temperature sensitive can be carried out. Also independent tanks in the system will allow such experiment.

3. **Regime 3 (15 ±1°C):** Although a variation of only ± 1°C was imposed, tank 11 had difficulty in cooling water. This was because of malfunctioning of one of the chillers. Tanks 1 through 8 were connected to first chill tank that had two chillers. But tanks 9 through 16 were connected to second chill tank where one of the chillers malfunctioned. One chiller in second chill tank had to provide cooling to tanks 9 through 16.

4. **Regime 4 (15 ± 5°C)**: The system had difficulty in cooling water to 10°C. It appears that two chillers together were not able to provide cooling to eight tanks simultaneously. Also the holding capacity of chill tank was around 500 L. However, total water that was to be cooled was around 1600 L. To be able to achieve optimum cooling in all tanks, additional chiller may be needed or bigger chill tanks could be used. For tank 10 the difference between expected and actual water temperatures was more than tank 3 because of chiller malfunction in second chill tank. However, additional chiller is needed for this chill tank also.

5. **Regime 5 (15 ± 10°C):** Both tanks had difficulty in cooling due to reasons explained in regime 4 (above). Although cooling patterns were similar for tanks 2 and 4, heating was different in tank 4 than tank 2. Heating in tank 2 was as expected. Heating in tank 4 was slow because one of the heaters in tank 4 became dysfunctional.

4.5 Discussion

These systems were effective in providing both constant temperatures and variable time-temperature histories. However, there are limitations on the effective rates at which heating and cooling can be performed by the system. Data presented in this chapter quantifies this information which can be useful for design of additional systems. Theoretical predictions as well as empirical data were used to size these systems, but the practical limitations, in particular, stem from excessive heat buildup due to the chillers. Future designs of both hardware and software should consider these limitations. For example, one could size equipment larger (bigger pumps and efficient heaters and chillers) to exchange more heat. Another option would be to heat some

tanks while chilling others to avoid simultaneous heating or cooling in all tanks. This will not only reduce load on chillers but also minimize concomitant overloads. Also, feed forward control could be used to reduce peak heating and chilling rates for a given hardware configuration. Clearly, the system was quite functional, but these additional changes could improve system performance and reduce energy costs.

4.6 Conclusions

1. **Experiment 1:** The temperature control system was able to maintain water temperatures within 0.2°C for a temperature range of 10 and 25°C.

2. **Experiment 2:** The temperature control system was able to simulate water temperatures within 0.2°C, for regimes 20 ± 1°C, 20 ± 5°C, 15 ± 1°C. If experiments were to be conducted below 8°C, additional chillers may be necessary. Alternatively, the program may be modified in such a way that eight tanks can be cooled at noon while other eight tanks can be heated at noon. This will conserve cold water and reduce load on chiller.

Chapter 5

Development of Real-Time Temperature Control Software for Use in Oyster Research

5.1 Introduction

Development and use of temperature control system has taken research on oysters to a new level. Experiments on oysters were carried out under constant and variable regimes to understand biology at a cellular level. But, in all experiments the control water temperature for oysters was set at 28°C. However in natural waters, this may be true only for a specific period of time. In natural waters, water temperature changes with time. To compare oysters under different regimes with natural waters, it is necessary to have a control that is same as actual water temperature from which oysters were procured.

The objective of this portion of the project was to develop real-time temperature control software for use in oyster research. The developed software *"Realwire"* was able to collect real-time water temperature data for Terrebonne bay from the official website of Louisiana Universities Consortium (LUMCON).

5.2 Methods

This part of the project was the result of collaboration between Louisiana University Marine Consortium (LUMCON), Louisiana State University Aquaculture Research Station (LSUARS) and Biological and Agricultural Engineering (BAE). An active server page (ASP) was made available to us by LUMCON authorities that contained time-temperature data for waters at Terrebonne bay.

Two types of software were used to develop real-time software:

- Python programming language Version 2.3: This is an object oriented programming language, compatible with Windows operating systems. The syntax is similar to C or C ++.

- GNUwget: A software package used mainly for retrieving files using hyper text transfer protocol (HTTP), also called a internet language. This package can work on Windows platform and even when the computer is logged off.

Python programming language, Version 2.3 and GNUwget software were downloaded from www.python.org and www.gnu.wget.org/software/wget on a personal computer in Aquacultural Engineering laboratory at LSU Biological and Agricultural Engineering. Because software (both) was public domain software, they could be freely used without licensing. Utilizing the downloaded software, a program was written using Python programming language. This new software was called *Realwire* and a copy of which is available in Appendix B. *Realwire* functioned in two steps:

1. Accessed lumcon.edu/ weather every 3 min and recorded time and corresponding water temperatures of Terrebonne bay.

2. Stored time and corresponding water temperatures on the host computer (BAE) in a text file. For easy bookkeeping and data analysis, each time-temperature observation was made to store on the next line, but in the same file. The file names were indicative of the date of data capturing. For example file named, "**0505.txt**" contains temperature data for fifth day of May. Also, the format supported our earlier programs that we developed for measuring and controlling water temperatures using temperature control system.

72

5.3 Results

   Realwire was able to read, collect, and store water temperature data associated with

Terrebonne bay, from the LUMCOM website. A typical comparison of actual website

temperatures (Figure 5.1) and temperatures captured by our software is shown (Figure

5.2).
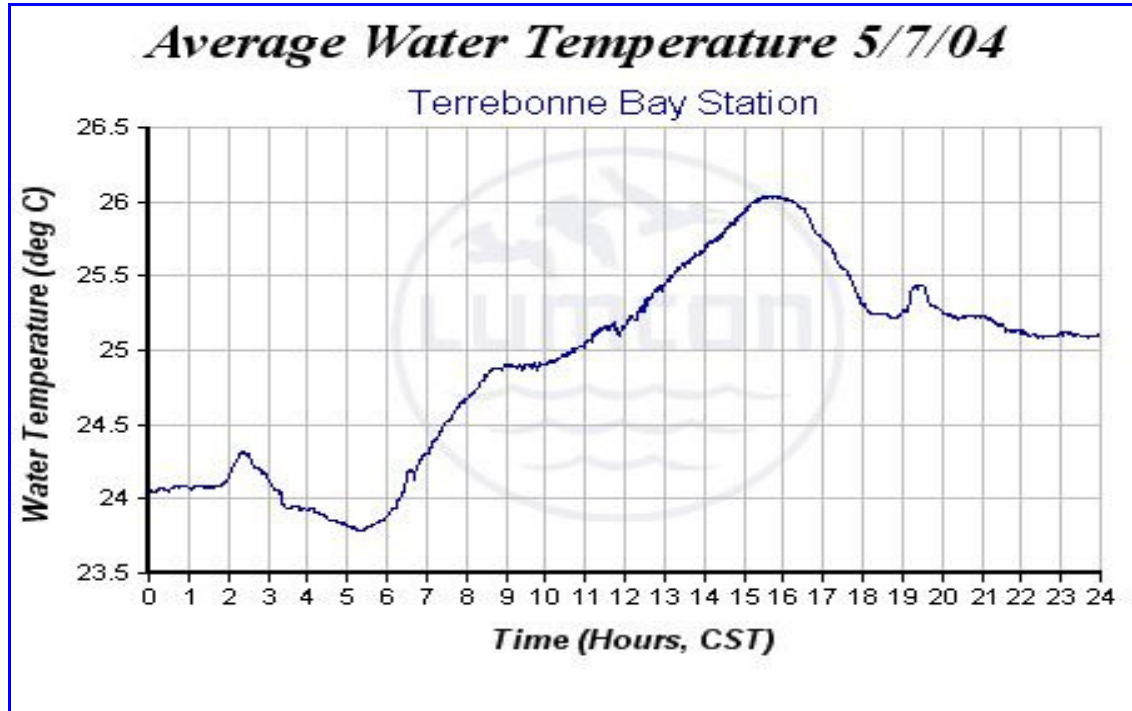
Figure 5.1 Typical water temperatures of Terrebonne bay recorded by LUMCON on
05/08/2004. Courtesy: weather.lumcon.edu/chartdata/cst

5.4 Discussion

   Although *Realwire* was able to collect real-time data, the system could not be tested for

real-time simulations due to time considerations. However, data files obtained from

Realwire were made compatible with our earlier programs that were developed on

Microsoft Windows Visual C+ + environment.
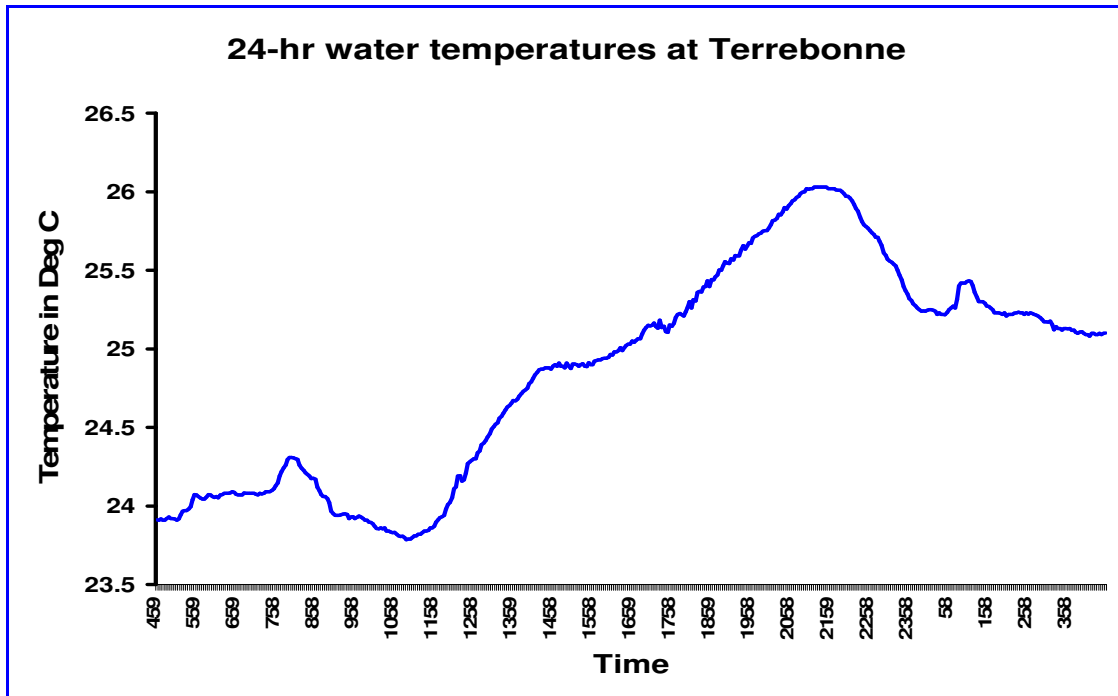
Figure 5.2 Temperatures collected for Terrebonne bay using *Realwire* for a 24-hr period (05/07/2004) were same as that were available on lumcon.edu

Development of *Realwire* could open new possibilities in oyster research. The same software could be modified and water temperatures across various weather stations can be collected. Because independent temperature control is available, it may be possible to simulate water temperatures of different locations in laboratories.

5.5 Conclusions

Real-time software namely *Realwire* was developed in Python programming language that can access and collect time-temperature data of water at Terrebonne bay from LUMCON website. *Realwire* was able to collect and store the water temperature every three minutes on a personal computer at BAE. The data can be utilized to simulate real-time coastal temperatures in laboratory.

Chapter 6

Summary and Conclusions

The goal of this thesis was to develop process control systems that improve research facilities for eastern oysters. A feedback based temperature control system was developed which helped researchers to focus on oyster research year round.

In the first part of this project (Chapter 3), a series of independent recirculating tanks connected to a central computer, capable of automatic temperature control were designed and developed in the Aquacultural Engineering laboratory at the Louisiana State University Aquaculture Research Station (LSUARS). Sixteen food grade 250-L tanks were used as holding tanks for oysters. Each tank was capable of independent heating and cooling mechanism. Biological filters coupled with a 5-cm airlift maintained water quality in each tank. The computer measured water temperature via type-T thermocouples that were connected to a CIOEXP32 multiplexer. The control over water temperature was achieved by a pair of PCIMDAS1602/16 analog and digital (A/D) boards through 3-32VDC solid state relays. When tested initially, the control over water temperatures were within $\pm\ 0.2°C$ for a temperature range of 10°C and 25°C. Besides maintaining water temperatures, the system was also able to maintain water quality in tanks.

In the second part of the project the temperature control system was used for studies on somatic cell proliferation rates in oyster under a variety of temperature regimes. Although the objective of this study was primarily to understand oyster cell biology, the temperature control system was evaluated from an engineering perspective. Although various temperatures regimes were imposed on the tanks by the system, it was found that,

with the present system configuration, temperatures below 8°C were difficult to maintain. However for regimes between 10°C and 25 °C, the system maintained water temperatures within ±0.2°C. Biology and engineering derived mutual benefits from this study. The feedback obtained from oyster responses were used to modify the system for efficient control of water temperatures. For instance, when found that heating and cooling at five minute intervals were stressful on oysters, the heating and cooling intervals were reduced from five minutes to three minutes. This reduced stress on oysters and improved efficiency of system by reducing excess cooling and heating.

Thirdly, software called *Realwire* was developed for measuring real-time water temperatures at Terrebonne bay. Help from Louisiana University Marine Consortium (LUMCON) was sought for this portion of study. An active server page (ASP) relevant to water temperature data of Terrebonne bay was made available to us. This page and official website of LUMCON were used to develop *Realwire*. Software called Python and GNUwget (both are public domain and freely available on internet without licensing) were used for developing **Realwire**. The software was written in Python programming language which was similar to Microsoft Visual C++. The software read the temperature data from the ASP every 3 min and stored on a computer in Aquacultural Engineering Laboratory at LSUBAE. But the collected temperature data could not be used for real-time simulations due to time constraints.

Process control systems are versatile and can be used for a variety of applications even in aquaculture research. Although, our system was able to maintain water temperatures well within ± 0.2°C, the design used for this system was based on one of the simplest and basic models available. Improvements are possible for efficiency and controllability. The

present system was designed on a feedback based mechanism. Extensive modifications are possible by incorporating feed forward controls, which is a complimentary to feedback process. For instance, in the present system, control of water temperatures was affected by changes in ambient air temperature. If this effect of ambient air temperature on water temperature can be quantified, the effect can be included in control algorithm. In other words, system can compensate for the effect. This can result in better control over water temperatures in tanks. Even the control used in the present system was limited to on and off. Developments incorporating proportional, integral and derivative or combination of all three (PID) are possible. For the present system, controllability can be improved by adapting a proportional control approach. For example, if difference between expected and actual water temperatures was small (±0.1°C), 100 % pumping of cool water is probably not needed. The pump may open its valve by 25%. This can result in monetary savings as cool water is conserved. However special pumps are needed for this kind of control.

Future work may be undertaken as a three-pronged approach:

1. Research efforts may be directed towards modifying the present system to be energy efficient. For example, it was observed that ambient air temperature was increased to 32°C. This energy may be used to heat the water in tanks in lieu of heaters. This can translate into monetary savings. Alternatively, the ambient temperature was one of the reasons why cooling below 8°C was slow. Although pumps were working at their full capacity, temperature gradient between ambient air and water in tanks made cooling difficult. If ambient heat can be utilized for

heating water in tanks, cooling of water below 8°C may be possible. Engineering

and economics problems may be solved with one research.

2. The control system may be developed for better controllability. Different types

   on controls (proportional, integral and derivative) may be used. Although each

   type of control has its own merits and demerits, each can be evaluated and most

   feasible of the three can be selected. Other approach may be to use feed forward

   mechanism. However, both approaches involve electronics and control

   engineering inputs.

3. Internet may be used to program the tanks to mimic coastal temperatures.

   Software for this purpose has been developed and available. However, studies to

   evaluate performance of the system to maintain real-time temperatures may be

   carried out. In fact all three control methods (proportional, integral and derivative)

   and feed forward approach could be used to make this system truly versatile.

Installation, Configuration and Operation of Data Acquisition System (DAS) Boards

Two types of data acquisition boards were used in this project. A through description of installation, calibration, and operation for each of the boards follows.

1. **PCIM-DAS1602/16 Analog and Digital I/O Board:** Manufactured by Measurement Computing, MA, this is a multifunction measurement and control board (Figure A. 1) The board is a PCI-compatible, plug and play board. Finer details of working, configuration, and calibration are illustrated in later sections of this chapter. For simplicity, the PCIM-DAS1602/16 Analog to Digital I/O Board is referred to as the A/D board.
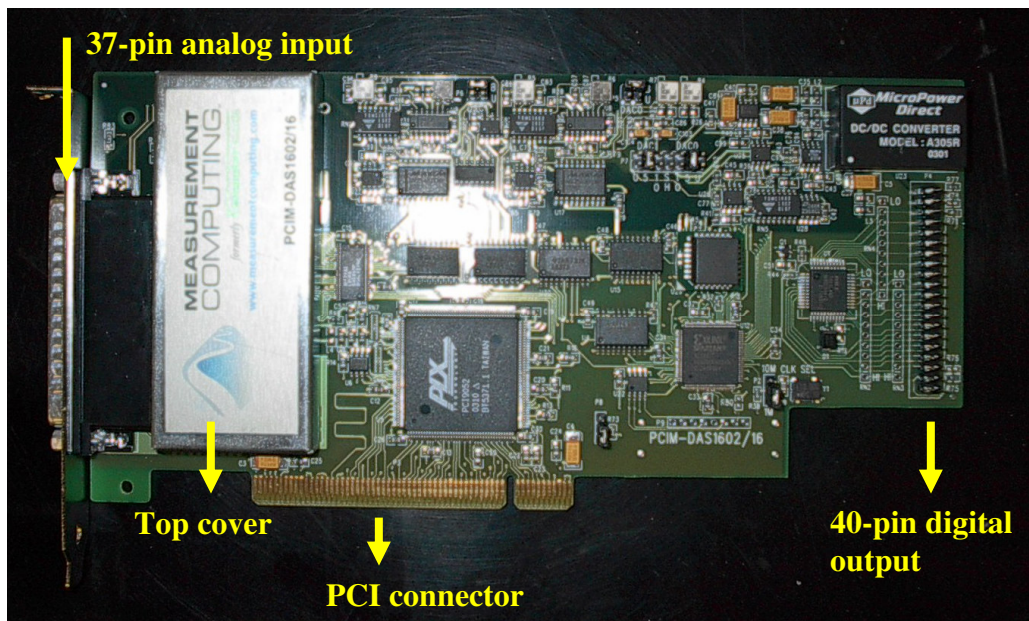
Figure A.1 PCIM-DAS1602/16 Analog to Digital I/O Board with the top cover. The analog inputs and digital outputs are shown using arrows. Top cover was opened to change the channel jumper switch from 8 differentials to 16 single ended channels.

2. **CIO-EXP 32 Multiplexer Board:** Also manufactured by Measurement Computing, MA. One of the main functions of this board is thermocouple signal

processing. The working details such as installations, configuration, and calibration of this board are explained later sections of this chapter. For simplicity, the CIO-EXP32 Multiplexer board is referred to as EXP32 board.
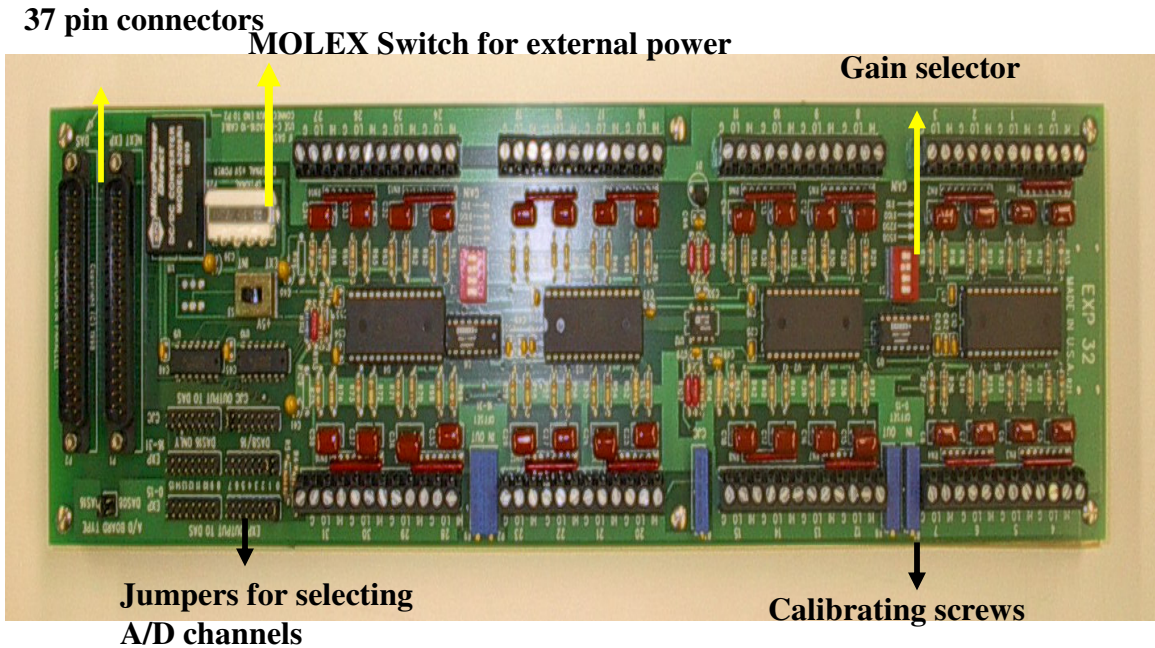
Figure A.2 CIO-EXP 32 Multiplexer board was used in this project to measure temperatures using thermocouples. Important components are shown using arrows.

PCIM-DAS1602/16 Analog and Digital I/O Board:

Installation of A/D board

Before installing the A/D board, it is necessary to load *InstaCal* software on the PC. *InstaCal* has information that is needed for the PC to detect the A/D board. The following steps will explain the installation process.

1. Turn the computer off and open the cover of the CPU.

2. Insert the board into an available PCI slot, as shown in the Figure A .3

3. Close the top of the computer and switch on the power.

4. A dialog box appears indicating that a new hardware has been detected. Follow the steps and run *InstaCal* software and to review the configuration of the board. The PC may prompt you to restart before you begin configuration of the A/D board.
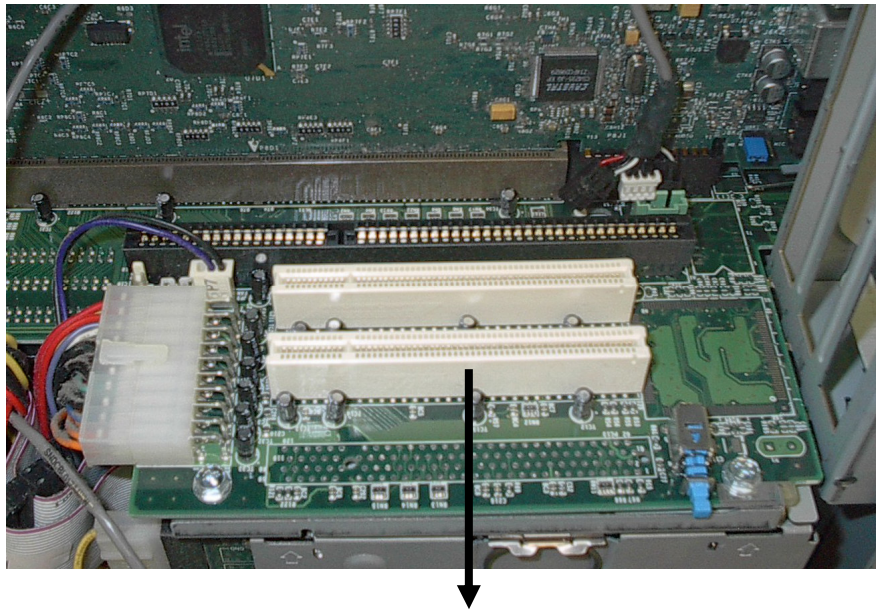
PCI slot on the mother board of the computer. The A/D board was inserted into one of the slots.

Figure A.3   Available PCI slots are shown. The A/D board is inserted into one of the slots. When the computer is restarted, a dialog box appears indicating that the new hardware has been detected.

Connecting the Board for I/O Signals

  The A/D board has a 37-pin male connector to receive analog inputs (Figure A .5). The analog connector can be accessed from the rear of the PC on the back plate. To connect the analog connector, a female 37-pin D-type connector may be used. The digital signals are transmitted via a male 40-pin connector that is mounted on the side of the A/D board (Figure A. 5). The digital signals are brought to the back plate of the PC via a BP40-37 cable (Figure A .4), manufactured by Measurement Computing Inc, MA.

81
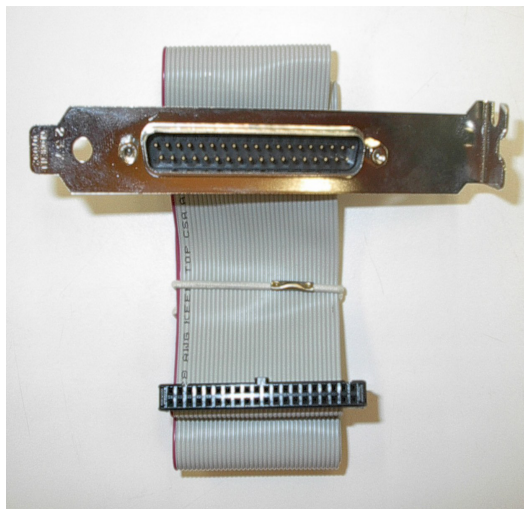
Figure A.4 A BP40-37 cable manufactured my Measurement Computing Inc, MA, was used to bring digital signals from the rear of the board to the rear of the PC on the back plate.

Default Board Configuration

The A/D board is factory configured to default settings (shown below). It is advisable to check and change the configuration settings depending on the requirement and conditions imposed by the design. When used with an EXP32 board, the channel select switch must be changed to 16 single ended channels, rather than eight differential channels (Default setting). The following table shows default settings of the switches and jumpers.

Switch/Jumper description	Default setting
Channel select switch	8 channels
A/D range select switch	Bipolar
Trigger edge select switch	Rising edge
Clock select jumper	1MHz
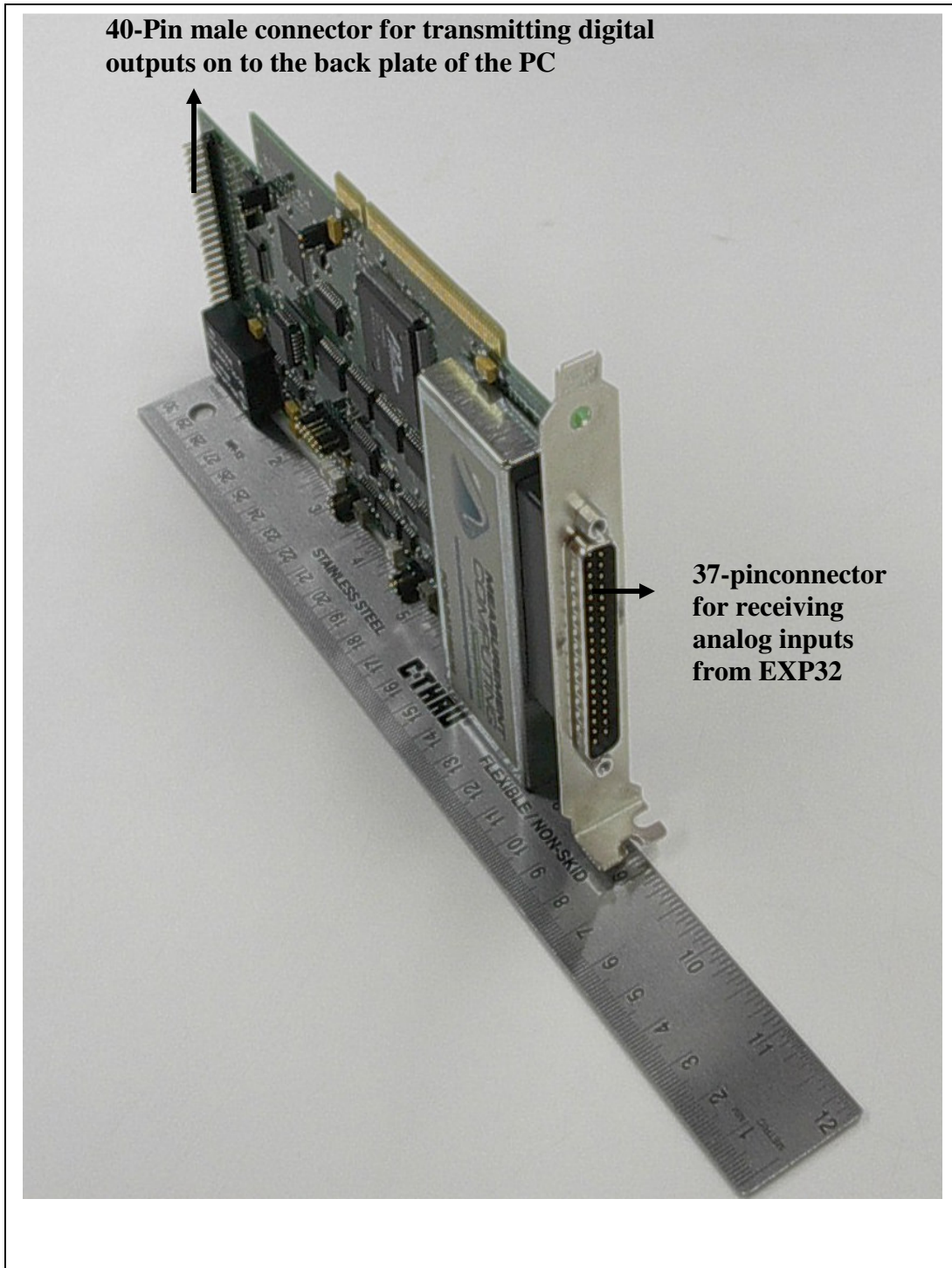
40-Pin male connector for transmitting digital outputs on to the back plate of the PC

37-pinconnector for receiving analog inputs from EXP32

Figure A.5 A side view of A/D board with analog inputs and digital outputs. A male 37-pin connector received the analog signals from EXP32 board. The digital signals were transmitted by BP40-37 cable, which brings the digital signals to the back on PC on the back plate.

It may be noted that the channel select switch and A/D range select switch are covered by the metal name plate on the A/D board (Figure A .6). To change the settings, the name plate has to be removed using a screw driver.

Channel Selection for the A/D Board

The A/D board can be used with either eight differential or 16 single ended input channels. For our project, 16 single ended options were used. Since, thermocouples were used for measuring temperatures, 16 single ended options reduced the electrical noise generated by the system. The top cover (Figure A .6) on the A/D board was removed and the jumper switch was changed from 8 differentials (default) to 16 single ended options.
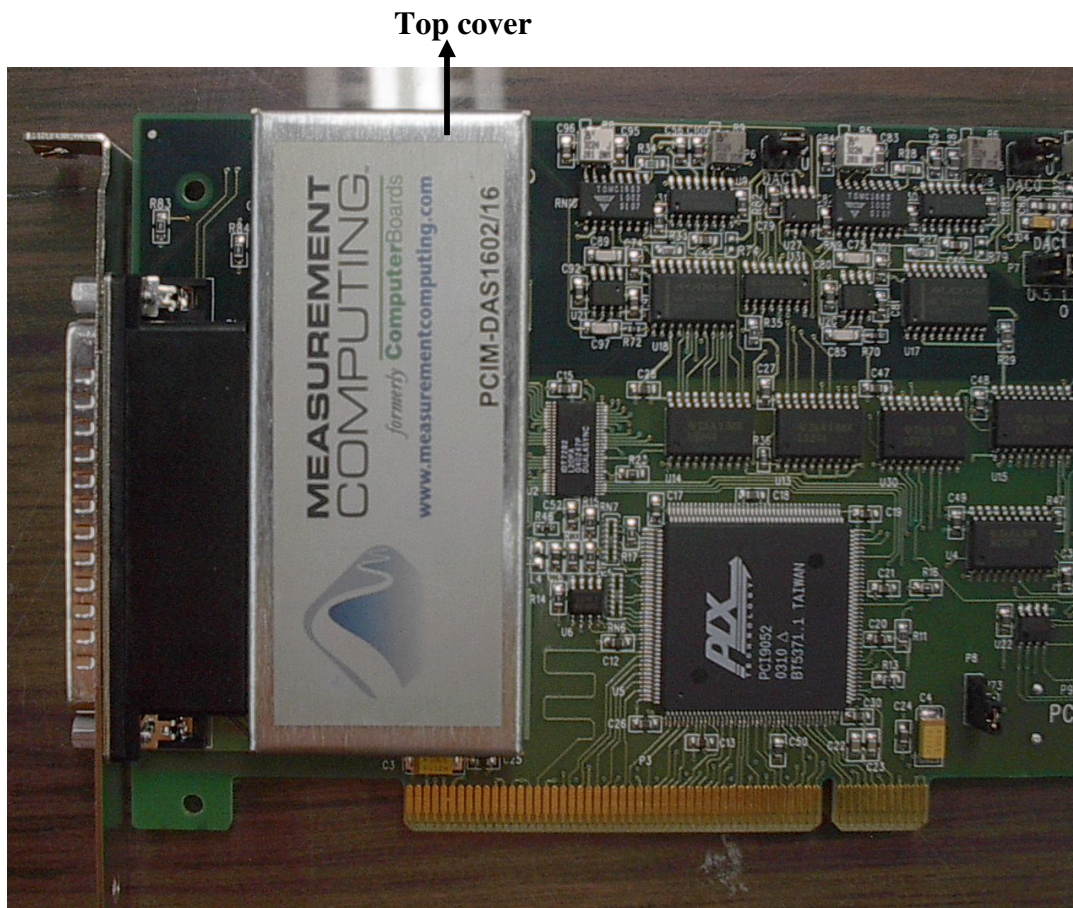
Figure A. 6 The top cover of the A/D board was removed to change the jumper switch from 8 differential channels to 16 single ended channels to reduce electrical noise.

Addition of EXP32 Board to A/D Board

This section explains the steps for adding data acquisition devices to the A/D board.

The steps are:

1. Open *Instacal* software as shown below
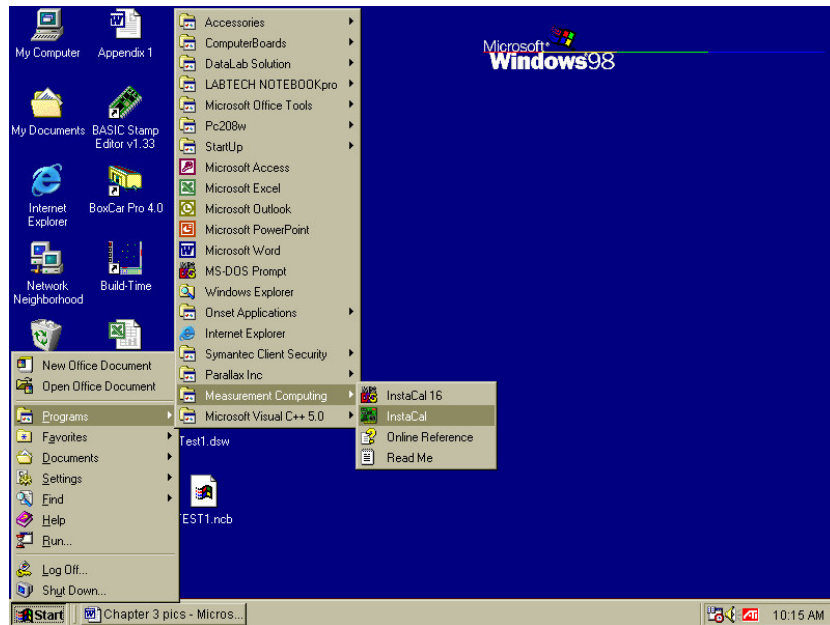
2. Double click on the *Instacal* icon. The following screen appears showing the all
   data acquisition boards used by the PC.

3. Right click on the PCIM-DAS1602/16 board to view a pull down menu (See following figures).
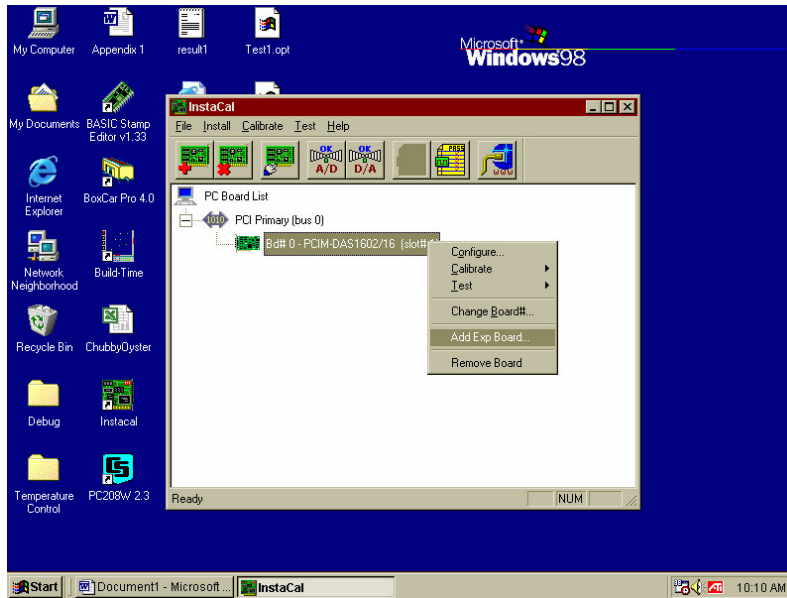
4. Select the board needed for data acquisition. For this project EXP32 boards are selected.

5. Connect the EXP32 board and analog connector of the A/D board with a special 37-conductor cable, CEXP2DAS16-10, manufactured by Measurement Computing, Inc, MA.

CIO-EXP32 Multiplexer Board: Figure A.2 shows a picture of the board. This is a 32-channel (0-31) external board that can be mounted either on a bench top or secured in a water proof case.

Powering the Board

Power required for the board is 5VDC. The power source can be either internal or external. The A/D board via a 37-conductor cable supplies internal power, which is adequate for two EXP32 boards. If internal power is used, the +5V switch (S3) on the

board should be set to INT (Figure A. 7). When three or more EXP32 boards are used,

external power may be required. The Personal Computer (PC) connected via a C-

PCPOWER-10 MOLEX cable (manufactured by Measurement Computing, MA) supplies

the external power. To connect this cable, the computer has to be opened after turning off

the power. One end of the MOLEX cable is connected to the expansion power connectors

in the PC, while other end is connected to the EXP32 connector labeled "OPTIONAL

EXTERNAL +5V POWER" The +5V switch (S3) on the EXP32 board should be set to

EXT.
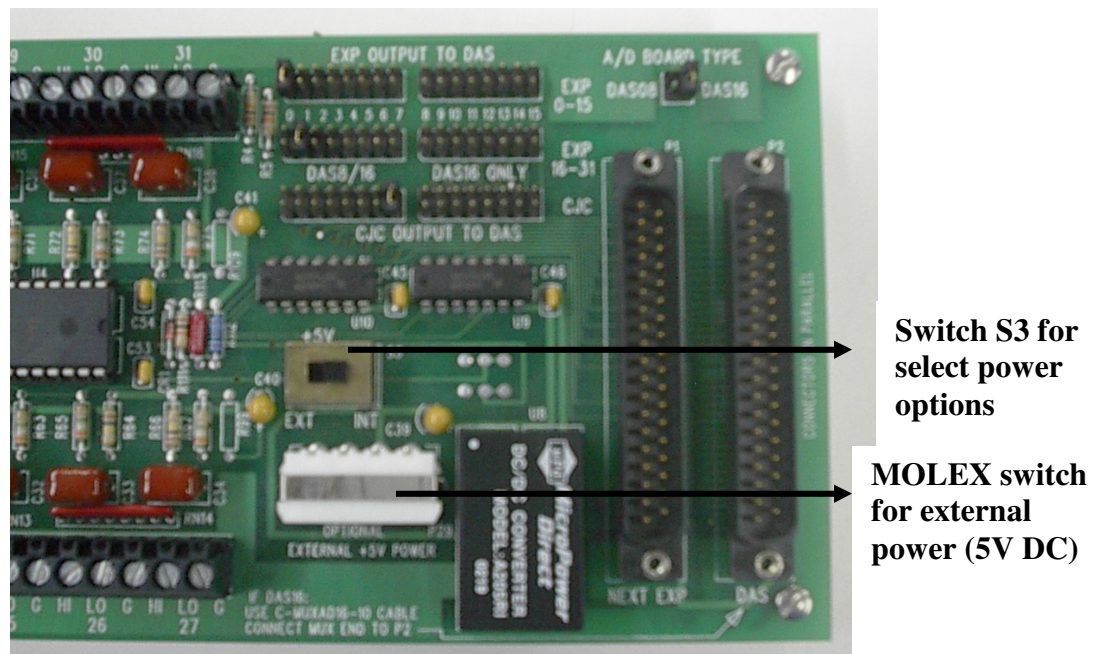
Figure A.7   Internal power (5V DC) was selected for this project. External power is required when two or more EXP32 boards are used.

Right Connections

On the EXP32 board, there are two identical 37D connectors, P1 and P2 (Figure A. 8).

The first connector (P2) is used to connect the EXP32 with an A/D board. Because the

pin relationships are not same between the EXP32 and A/D board, a special 37-conductor

cable, CEXP2DAS16-10 (manufactured by Measurement Computing Inc, MA) is required. One end (labeled MUX) of the CEXP2DAS16-10 (Figure A. 9) cable connects the first connector (P2) on the EXP32 and the other end connects the A/D board's analog connector on the back plate of the PC. The second D37 connector (P1) is used only when two or more EXP32 boards are used.
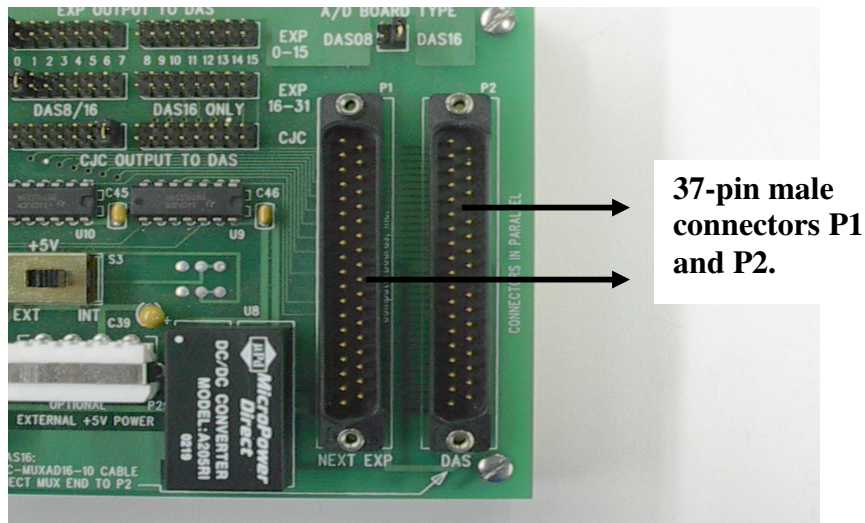
Figure A.8   Male 37-pin connector P2 was used by CEXP2DAS16-10 to connect EXP32 with A/D board. P1 may be used when more than one EXP32 boards are used to capture data.

Figure A.9 A CEXP2DAS16-10 cable was used to connect EXP32 with the A/D board. The cable connects the A/D board from then rear of the PC. One end marked MUX was connected to EXP32 and other end connected the A/D board

Channel Selection

The EXP32 has 32 channels (0-31) for input. These are divided into two sections of 16 channels each. Section 1 is connected to the jumper block labeled "EXP 0-15" and section 2 is connected to jumper block labeled "EXP 16-31". Each section of 16 inputs is read by one channel of the A/D board. For this project channels 0-15 of EXP32 are multiplexed into channel 0 of the A/D board and channels 16-31 are multiplexed into channel 1 of the A/D board. Figure A.10 shows the sections 1 and 2 and corresponding jumpers set to channels 0 and 1 for the A/D board.
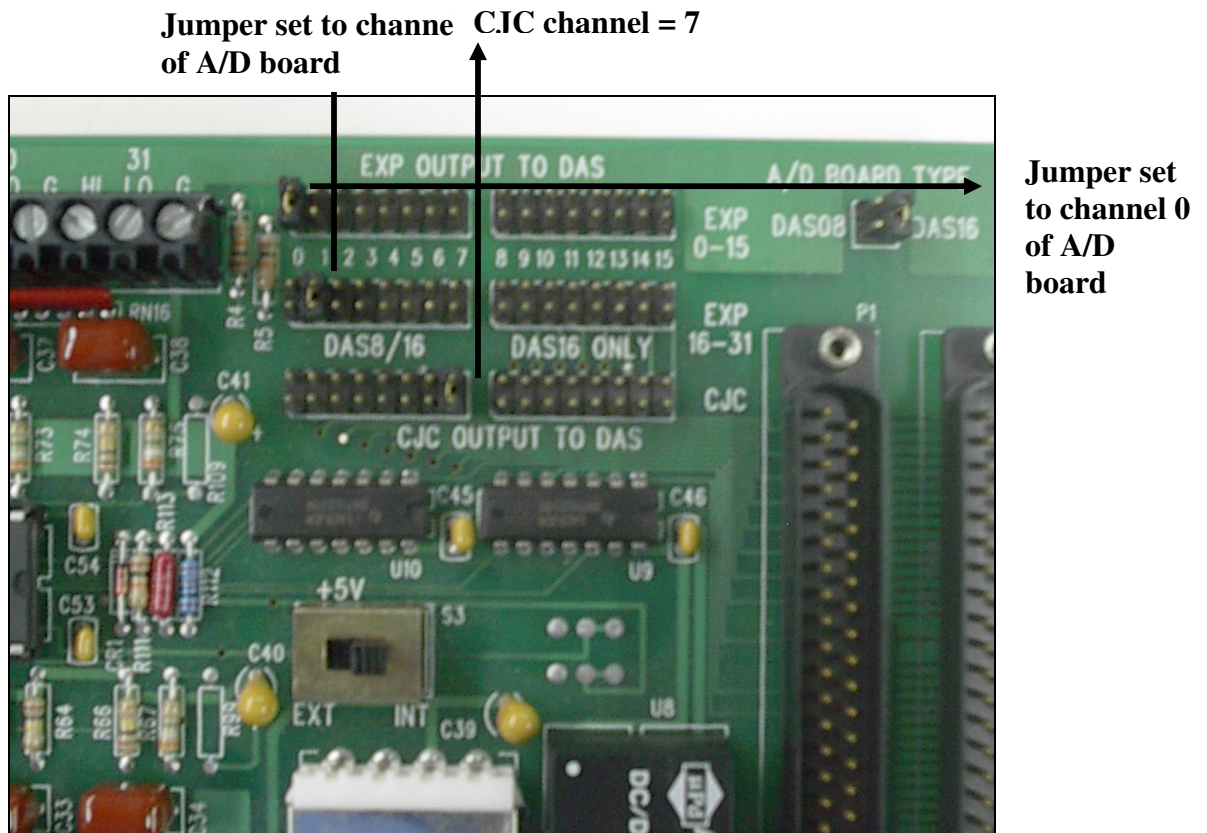
Figure A.10   Jumper block for first fifteen channels 0-15 was connected to channel 0 of A/D board and jumper block for channels 16-31 was connected to channel 1 of A/D board.

The channel inputs are screw terminals capable of accepting 12-22 AWG wire and each channel consists of a screw terminal for signal high, signal low and ground. All inputs from signal source are floating differential, requiring two wires, signal high and signal low. The ground is not used for temperature measurements using thermocouples. However, all grounds (G pads) are bridged using solder.

Cold Junction Compensation (CJC)

The EXP32 board has a semiconductor temperature sensor to measure the temperature of the screw terminals on the board. This is cold junction temperature. When thermocouples are used to measure the temperatures, cold junction temperatures are needed for accurate temperature measurement. The CJC jumper has to be set on the corresponding channel (Figure A.10). For this project, the CJC jumper was set to channel 7 (the default configuration).

Selecting the Gain

For each section of 16 input channels, there is a bank of four DIP switches, located in the central portion of each section (Figure A.11). These four switches control the amplification factor of the thermocouple input signals. One or more switches can be turned on or off simultaneously depending on the desired amplification (gain). It is important to note that these gains are additive. A gain of 100 was selected for this project.

Configuring EXP32

Each channel of the EXP32 has to be configured before measuring temperature using thermocouples. This can be done by bridging the V, G, and C pads on the etch side (underside) of the EXP32 by soldering the pads. It is important to use solder provided by

the board manufacturer only, since the solder contains flux core and is water-soluble. The

A/D board has to be configured to 16 single-ended to be compatible with EXP32.
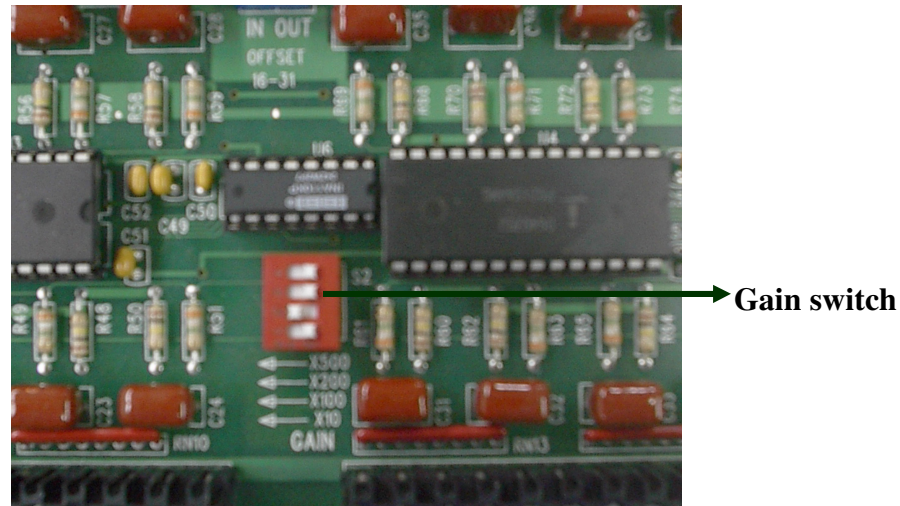
Figure A.11 A gain of 100 was selected for this project by turning on the second switch.
The gains were additive across switches and ranged from 10 to 810.

Calibration of EXP32

  Calibration of EXP32 requires *InstaCal* software supplied by board manufacturer.

InstaCal is the installation, calibration, and testing software for boards manufactured by

Measurement Computing Inc, MA. Ideally, *InstaCal* and *Universal Library* software

must be loaded before adding any boards to the PC. *Instacal* is compatible with Windows

3.x, Windows 95, Windows 98, Windows 2000, Windows ME, and Windows XP.

The software is stored in a default folder called **Measurement Computing**. After

installation of *InstaCal* , the board properties can be changed by double clicking on the

board name. For the present project, the board properties were selected as:

1. CJC: Channel 7

2. EXP32 channels 0-15: channel 0 of A/D board, 16-31 to channel 1 of A/D board

3. Power options: +5VDC External

4. Gain: 100

The calibration process for EXP32 is simple. The following steps explain the steps for calibration

a. Open *Instacal* software and right click on the EXP32 board as shown below.
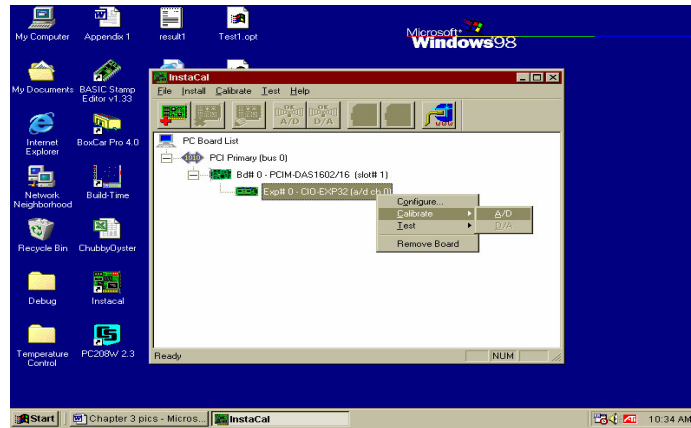
b. Select the A/D option and the following screen appears for calibrating the EXP32 board. Follow the instructions and set the CJC temperature to the room temperature by adjusting the calibrating screws (Figure A.13.)

92
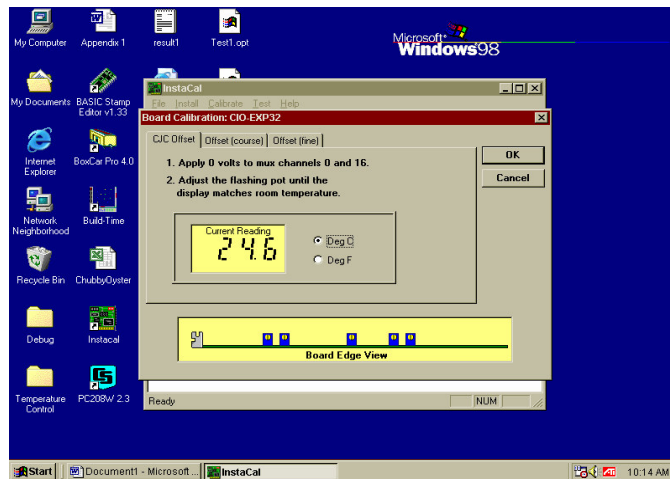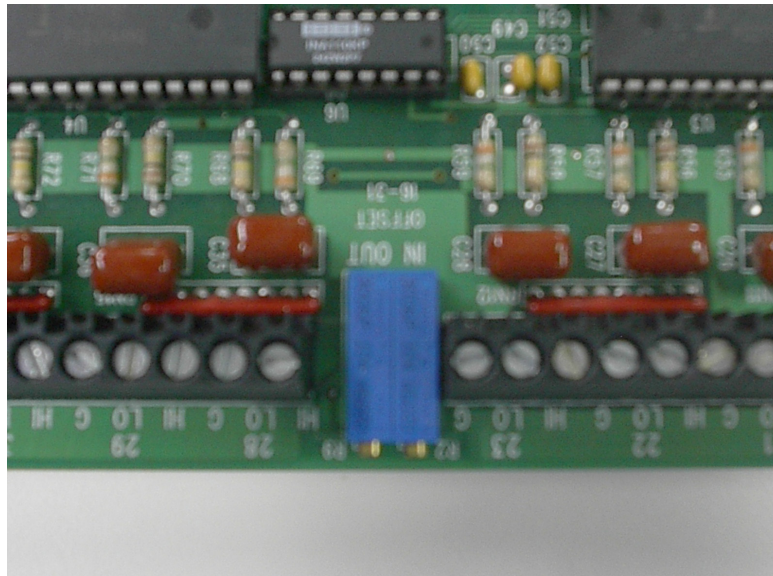
Figure A.12 Then calibrating screws were turned either clockwise or anti-clock wise depending on whether the measured temperature was higher or lower than actual room temperature.

   After calibration, the EXP32 was ready for measure the temperatures. This simple

process, coupled with regular calibration of thermocouples allowed for accurate

measurement of temperatures within $\pm$ 0.5 °C.

Appendix B

Software Programs for Temperature Control

Following were software programs that were developed for the process control system. Three programs were developed:

1. **Dead Band**: This program was developed to maintain constant water temperatures in tanks. Written in Microsoft visual C++, this program allows for constant temperatures and temperatures within a range. For example, if a temperature range of 22-23 °C is desired, then this program will maintain water temperatures between 22 and 23°C. Control will be activated only of water temperatures go beyond the range.

2. **Sine Code**: This program was developed (in Microsoft visual C++), to impose diurnal temperature variations in tanks. For example, if a variation of ± 2 °C is to be imposed on an average temperature of 20°C, then highest temperature (22°C) will be at 12 noon and lowest temperature (18°C) will be at 12 midnight. Control will be executed for every three-minute cycle (which can be changed depending on application).

3. *Realwire*: This program was developed to capture real-time temperature data of Terrebonne bay from LUMCON website. Written in Python (Version 2.3) programming language, the software reads temperature every three minutes and stores on a personal computer (PC) in Aquacultural Engineering laboratory at LSUBAE.

A copy of all software developed is reproduced below

Dead Band

```
#include<stdio.h>
#include<conio.h>
#include "c:\mcc\c\cbw.h"          //   INCLUSION OF HEADER FILES //
#include<time.h>
#include<dos.h>
#include<math.h>



void main()                       //  VARIABLE DECLARATIONS //
{

        float tempvalue,temperature[18][1000],sum[18],avg[18],tempavg[500][18];
        float lowtemp[18],hightemp[18];
```

```
float slope[18]= {(float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000,
            (float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000,

(float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000,

(float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000, (float)1.0000, (float)1.0000};
    float intercept[18]= {(float)0.0000,(float)0.0000,(float)0.0000,(float)0.0000,

(float)0.0000,(float)0.0000,(float)0.0000,(float)0.0000,

(float)0.0000,(float)0.0000,(float)0.0000,(float)0.0000,

(float)1.3000,(float)2.5000,(float)0.9999,(float)1.3000, (float)0.0000, (float)0.0000};

    int portnum,cond;
    int direction;
    int x=0,j,y=0,m,low=0,high=0;
    unsigned short A=0,B=0,C=0,A1=0,B1=0,C1=0;
    int
cmdoutA[8],cmdoutB[8],cmdoutC[8],cmdoutA1[8],cmdoutB1[8],cmdoutC1[8];
    double i;
    FILE *fp;
    time_t prevti,preti,prevti1,preti1;


    direction = DIGITALOUT;      //  OUTPUT PORTS INTIALIZATION //
    portnum = FIRSTPORTA;
    cbDConfigPort(2,portnum, direction);
    cbDConfigPort(1,portnum, direction);
    portnum = FIRSTPORTB;
    cbDConfigPort(2,portnum, direction);
    cbDConfigPort(1,portnum, direction);
    portnum = FIRSTPORTCL;
    cbDConfigPort(2,portnum, direction);
    cbDConfigPort(1,portnum, direction);
    portnum = FIRSTPORTCH;
    cbDConfigPort(2,portnum, direction);
    cbDConfigPort(1,portnum, direction);

    for(j=0;j<18;j++)       // INTIALIZATION OF VARIABLES //
    {
            sum[j]=0.0;
            avg[j]=0.0;
    }
    for(j=0;j<8;j++)
    {
```

```
        cmdoutA[j]=0;
        cmdoutB[j]=0;
        cmdoutC[j]=0;
        cmdoutA1[j]=0;
        cmdoutB1[j]=0;
        cmdoutC1[j]=0;
      }

      for(j=0;j<18;j++)       // READING IN THE DESIRED MAXIMUM AND
MINIMUM TEMPERATURES //
                                              //  FOR EACH
TANK   //
      {
              printf("Enter the maximum and minimum temperature for tank %d  : ", j);
              scanf("%f %f", &lowtemp[j],&hightemp[j]);
              printf("\n");

      }

      cond=1;
      while(cond==1)
      {
              time(&preti);
              time(&prevti);
              while(preti<prevti+1800)
              {
              time(&prevti1);
              time(&preti1);
              while(preti1<prevti1+300)
              {

              for(j=0;j<18;j++)
              {
                      cbTIn(2,16+j,0,&tempvalue,0);        // READING IN THE
TEMPERATURE VALUES //
                      temperature[j][x]=(tempvalue*slope[j])+intercept[j];
              }

              x++;
              for(i=0;i<30000000;i++)            //  TO INDUCE SOME DELAY //
                    { }
              time(&preti1);
              }

              printf(" Average Temperatures after the end of interval number
%d\n",y+1);
```

96

```c
            for(j=0;j<18;j++)
            {

            for(m=0;m<x;m++)
            {
                    sum[j]=sum[j]+temperature[j][m];        // CALCULATING THE
SUM AND AVERAGE TEMPERATURES //
            }
      // FOR EACH FIVE MINUTE INTERVAL //
            avg[j]=sum[j]/(float)(x);
            tempvalue=avg[j];
            printf("Tank  %d  = %f\n ",j, tempvalue);
            if(avg[j]<lowtemp[j])          // COMPARING WITH THE DESIRED
MINIMUM TEMPERATURE  //
            {
                    low=1;

            }
            else if(avg[j]>hightemp[j])      //  COMPARING WITH THE DESIRED
MAXIMUM TEMPERATURE //
            {
              high=1;
            }
            else
            {
              low=0;
              high=0;
            }

            switch(j)
            {
              case 0: if(low==1)
                            cmdoutA[0]=1;
                            else if(high==1)
                                  cmdoutA[1]=1;
                            else
                            {
                                  cmdoutA[0]=0;
                                  cmdoutA[1]=0;
                            }
                            break;
                case 1: if(low==1)
                            cmdoutA[2]=1;
                            else if(high==1)
                                  cmdoutA[3]=1;
                            else
```

```
                        {
                                cmdoutA[2]=0;
                                cmdoutA[3]=0;
                        }
                        break;
        case 2: if(low==1)
                                cmdoutA[4]=1;
                        else if(high==1)
                                cmdoutA[5]=1;
                        else
                        {
                                cmdoutA[4]=0;
                                cmdoutA[5]=0;
                        }
                        break;
        case 3: if(low==1)
                                cmdoutB[0]=1;
                        else if(high==1)
                                cmdoutB[1]=1;
                        else
                        {
                                cmdoutB[0]=0;
                                cmdoutB[1]=0;
                        }
                        break;
        case 4: if(low==1)
                                cmdoutB[2]=1;
                        else if(high==1)
                                cmdoutB[3]=1;
                        else
                        {
                                cmdoutB[2]=0;
                                cmdoutB[3]=0;
                        }
                        break;
        case 5: if(low==1)
                                cmdoutB[4]=1;
                        else if(high==1)
                                cmdoutB[5]=1;
                        else
                        {
                                cmdoutB[4]=0;
                                cmdoutB[5]=0;
                        }
                        break;
        case 6: if(low==1)
```

```
                        cmdoutC[0]=1;
            else if(high==1)
                        cmdoutC[1]=1;
            else
            {
                        cmdoutC[0]=0;
                        cmdoutC[1]=0;
            }
            break;
case 7: if(low==1)
                        cmdoutC[2]=1;
            else if(high==1)
                        cmdoutC[3]=1;
            else
            {
                        cmdoutC[2]=0;
                        cmdoutC[3]=0;
            }
            break;
case 8: if(low==1)
                        cmdoutA1[0]=1;
            else if(high==1)
                        cmdoutA1[1]=1;
            else
            {
                        cmdoutA1[0]=0;
                        cmdoutA1[1]=0;
            }
            break;
case 9: if(low==1)
                        cmdoutA1[2]=1;
            else if(high==1)
                        cmdoutA1[3]=1;
            else
            {
                        cmdoutA1[2]=0;
                        cmdoutA1[3]=0;
            }
            break;
case 10: if(low==1)
                        cmdoutA1[4]=1;
            else if(high==1)
                        cmdoutA1[5]=1;
            else
            {
                        cmdoutA1[4]=0;
```

```
                    cmdoutA1[5]=0;
            }
            break;
case 11: if(low==1)
                    cmdoutB1[0]=1;
            else if(high==1)
                    cmdoutB1[1]=1;
            else
            {
                    cmdoutB1[0]=0;
                    cmdoutB1[1]=0;
            }
            break;
case 12: if(low==1)
                    cmdoutB1[2]=1;
            else if(high==1)
                    cmdoutB1[3]=1;
            else
            {
                    cmdoutB1[2]=0;
                    cmdoutB1[3]=0;
            }
            break;
case 13: if(low==1)
                    cmdoutB1[4]=1;
            else if(high==1)
                    cmdoutB1[5]=1;
            else
            {
                    cmdoutB1[4]=0;
                    cmdoutB1[5]=0;
            }
            break;
case 14: if(low==1)
                    cmdoutC1[0]=1;
            else if(high==1)
                    cmdoutC1[1]=1;
            else
            {
                    cmdoutC1[0]=0;
                    cmdoutC1[1]=0;
            }
            break;
case 15: if(low==1)
                    cmdoutC1[2]=1;
            else if(high==1)
```

```
                                        cmdoutC1[3]=1;
                        else
                        {
                                cmdoutC1[2]=0;
                                cmdoutC1[3]=0;
                        }
                        break;
                case 16: if(low==1)
                                cmdoutA[6]=1;
                        else if(high==1)
                                cmdoutA[7]=1;
                        else
                        {
                                cmdoutA[6]=0;
                                cmdoutA[7]=0;
                        }
                        break;
                case 17: if(low==1)
                                cmdoutB[6]=1;
                        else if(high==1)
                                cmdoutB[7]=1;
                        else
                        {
                                cmdoutB[6]=0;
                                cmdoutB[7]=0;
                        }
                        break;

        }

        tempavg[y][j]=avg[j];

        low=0;
        high=0;
        }

        for(j=0;j<8;j++)          //  CALCULATING THE OUTPUT VALUE
FOR EACH PORT //
        {
                A=A+(cmdoutA[j]*(int)(pow(2,j)));
                B=B+(cmdoutB[j]*(int)(pow(2,j)));
                C=C+(cmdoutC[j]*(int)(pow(2,j)));
                A1=A1+(cmdoutA1[j]*(int)(pow(2,j)));
                B1=B1+(cmdoutB1[j]*(int)(pow(2,j)));
                C1=C1+(cmdoutC1[j]*(int)(pow(2,j)));
        }
```

```
            portnum=FIRSTPORTA;   //  EXECUTING THE OUTPUT
COMMANDS //
            cbDOut(2,portnum,A);
            cbDOut(1,portnum,A1);
            portnum=FIRSTPORTB;
            cbDOut(2,portnum,B);
            cbDOut(1,portnum,B1);
            portnum=FIRSTPORTCL;
            cbDOut(2,portnum,C);
            cbDOut(1,portnum,C1);


                        //  RE-INTIALIZATION OF VARIABLES //

            y++;
            x=0;
            A=0;B=0;C=0;A1=0;B1=0;C1=0;
            for(j=0;j<8;j++)
            {
                    cmdoutA[j]=0;
                    cmdoutB[j]=0;
                    cmdoutC[j]=0;
                    cmdoutA1[j]=0;
                    cmdoutB1[j]=0;
                    cmdoutC1[j]=0;
            }
            for(j=0;j<18;j++)
            {
                    sum[j]=0.0;
                    avg[j]=0.0;
            }
            time(&preti);
            }

            fp=fopen("result1.txt","w");      // ENTERING THE DATA INTO THE
OUTPUT FILE //
            if(fp==NULL)
            {
                    printf("\nFile pointer became NULL");
            }
            else
            {
                    for(m=0;m<y;m++)
                    {
                            for(j=0;j<18;j++)
                            {
```

```
                                    tempvalue=tempavg[m][j];
                                    fprintf(fp,"%f\n",tempvalue);
                            }
                    }
                    fclose(fp);
            }
            if(kbhit()!=0)
            {
                    portnum=FIRSTPORTA;
                    cbDOut(2,portnum,0);
                    cbDOut(1,portnum,0);
                    portnum=FIRSTPORTB;
                    cbDOut(2,portnum,0);
                    cbDOut(1,portnum,0);
                    portnum=FIRSTPORTCL;
                    cbDOut(2,portnum,0);
                    cbDOut(1,portnum,0);
                    cond=0;
            }
        }
}
```

Sine Code

```
include<stdio.h>
#include<conio.h>
#include "c:\mcc\c\cbw.h"        //  INCLUSION OF HEADER FILES //
#include<time.h>
#include<dos.h>
#include<math.h>

void main()                      //  VARIABLE DECLARATIONS //
{
        float tempvalue, temperature[18][1000], sum[18], avg[18], tempavg[5000][18];
float vartemp[18],reqtemp[18],ptime,rtemp,ptimearr[5000];
float slope[18]={(float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000,
                (float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000,

(float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000,

(float)1.0000,(float)1.0000,(float)1.0000,(float)1.0000,
                                (float)1.0000,(float)1.0000};



float intercept[18]={(float)0.0000,(float)0.0000,(float)0.0000,(float)0.0000,
                (float)0.0000,(float)0.0000,(float)0.0000,(float)0.0000,
```

```
                (float)0.0000,(float)0.0000,(float)1.5000,(float)1.2000,

(float)1.0071,(float)2.2000,(float)1.0000,(float)1.3000,
                (float)4.0000,(float)4.3000};
int portnum,cond;
int direction;
int x=0,j,y=0,m,low=0,high=0;
unsigned short A=0,B=0,C=0,A1=0,B1=0,C1=0;
int cmdoutA[8],cmdoutB[8],cmdoutC[8],cmdoutA1[8],cmdoutB1[8],cmdoutC1[8];
double i;
FILE *fp, *dbgfp;
time_t prevti,preti,prevti1,preti1,aclock;
struct tm *newtime;

direction = DIGITALOUT;        //  OUTPUT PORTS INTIALIZATION //
portnum = FIRSTPORTA;
cbDConfigPort(2,portnum, direction);
cbDConfigPort(1,portnum, direction);
portnum = FIRSTPORTB;
cbDConfigPort(2,portnum, direction);
cbDConfigPort(1,portnum, direction);
portnum = FIRSTPORTCL;
cbDConfigPort(2,portnum, direction);
cbDConfigPort(1,portnum, direction);
portnum = FIRSTPORTCH;
cbDConfigPort(2,portnum, direction);
cbDConfigPort(1,portnum, direction);

for(j=0;j<18;j++)        // INTIALIZATION OF VARIABLES //
{
        sum[j]=0.0;
        avg[j]=0.0;
}
for(j=0;j<8;j++)
{
 cmdoutA[j]=0;
 cmdoutB[j]=0;
 cmdoutC[j]=0;
 cmdoutA1[j]=0;
 cmdoutB1[j]=0;
 cmdoutC1[j]=0;
}

for(j=0;j<18;j++)        // READING IN THE DESIRED MAXIMUM AND MINIMUM
TEMPERATURES //
                         //  FOR EACH TANK  //
```

```c
{
        printf("Enter the average temperature for tank %d  : ", j);
        fflush(stdin);
        scanf("%f",&reqtemp[j]);
        printf("\n");
        printf("Enter the temperature variation that is allowed ");
        fflush(stdin);
        scanf("%f",&vartemp[j]);
}

// Opening debug file
  dbgfp = fopen("Errors.txt","w");
cond=1;
while(cond==1)
{
        time(&preti);
        time(&prevti);
        while(preti<prevti+1800)
        {
                time(&prevti1);
                time(&preti1);
                while(preti1<prevti1+180)
                {
                        for(j=0;j<18;j++)
                        {
                                cbTIn(2,16+j,0,&tempvalue,0);        // READING IN THE
TEMPERATURE VALUES //
                                temperature[j][x]=(tempvalue*slope[j])+intercept[j];
                        }

                        x++;
                        for(i=0;i<10000000;i++)      //  TO INDUCE SOME DELAY //
                                { }
                        time(&preti1);
                }

                printf("\t\nTank #\t    Time\t  Required temperature\t Average
temperature");
                printf("\n");

        printf("_____
_____");

                for(j=0;j<18;j++)
                {
```

```c
            for(m=0;m<x;m++)
            {
                    sum[j]=sum[j]+temperature[j][m];        // CALCULATING THE
SUM AND AVERAGE TEMPERATURES //
            }
      // FOR EACH FIVE MINUTE INTERVAL //
            avg[j]=sum[j]/(float)(x);

            time(&aclock);
            newtime=localtime(&aclock);
            ptime=((float)(newtime->tm_hour))+((float)((newtime->tm_min))/60);
            ptimearr[y]=ptime;
            fprintf(dbgfp,"\nCalculated: %f",ptime);
            rtemp=reqtemp[j]+(vartemp[j])*(float)sin((12-(ptime-6))*2*3.1428/24);

            tempvalue=avg[j];
            printf("\n  %d\t %f\t\t%f\t\t%f",j+1,ptime,rtemp,tempvalue);

            if(avg[j]<rtemp)            // COMPARING WITH THE DESIRED
MINIMUM TEMPERATURE  //
            {
            low=1;
            }
            else if(avg[j]>rtemp)      //  COMPARING WITH THE DESIRED
MAXIMUM TEMPERATURE //
            {
            high=1;
            }
            else
            {
            low=0;
            high=0;
            }

            switch(j)
            {
              case 0: if(low==1)
                            cmdoutA[0]=1;
                      else if(high==1)
                            cmdoutA[1]=1;
                      else
                      {
                            cmdoutA[0]=0;
                            cmdoutA[1]=0;
                      }
                      break;
```

```
   case 1: if(low==1)
                   cmdoutA[2]=1;
           else if(high==1)
                   cmdoutA[3]=1;
           else
           {
                   cmdoutA[2]=0;
                   cmdoutA[3]=0;
           }
           break;
case 2: if(low==1)
                   cmdoutA[4]=1;
           else if(high==1)
                   cmdoutA[5]=1;
           else
           {
                   cmdoutA[4]=0;
                   cmdoutA[5]=0;
           }
           break;
case 3: if(low==1)
                   cmdoutB[0]=1;
           else if(high==1)
                   cmdoutB[1]=1;
           else
           {
                   cmdoutB[0]=0;
                   cmdoutB[1]=0;
           }
           break;
case 4: if(low==1)
                   cmdoutB[2]=1;
           else if(high==1)
                   cmdoutB[3]=1;
           else
           {
                   cmdoutB[2]=0;
                   cmdoutB[3]=0;
           }
           break;
case 5: if(low==1)
                   cmdoutB[4]=1;
           else if(high==1)
                   cmdoutB[5]=1;
           else
           {
```

```
                        cmdoutB[4]=0;
                        cmdoutB[5]=0;
                }
                break;
        case 6: if(low==1)
                        cmdoutC[0]=1;
                else if(high==1)
                        cmdoutC[1]=1;
                else
                {
                        cmdoutC[0]=0;
                        cmdoutC[1]=0;
                }
                break;
        case 7: if(low==1)
                        cmdoutC[2]=1;
                else if(high==1)
                        cmdoutC[3]=1;
                else
                {
                        cmdoutC[2]=0;
                        cmdoutC[3]=0;
                }
                break;
        case 8: if(low==1)
                        cmdoutA1[0]=1;
                else if(high==1)
                        cmdoutA1[1]=1;
                else
                {
                        cmdoutA1[0]=0;
                        cmdoutA1[1]=0;
                }
                break;
case 9: if(low==1)
                        cmdoutA1[2]=1;
                else if(high==1)
                        cmdoutA1[3]=1;
                else
                {
                        cmdoutA1[2]=0;
                        cmdoutA1[3]=0;
                }
                break;
        case 10: if(low==1)
                        cmdoutA1[4]=1;
```

```
                else if(high==1)
                        cmdoutA1[5]=1;
                else
                {
                        cmdoutA1[4]=0;
                        cmdoutA1[5]=0;
                }
                break;
case 11: if(low==1)
                        cmdoutB1[0]=1;
                else if(high==1)
                        cmdoutB1[1]=1;
                else
                {
                        cmdoutB1[0]=0;
                        cmdoutB1[1]=0;
                }
                break;
case 12: if(low==1)
                        cmdoutB1[2]=1;
                else if(high==1)
                        cmdoutB1[3]=1;
                else
                {
                        cmdoutB1[2]=0;
                        cmdoutB1[3]=0;
                }
                break;
case 13: if(low==1)
                        cmdoutB1[4]=1;
                else if(high==1)
                        cmdoutB1[5]=1;
                else
                {
                        cmdoutB1[4]=0;
                        cmdoutB1[5]=0;
                }
                break;
case 14: if(low==1)
                        cmdoutC1[0]=1;
                else if(high==1)
                        cmdoutC1[1]=1;
                else
                {
                        cmdoutC1[0]=0;
                        cmdoutC1[1]=0;
```

```
                    }
                    break;
            case 15: if(low==1)
                            cmdoutC1[2]=1;
                    else if(high==1)
                            cmdoutC1[3]=1;
                    else
                    {
                            cmdoutC1[2]=0;
                            cmdoutC1[3]=0;
                    }
                    break;
    case 16: if(low==1)
                            cmdoutA[6]=1;
                    else if(high==1)
                            cmdoutA[7]=1;
                    else
                    {
                            cmdoutA[6]=0;
                            cmdoutA[7]=0;
                    }
                    break;
    case 17: if(low==1)
                            cmdoutB1[6]=1;
                    else if(high==1)
                            cmdoutB1[7]=1;
                    else
                    {
                            cmdoutB1[6]=0;
                            cmdoutB1[7]=0;
                    }
                    break;
            }

            tempavg[y][j]=avg[j];
            low=0;
            high=0;

    }//End of top for

    for(j=0;j<8;j++)          //  CALCULATING THE OUTPUT VALUE FOR
EACH PORT //
    {
            A=A+(cmdoutA[j]*(int)(pow(2,j)));
            B=B+(cmdoutB[j]*(int)(pow(2,j)));
            C=C+(cmdoutC[j]*(int)(pow(2,j)));
```

110

```
            A1=A1+(cmdoutA1[j]*(int)(pow(2,j)));
            B1=B1+(cmdoutB1[j]*(int)(pow(2,j)));
            C1=C1+(cmdoutC1[j]*(int)(pow(2,j)));
        }

        portnum=FIRSTPORTA;    //  EXECUTING THE OUTPUT COMMANDS //
        cbDOut(2,portnum,A);
        cbDOut(1,portnum,A1);
        portnum=FIRSTPORTB;
        cbDOut(2,portnum,B);
        cbDOut(1,portnum,B1);
        portnum=FIRSTPORTCL;
        cbDOut(2,portnum,C);
        cbDOut(1,portnum,C1);


                    //  RE-INTIALIZATION OF VARIABLES //


        y++;
        x=0;
        A=0;B=0;C=0;A1=0;B1=0;C1=0;
        for (j=0;j<8;j++)
        {
            cmdoutA[j]=0;
            cmdoutB[j]=0;
            cmdoutC[j]=0;
            cmdoutA1[j]=0;
            cmdoutB1[j]=0;
            cmdoutC1[j]=0;
        }

        for(j=0;j<18;j++)
        {
        sum[j]=0.0;
        avg[j]=0.0;
        }
        time(&preti);
    }

  fp=fopen("result1.txt","w");       // ENTERING THE DATA INTO THE OUTPUT
FILE //
  if(fp==NULL)
  {
        //Indicating file pointer error -- writing to error.txt
        fprintf(dbgfp,"\nFile Pointer became NULL.");
```

```c
          }
    else
    {
            //Writing complete set of readings

            for(m=0;m<y;m++)
            {
                    fprintf(fp,"%6.4f\t", ptimearr[m]);
                    for(j=0;j<18;j++)
                    {
                                    tempvalue=tempavg[m][j];
                //        fwrite(&tempvalue,sizeof(float),1,fp);
                                    fprintf(fp,"%6.4f\t", tempvalue);
                    }
                    fprintf(fp, "\n");
            }
            fclose(fp);
            cond = 1;
    }
  /* if(kbhit()!=0)
   {
            fclose(fp);
            fclose(dbgfp);
      fprintf(dbgfp,"\nKey hit...");
            portnum=FIRSTPORTA;
      cbDOut(2,portnum,0);
      cbDOut(1,portnum,0);
      portnum=FIRSTPORTB;
      cbDOut(2,portnum,0);
      cbDOut(1,portnum,0);
      portnum=FIRSTPORTCL;
      cbDOut(2,portnum,0);
      cbDOut(1,portnum,0);
      cond=0;
   }*/
   fprintf(dbgfp,"\nNext Loop (infinite)");
 }
 fprintf(dbgfp,"\nExited Loop");
 }
```

Realwire

```python
#! /usr/bin/env python
import time
t = time.localtime()
fname = '%d%02d%02d' % (t[0], t[1], t[2])
```

```
f = file('c:/wget/temp.txt','r')
str = f.read()
f.close()
lst = str.split('<br>')
f = file('c:/wget/%s.dat' % fname,'a')
f.write('%s\n' % lst[1])
f.close()
```
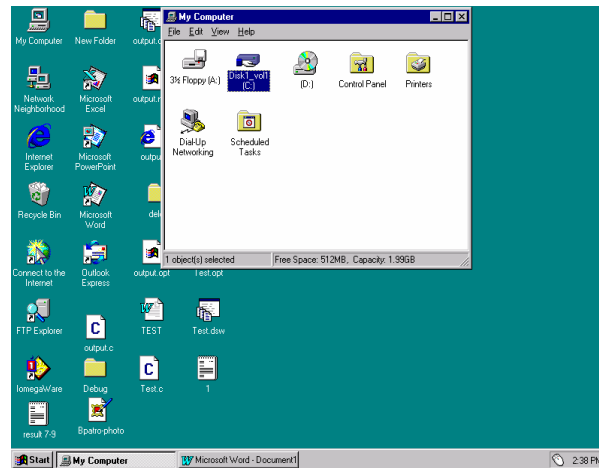
Appendix C

Standard Operating Procedures

1. To use the system for maintaining constant water temperatures in tanks

The following steps will illustrate how to program the system to maintain constant temperatures

- Start the computer
- Select and Double click C Drive as shown below

- Select the folder "**Desktop"** and double click it.
- Desktop opens folders containing programs needed for this system (See figure below).


- Select folder called **Deadband** and double click it. When the folder opens, double click on **deadband.dsw** icon and program opens on the screen

- Go to pull down menu called "**Build"** (on the top right of the screen).
- Click **Rebuild All.** The program is compiled and returns "0 errors, 0 warnings".
- In the same menu (Build) click on "**Execute"**.
- A black screen appears as shown below asking for maximum and minimum temperatures desired in each tank.
- Enter the values and after last value is entered, the program starts.
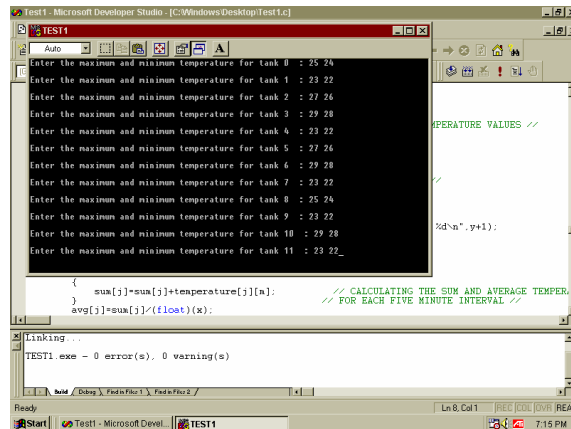
2. To use the system for maintaining constant water temperatures in tanks

- Open the **Desktop** folder as explained above.
- Select the folder called "Output" and double click it.
- The folder opens (see below)
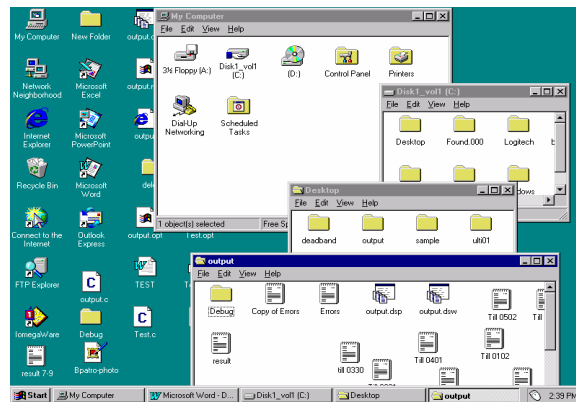
- Select and open file **output.dsw**

- Click **Rebuild all on** the **Build** menu and program returns **0 errors and 0 warnings** and as shown below.
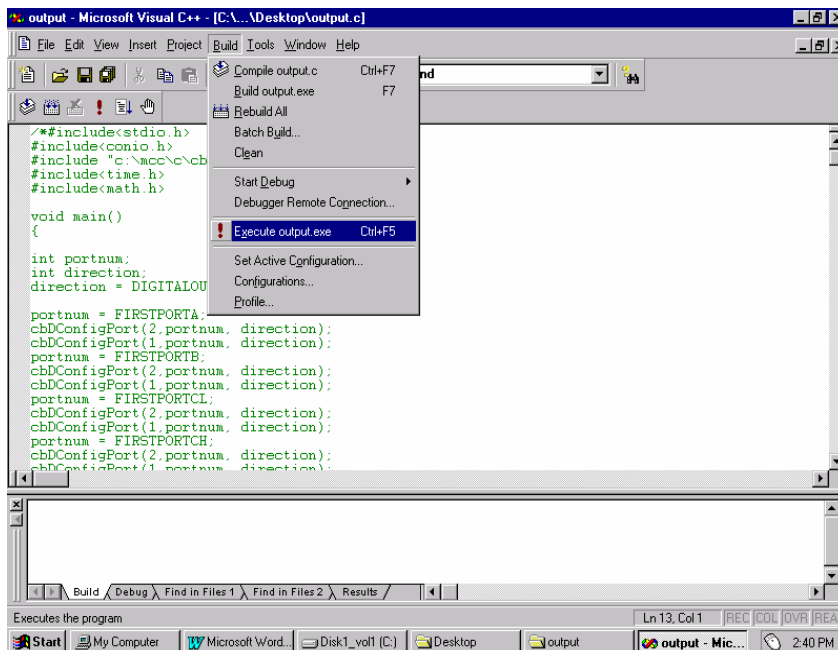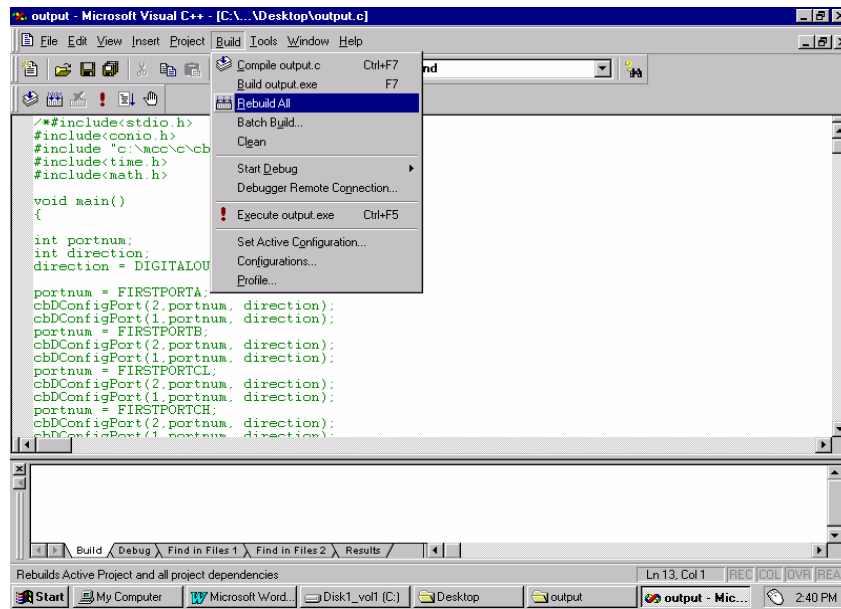
- Execute as shown below.

- A blank screen appears asking for average temperatures and variations in temperatures. For example if a diurnal simulation of 15 ± 5°C is desired, then average is 15°C and variation is 5°C. The water temperature would be around 20°C at noon and 10°C at midnight. Enter the temperatures as desired (see next page)

116
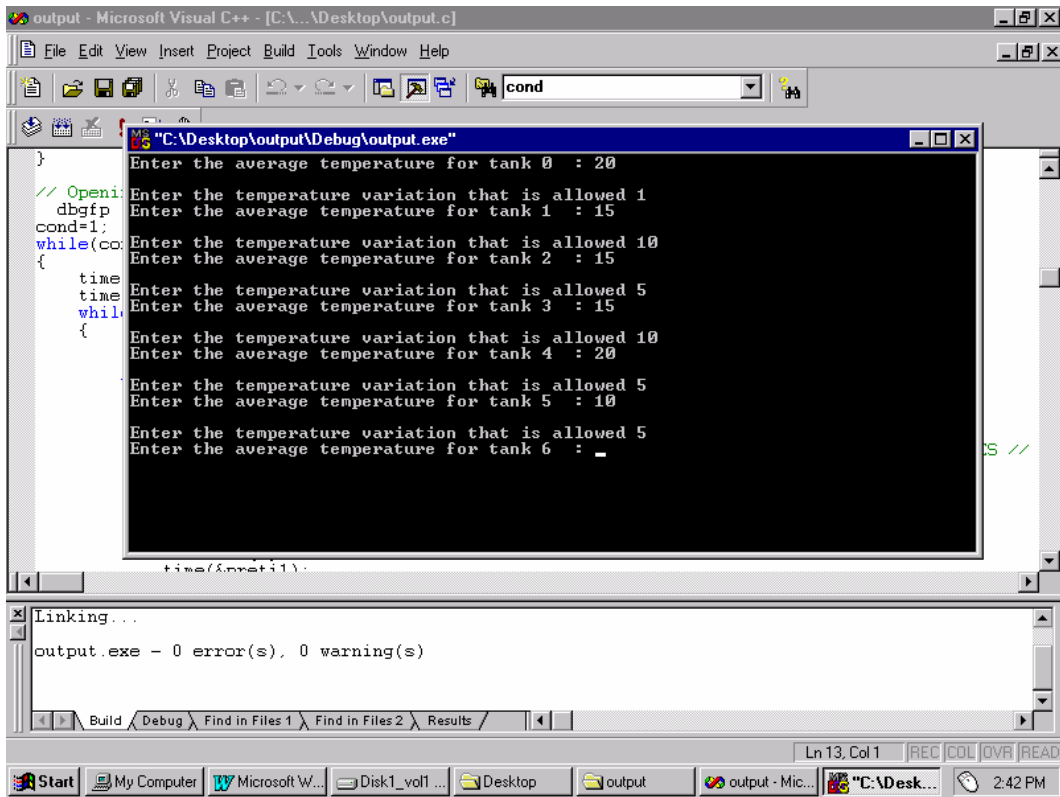
Vita

Praveen Kolar was born on July 1, 1970 in Hyderabad, India. He attended Sri Venkateswara University at Thirupati, India, where he received his Bachelor of Technology with honors in civil engineering in 1991. There after he worked for Sri Vishnu Cements Limited, Hyderabad, India, as quality control engineer till 1993. In Feb1993 he scored 99.8 percentile in the Graduate Aptitude Test in Engineering (GATE), which secured him an admission into a master's program in aquacultural engineering at the Indian Institute of Technology (IIT), Kharagpur. Upon completion of his masters program in January 1995, Praveen joined Navayuga Exports Limited, Bhubaneswar, India, as Technical Manager. For more than seven years, Praveen was involved in the Navayuga's organizational affairs, from purchasing raw materials, quality control, administration, to processing seafood for Japanese and American markets. In fall 2002 he and his wife Chandrika, moved to Baton Rouge, Louisiana, for a combined master's and doctoral degrees program in biological and agricultural engineering at Louisiana State University (LSU), Baton Rouge. While a graduate student at LSU, Praveen worked with Dr. Steven G. Hall as a graduate assistant, during which he attended various national and international conferences. He was also awarded the third best abstract at the 25th annual meeting of the American Fisheries Society (AFS) at Baton Rouge. His research mainly focused on developing process control systems for studying aquatic species. Praveen is currently a candidate for the degree of Master of Science in Biological and Agricultural Engineering on August 5, 2004.