4-15-2015 12:00 AM

Configuration, Programming, Implementation, and Evaluation of Distributed Control System for a Process Simulator

Ximing Liu, *The University of Western Ontario*

Supervisor: Jin Jiang, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

CONFIGURATION, PROGRAMMING, IMPLEMENTATION AND EVALUATION OF DISTRIBUTED CONTROL SYSTEM FOR A PROCESS SIMULATOR

(Thesis format: Monograph)

by

Ximing Liu

Graduate Program in
Electrical and Computer Engineering

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Engineering Science

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

A common industrial distributed control system (DCS), DeltaV, is configured and programmed to control and monitor the Nuclear Process Control Test Facility (NPCTF). A cabinet which holds the hardware of the DelatV DCS system, including programmable logic controller (PLC), power supplies, input/output (I/O) cards, terminals, and relays are configured and wired to field devices of NPCTF. A workstation and HMI screen are configured and setup.

To implement the main functions of NPCTF in the DelatV system, the programming architecture is designed in the DelatV system. The main control and monitoring functions of NPCTF are programmed using industrial languages of Function Block Diagram (FBD) and Sequential Function Chart (SFC) by IEC61113-3. Safety interlocks are added in the program to protect the NPCTF devices from damage. A HMI is developed to operate and monitor the NPCTF. Through the HMI, the operator can monitor the parameters of process of NPCTF, operate the NPCTF, change parameters of the controller, and force the devices.

The process model of SG (Steam Generator) Tank level control is developed using the MATLAB System Identification tool. The model is taken as an example to demonstrate the process of analysis and design the controller of process control. PID is used as the controller algorithm.

The main control and monitoring functions of NPCTF in the DeltaV system are commissioned, tested and evaluated. The evaluation results conclude that the DelatV DCS system can control the NPCTF to achieve the main functions of the NPCTF.

Keywords: Distributed Control System (DCS), process control, Programmable Logic Controller (PLC), Proportional-Integral-Derivative (PID) Control

Acknowledgments

I would like to express my appreciation and thanks to my advisor Dr. Jin Jiang, whose expertise, understanding and patience added considerably to my graduate experience. I appreciate his thorough guidance through my research and graduate life. I must thank the members of the examining board for taking time from their busy schedules to provide valuable feedback.

I would also like to thank Dr. Xinhong Huang for her encouragement, patience and assistance in writing this thesis and Dr. Ataul Bari for his support in this research, and for ordering the devices for it.

I am also thankful to all the members of the Control Instrumentation and Electrical Systems research group of the University of Western Ontario, in particular Drew Rankin and Jianping Ma, for their support in various aspects of my research work. Our debates, and exchanges of knowledge and skills throughout my graduate program have enriched my graduate experience.

I give special thanks to my wife. Words cannot express how grateful I am to her for all her sacrifices. Without her support, encouragement, and love I would not have returned to academia and completed this thesis.

I also acknowledge that this research would not have been possible without the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the University Network of Excellence in Nuclear Engineering (UNENE), the University of Western Ontario Graduate Studies, the Department of Electrical and Computer Engineering at the University of Western Ontario (Teaching Assistantships, Graduate Research Scholarships). I express my gratitude to those organizations.

Table of Contents

List of Tables

List of Figures

Abbreviations and Nomenclature

Abbreviations

2oo3	Two-out-of-three voting
A/D	Analog to Digital
AECL	Atomic Energy of Canada Limited
AI	Analog Input
AO	Analog Output
CANDU	Canadian Deuterium Uranium
CB	Circuit Breaker
CPU	Central Processing Unit
CV	Control Valve
D/A	Digital to Analog
DCS	Distributed Control System
DDC	Direct-digital Control
DI	Digital Input
DO	Digital Output
D_2O	Deuterium Oxide, Heavy Water
ECC	Emergency Core Cooling

FBD	Function Block Diagram
FOPDT	First-order-plus-dead-time
FPE	Final Prediction Error
HMI	Human Machine Interface
I&C	Instrumentation and Control
ICI	Imperial Chemical Industries
IL	Instruction List
IMC	Internal Model Control
I/O	Input / Output
LD	Ladder Diagram
LOCA	Loss-of-coolant Accident
mA	milli-ampere
MSE	Mean Square Error
MWe	Megawatt Electrical
NC	Normal Close
NO	Normal Open
NPCTF	Nuclear Power Control Test Facility
NPP	Nuclear Power Plant

OLE	Object Linking and Embedding
OPC	OLE for Process Control
PEM	Prediction Error Method
PID	Proportional-integral-derivative
PLC	Programmable Logic Controller
P&ID	Piping and Instrumentation Diagram
PV	Process variable
SDS1	Shutdown System Number 1
SDS2	Shutdown System Number 2
SFC	Sequential Function Chart
SG	Steam Generator
SIMC	Simple Internal Model Control
SOPDT	Second-order-plus-dead-time
SP	Set-point
ST	Structured Text
VDC	Voltage Direct Current

Nomenclature

K_c	controller gain
τ_I	integral time
T_s	scan interval
τ_D	derivative time
K_p	process gain
τ_p	time constant
θ	process delay
ω_c	critical frequency
ω_g	gain cross frequency

Chapter 1

Introduction

1.1 Background

1.1.1 CANDU Nuclear Power Plant (NPP)

The CANDU (Canadian Deuterium Uranium) is a Canadian invented, pressurized heavy water reactor. It is characterized by heavy water as coolant and moderator, and use natural uranium as fuel [1]. CANDU NPPs (nuclear power plants) are developed and maintained by AECL (Atomic Energy of Canada Limited) [2] [3]. There are 22 units of CANDU reactors with a capacity of 15,358 MWe. It provides approximately 15% of Canada's total electricity needs (over 50% in Ontario, Canada) [4]. CANDU reactors have been installed in seven countries [5].

The main energy exchange process in the CANDU nuclear power plant (NPP) is illustrated in Figure 1.1. The heavy water (D_2O) of the heat transport system is pumped to the reactor to be heated and becomes heated heavy water. The heated heavy water is then transported to steam a generator. Inside the steam generator, the heated heavy water heats the feed-water to saturated steam and becomes cooled heavy water. The cooled heavy water feeds back to the reactor. The saturated steam is fed to drive the turbine and generator to produce power. Thereafter, steam becomes water and is pumped back to the steam generator.

The process control system of CANDU NPP mainly consists of reactor regular system, pressure and inventory control system, steam generator pressure and level control system, and turbine control system. The reactor regulator system adjusts the reactivity of the reactor and, thus, the power of the reactor. The pressure and inventory control system regulates the pressure and inventory of the coolant in the heat transport system. The steam generator pressure and level control system controls the pressure and level of the steam generator. The turbine control system controls the flow or pressure of the main steam via turbine governors.

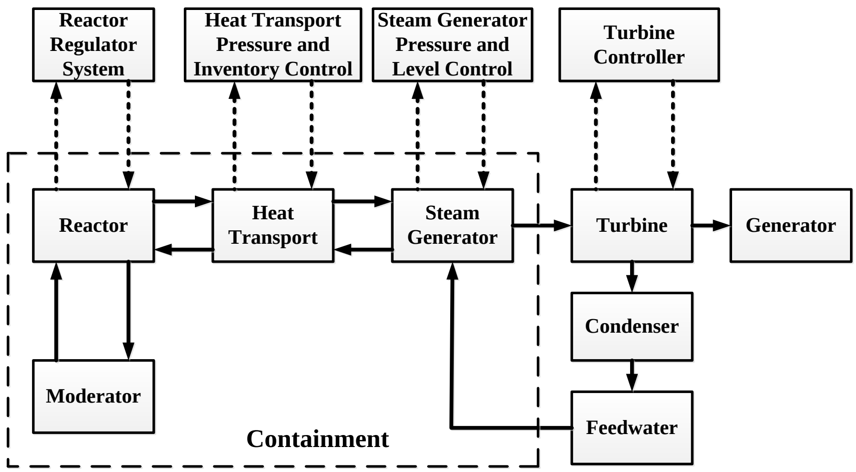

Figure 1.1 Simplified block diagram of main process and control systems of a CANDU NPP

The heat transport system has two operation modes: Normal mode and Solid mode. Under Normal operating mode, the pressurizer keeps the pressure of the heat transport system while the feed and bleed system controls the inventory through pressurizer level, as shown in Figure 1.2. Under the Solid operating mode, the pressurizer is isolated. Instead, the feed and bleed system controls both pressure and inventory, as shown in Figure 1.2.

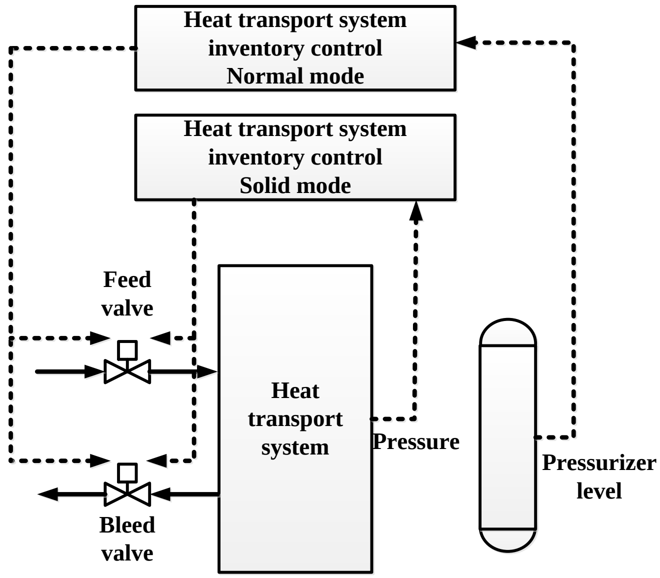

Figure 1.2 Heat transport system inventory control [6]

The reactor safety system performs the functions necessary to maintain the plant in safe conditions. SDS1 (Shutdown System Number 1), SDS2 (Shutdown System Number 2), and ECC (Emergency Core Cooling) are three main reactor safety systems. SDS1 and SDS2 are two shutdown systems that shut down the reactor for safety reasons. SDS1 consists of mechanical shutdown rods while SDS2 injects a gadolinium nitrate solution into the moderator. These two shutdown systems are independent, and either of them can shut down the reactor and maintain the shut down for all design basis events. Emergency Core Cooling system supplies coolant to the reactor in the event of a loss-of-coolant accident (LOCA).

1.1.2 Distributed Control System (DCS)

To study the dynamic properties of CANDU NPP, a physical component known as Nuclear Power Control Test Facility (NPCTF) has been constructed in the CIES lab of the University of Western Ontario. In this research, a DCS system is developed to implement its functions.

A Distributed control system is a computer based control system of an industry process where its elements are distributed through different places [7] [49].

In the 1950's, plants used large, local pneumatic controllers; the computers could only be used for supervisory tasks. Many companies started the research to determine if computers could be used for process control.

In the 1960s, the development of electronic integrated circuits led to the development of micro-computers [8]. In 1962, an Argus computer developed by Ferranti Ltd. replaced a complete analog control system in an ammonia/soda plant by a British chemistry company, Imperial Chemical Industries (ICI). It is a direct-digital control (DDC) computer and is expensive; however, it is still cheaper than the relay based control system [9].

The development of computer technology, which is faster, cheaper and more reliable, incubated the first distributed control system, TDC2000, and was introduced by Honeywell in 1975 [10] [11]. However, this DCS was not suitable for discrete and batch applications until the incorporating of PLC with DCS [12].

The further advances in computer hardware and software technology in 1980s caused the DCSs to expand its function beyond process control. It facilitated the production, such as process modeling, process accounting, expert system, and production planning and dispatch. Another feature of DCS during this period was that DCSs were shifted from proprietary systems to open systems to accommodate third party software. UNIX, one of the operating systems, was a popular choice for operating system of DCS [13].

In the 1990s, Microsoft entered the related software field. The operating system of DCS was moved from a UNIX to Windows environment [14]. OLE for process control (OPC) technology was developed for DCS systems. OPC was only available on MS windows and is now a de facto industry connectivity standard. It provides the "plug and play" solution to the problem of integrating a device to the DCS [15]. Microsoft application tools Visual Basic and Visual C++ coupled with Active tools made complex and highly specialized applications affordable for users.

Right now, DCS has been applied extensively in process control of different industries: chemical plants, petrochemical industries and refines, power plant systems, nuclear power plants, environmental control systems, water management systems, oil refining plants, metallurgical process plants, and pharmaceutical manufacturing [11].

A nuclear power plant is a complex system. The distribute control system can perform complex tasks and, thus, has been applied to process control in NPP [16] [17].

A typical DCS system structure includes devices in a control and supervisory level, as shown in Figure 1.3. The control level collects the data from devices in the field which are grouped with the name of the field level, does calculation and sends the results to control the field devices. It includes a controller, such as PLC, IO cards, and communication modules. The supervisory level includes operator station through human machine interface (HMI) and an engineer station. Operators operate and monitor the process through HMI. The engineer station configures the control strategies, maintains the system and tunes the controller. The field level includes different kinds of field devices, such as transmitters, sensors, and actuators. Transmitters convert the process variables to electric signals, such as 4-20mA, or digital signal which are sent to the controller. Valves actuators convert the electric from controller to mechanical movement to drive the valves.

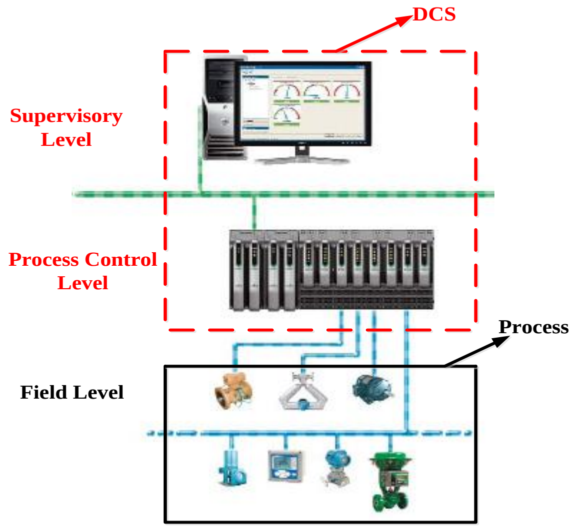

Figure 1.3 Typical DCS system structure [18]

1.1.3 Programmable logic controller (PLC)

In the DCS system designed in this research, PLCs (programmable logic controllers) are used as the control equipment.

Originally, the process was often controlled by relays where the control room was full of relays, terminals blocks and wire. For the relay-based control system, significant changes are required to expand the system and add more functions. Moreover, it is difficult to conduct troubleshooting considering there are huge number of relays and wiring. In the 1960’s, engineers at GM proposed to develop a prototype to eliminate costly scrapping of assembly-line relays during changeovers, and replace unreliable relays. In 1971, Allen Bradly developed a new concept, Bulletin 1774 PLC, and named it “Programmable Logic Controller” (PLC). From then on, the PLC terminology became the industrial standard. Several improvements have been made for PLC since it was first developed [19].

In the early 1980s, the industrial processes became more complex and a single PLC could not control the entire system. Therefore, PLC began to incorporate distributed control functions so they could be linked much in the way that DCSs were linked [20] [21].

During the 1990s, standardization and an open system enabled further improvement in PLC development.

In the market, there are different sizes of configurations: Micro PLC with less than 100 I/O, Small PLC with 100 to 200 I/O, Medium with less than 1000 I/O and large PLC with more than 1000 I/O.

PLC has been extensively applied in different industry areas. According to a recent Control Engineering magazine poll, “The major applications for PLCs include machine control (87%), process control (58%), motion control (40%), batch control (26%), diagnostic (18%), and others (3%).” The results do not add up to 100% because a single control system generally has multiple applications [19].

A typical PLC mainly consists of CPU (central processing unit), power supply, memory unit, and I/O (input/output) system, as illustrated in Figure 1.4. Power supply converts the power line voltages to those required by the PLC. It drives the central processing unit, memory unit, and I/O signals. CPU is the central component of PLC. It performs necessary tasks to fulfil PLC function, such as scanning, executing programs, communicating with other devices, and performing self-diagnostics. The memory unit stores the application program and PLC execution program. The control data from the CPU and the process data from I/O modules are also temporarily stored in the memory unit. I/O modules function as the interface between field devices and CPU. The input signal gives the CPU real time status of process variables. These signals could be either analog or digital. Water flow is an example of analog signal and contact of relay is an example of digital signal. They are presented to input cards as a varying voltage, or current. Discrete and digital are common output categories. Discrete output can be sent to turn on or turn off pilot lights, or open or close solenoid valves. Analog output can drive signals to variable speed drives or I/P (current to air) converters and thus to control valves. I/O systems are arranged in modules each of which contains a couple of I/O points. These modules are plugged into the existing bus structure which carries information back and forth between the I/O modules and central unit.

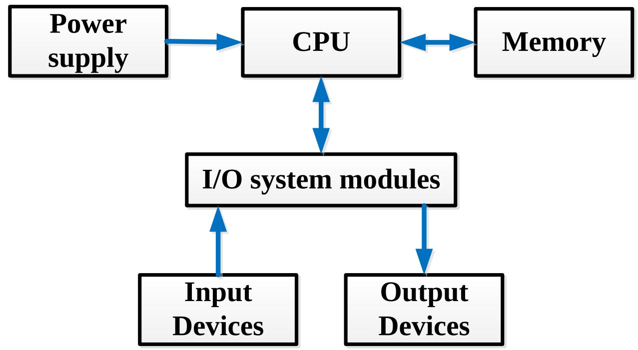

Figure 1.4 PLC architecture

1.2 Motivations

DCS is applied in many industries [22], including different types of power stations. Nuclear power plants are large, complex systems with potential risks [23]. An accident could cause serious loss and impact on society. Therefore, any new control strategy or function needs to be proved before applying it to a real nuclear power plant. In the past, software which simulates the process of an NPP is the common method to test and prove a new control strategy and function. However, process in the software is created by a model which is always different from real plants. A system between software and a real NPP will improve the validation process. Therefore, the NPCTF (Nuclear Process Control Test Facility) is built in the CIES research lab at the University of Western Ontario to simulate the main processes in a CANDU NPP and some other industrial processes for I&C research. Even though it is not a real NPP, it is a real physical system.

In addition, this DCS operated NPP simulator can be used to train operators and students. For students, it helps to create some concepts about CANDU NPP. For operators, it helps to improve operating skills and emergency handling capabilities.

This research project is to design the DCS to control and operate the NPCTF. The popular industrial DCS, Delta-V manufactured by Emerson, is used to design this system [24].

1.3 Objectives and Scope

As described in Section 1.2, it is important to develop a DCS system to implement the functions of NPCTF. The following are the main objectives of this research:

1. Design and implementation of a DCS system using a DeltaV platform to control the NPCTF.

2. Development of programs to implement the main functions of the NPCTF system

3. The process model of steam generator (SG) level control is developed and analyzed to present the process control strategies design process.

4. Development of the operator interface to operate the NPCTF system.

5. Evaluation of the main functions of a DCS system to demonstrate the functionalities of DeltaV DCS system to control NPCTF.

The goal of this work is to develop the DeltaV DCS system to control the NPCTF system, and demonstrate the functionalities of NPCTF. Therefore, the following issues are out of the scope of this research:

1. The specifications and the system design of NPCTF

1.4 Contributions

The following are the major contributions:

1. Control system development: A DCS system is developed to implement the process control, sequence control, designed functions of safety systems and operation mode selection on the NPCTF.

2. Programming: The control system of the NPCTF system is programmed in DeltaV DCS system. HMI (Human Machine Interface) is developed to operate and monitor this system.

3. Model development: A Steam Generator (SG) level process model within operating range of 36% to 44% is constructed using MATLAB and used to design the controller of SG level control.

4. Tuning technique: The tuning technique is compared in MATLAB and on NPCTF system.

5. Commissioning of the DCS: The control system of the NPCTF described in Chapter 2 are commissioned.

6. Evaluation of the DCS system: The control system described in chapter 2 are tested and evaluated.

1.5 Organization of the Thesis

This thesis is organized as follows.

Chapter 2 introduces and describes the main functions to be implemented and achieved in the NPCTF simulator through the DCS system. The functions are expressed in logic diagrams.

Chapter 3 describes the DeltaV DCS platform used in this thesis, and how the DeltaV DCS platform is used to program and implement the functions described in Chapter 2. The hardware design and construction of the DCS system using the DeltaV platform are discussed. The control architecture design of NPCTF is also presented. In addition, the main programming languages used in this research are introduced. Furthermore, how to use these two kinds of languages to program and implement the functions described in Chapter 2 is also presented. In the end of Chapter 3, the details of developing HMI for monitoring and operating the NPCTF are also discussed.

Since the process control is one of the main functions of the DCS system, Chapter 4 focuses on the development of process control. The control algorithm of PID, tuning techniques and stability analysis standard used in the process control of this research are presented. Steam Generator (SG) level control is taken as an example to explain the control process. In this example, the process model of Steam Generator level control is created and two tuning techniques are compared in both MATLAB and NPCTF system.

Chapter 5 presents the results of the experiments and evaluations performed to demonstrate the main functions of the DCS system for the NPCTF.

Chapter 6 concludes the research which summarizes the main contributions and discusses potential future research.

Chapter 2

Control System Requirements for Nuclear Process Control Test Facility (NPCTF)

The simulator NPCTF (Nuclear Process Control Test Facility) has been built in the CIES research lab at the University of Western Ontario. There are mainly four functions of this simulator to be implemented in this research: process control, sequence control, designed functions of safety systems, and operation modes. Process control is to control the process variables of NPCTF. The main purpose of Sequence control is to fill in the pipes and tanks in order for the system to be ready for operation. Safety system is used to investigate the safety shut down functions: SDS1, SDS2, and ECC. Operation mode is to make the NPCTF system run under different situations: Normal mode, Open mode and Solid mode. These functions are programed and evaluated in this research.

2.1 System Introduction

The physical system of NPCTF is shown in Figure 2.1. As shown in Figure 2.1, Heater, Pressurizer, HX tank, Turbine, and Generator are used to represent reactor, pressurizer, steam generator, turbine, and generator in CANDU NPP, respectively. Its P&ID diagram is shown in Figure 2.2. Before the system is to be controlled to desired status, the tanks and pipes need to be filled up. As shown in Figure 2.1 and 2.2, Pump 2 feeds water through valve CV-16 to fill in the pipes and tanks to make the system ready for control. After the system is ready, an operator can select different operating modes to run the system. In the sections 2.5, detail of different operating modes and their related operations will be further described.

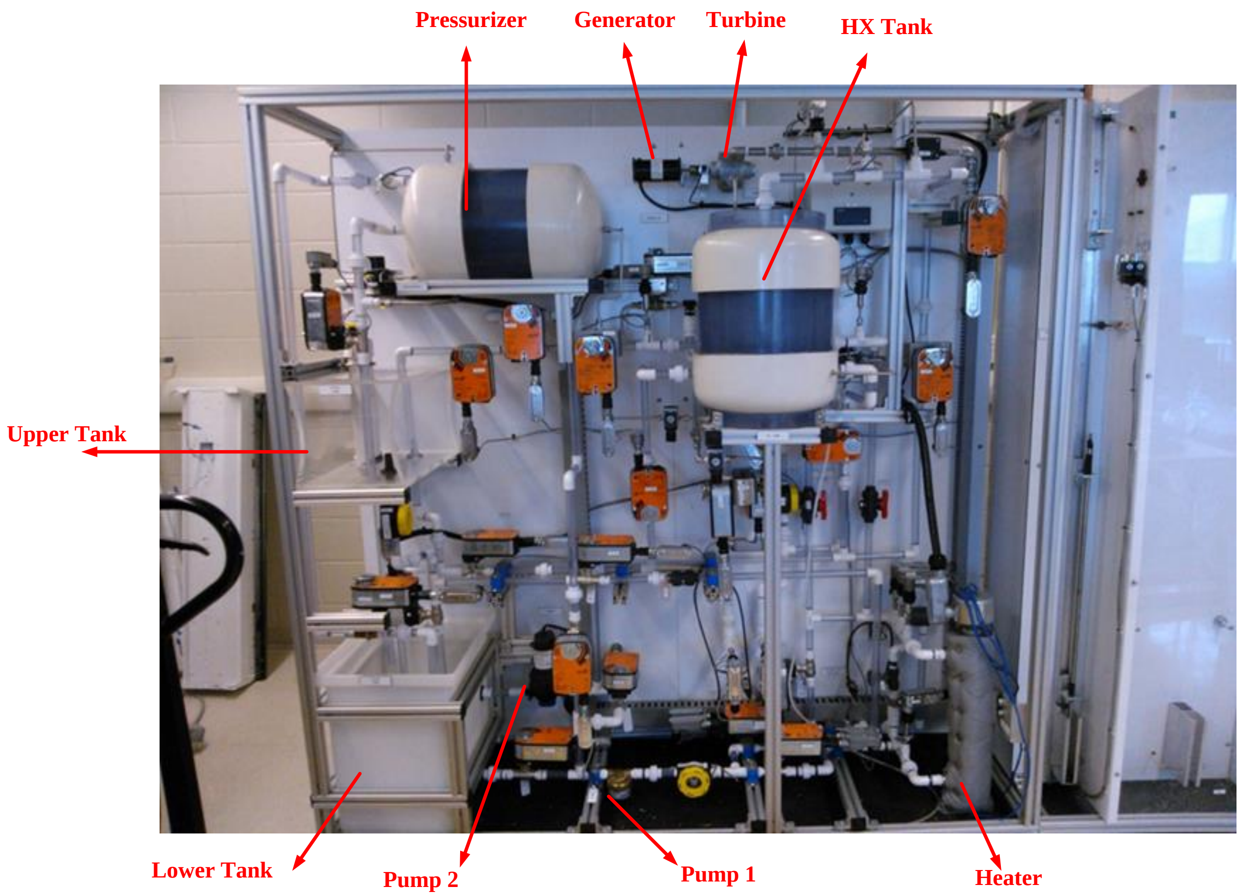

Figure 2.1 Physical system of NPCTF

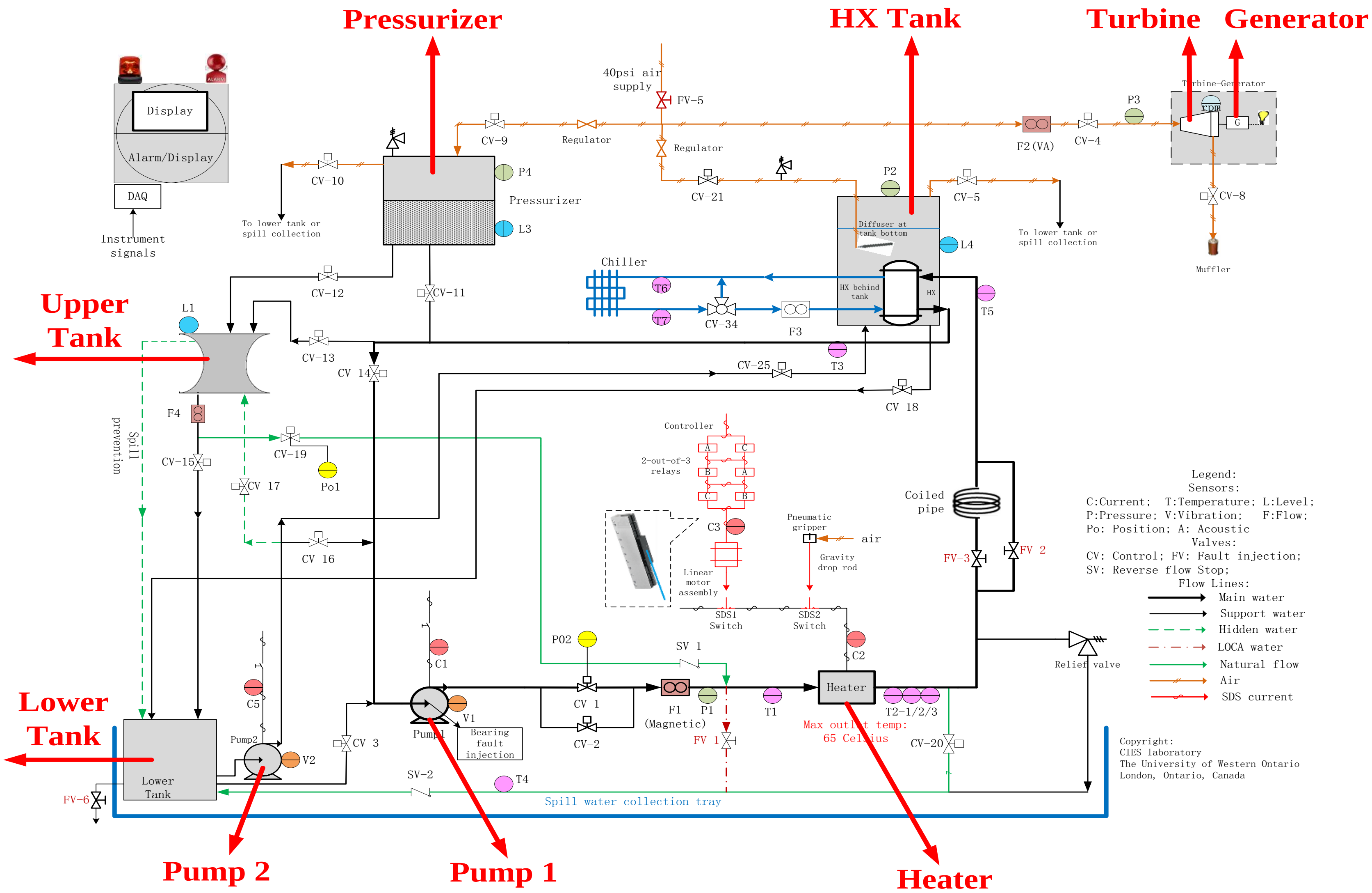

Figure 2.2 P&ID Diagram for NPCTF [25]

2.2 Process Control

For the simulator, process control is the process that some process parameters of the simulator are controlled to maintain the desired values. This simulator simulates 11 process control loops. These loops are designed to maintain the desired values of the process parameters, including the primary water flow (F1), outlet temperature of the Heater (T2), outlet temperature of the Chiller (T3) (HX outlet temperature), the HX Tank level (L4) and pressure (P2), the Pressurizer pressure (P4) and level (L3), the Upper Tank level (L1), water inventory pressure (P1), and air pressure (P3) and flow (F2). These process variables are shown in Figure 2.3 with purple boxes. The methods of manipulating these process variables are categorized into two types. The first type is manipulated by one variable and the second type is manipulated by two variables. The primary water flow (F1), outlet temperature of the Heater (T2), outlet temperature of the Chiller (T3), the HX Tank level (L4) and pressure (P2), and air pressure (P3) and flow (F2) are controlled through valve CV-1, the Heater current C2, valve CV-34, valve CV-25, CV-21, CV-4, and CV-8, respectively. Each of the Pressurizer pressure (P4) and level (L3), the Upper Tank level (L1), and water inventory pressure (P1) is controlled by two valves. One is set as the main manipulated valve, while another has been fixed to remain open to avoid the main valve being fully closed, but the process variable is still higher than the desired value. Detail is shown in appendix B. All of these manipulated parameters are shown in Figure 2.3 with brown boxes.

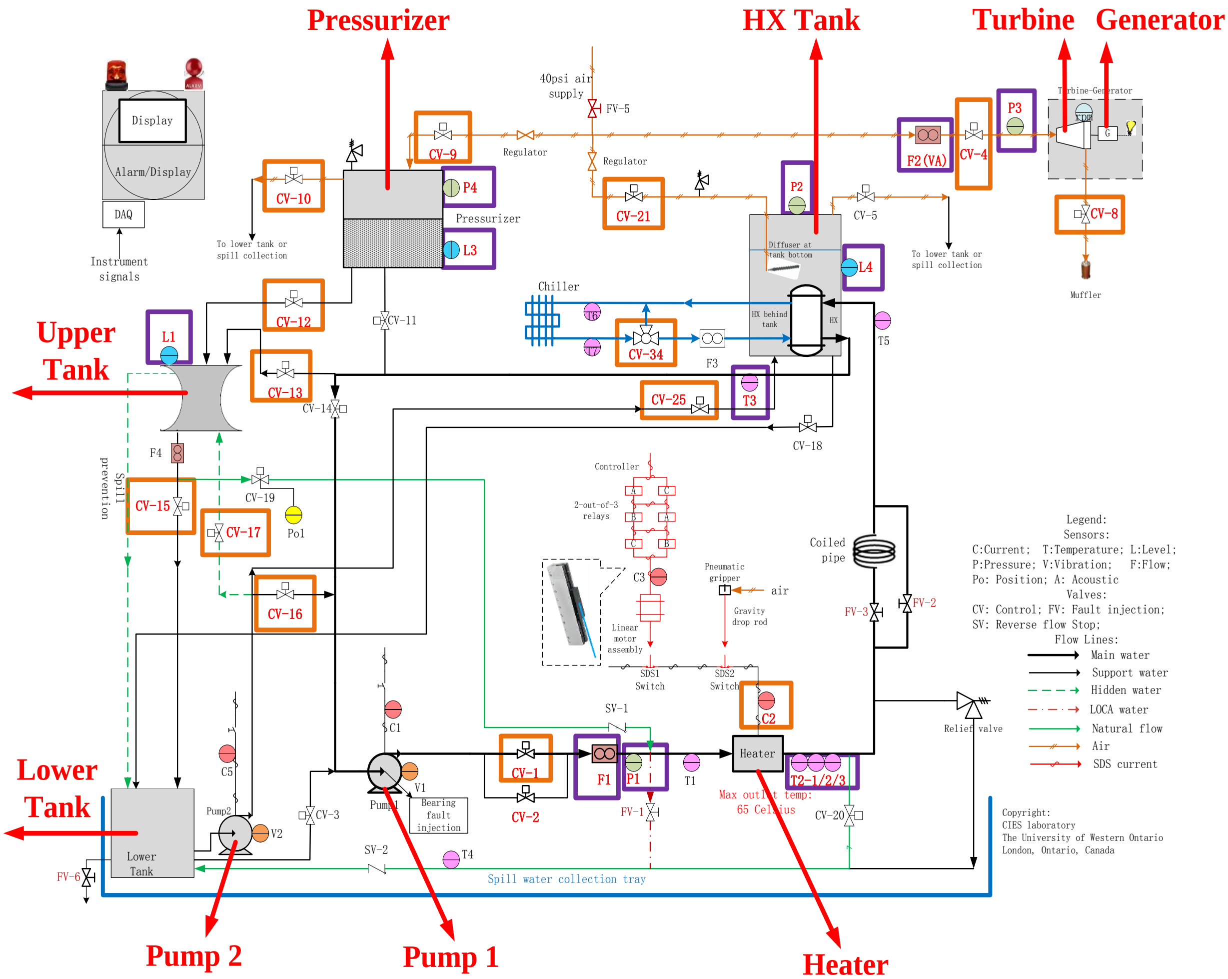

Figure 2.3 P&ID Diagram for NPCTF process variables

2.3 Sequence Control

Before the system starts, the tanks and pipes are required to be filled up and ready for operation. The Start-up procedure is designed for this objective. When the START button on the HMI is pressed, the pump begins to charge the Pressurizer, HX Tanks, and Upper Tank in sequence. When the levels of tanks reach certain values, such as 30% to 60%, the tanks are considered to be finished charging. In order to charge the related tanks, their corresponding valves are open, and then close after finishing the charge, as shown in Figure 2.4. For example, when charging the Pressurizer in the first step, valves CV-16, CV-1, CV-2, CV-11, CV-14, and CV-16 are open to fill in the pipes and Pressurizer tank.

Meanwhile, valves CV-3, CV-9, CV-12, CV-13, CV-15, CV-17, CV-18, CV-19, CV-20, CV-21, and CV-25 are closed. Refer to system P&I, diagram Figure 2.2 for the detail of these valves.

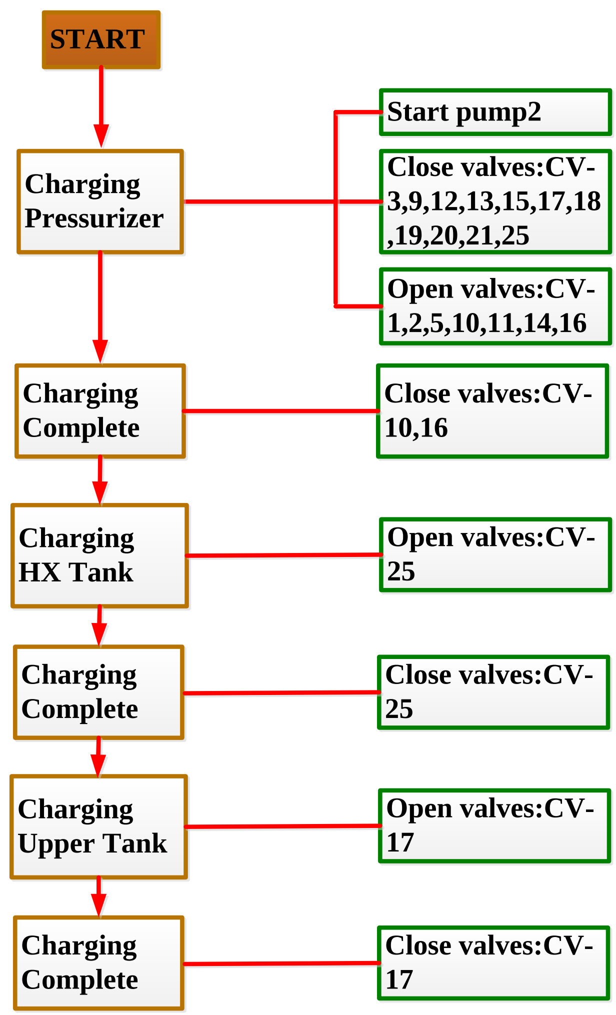

Figure 2.4 Logical diagram for start-up procedure

2.4 Safety Systems

There are three main safety systems for this NPCTF: SDS1 (Shutdown System Number 1), SDS2 (Shutdown System Number 2), and ECC (Emergency Core Cooling). SDS1 and SDS2 are designed to prevent the potential damage of the NPCTF when key operating parameters are out of the prescribed limit (see Appendix C for detail). If the cooling of the Heater is judged to be insufficient, the ECC system will act to cool the Heater. In CANDU NPP, the safety systems are not designed in DCS system. However, the safety systems take action to shut down reactor when the trip parameters are out of limits, such as low pressurizer level. The trip parameters are measured with 2oo3 voting. The simulator is used to simulate this protected process, for the system to take action in response to the trip parameters and 2oo3 voting.

2.4.1 Shutdown System Number 1 (SDS1)

If there are urgent situations, the reactor needs to shut down. The function designed simulates such urgent situations as HX Tank has abnormal water level (L4), and Heater has high outlet temperature (T2) or no enough cooling flow (F3). If any of these situations occur, the SDS1 signal is created and the heater is shut down. The logic is shown in Figure 2.5. The design philosophy of Shut Down Systems (SDS1 and SDS2) is based on triplicating the measurement of each signal, and initiating the protective action when any of the three channels indicate that a trip condition exists. This is named as 2oo3 voting. In order to simulate the 2oo3 voting, relays A, B, and C are used to simulate different channels. The logic is shown in Figure 2.6. If any two signals from channel A, B, or C become true, the SDS1 signal is created to shut down the reactor, i.e. the Heater in the simulator. In figure 2.5, in order to test the 2oo3 voting, parameters should be same. For example, if relay A is activated due to “HTR T2 high”, relay B or C should be activated by same parameters of “HTR T2 high” in order to activate SDS1 signal.

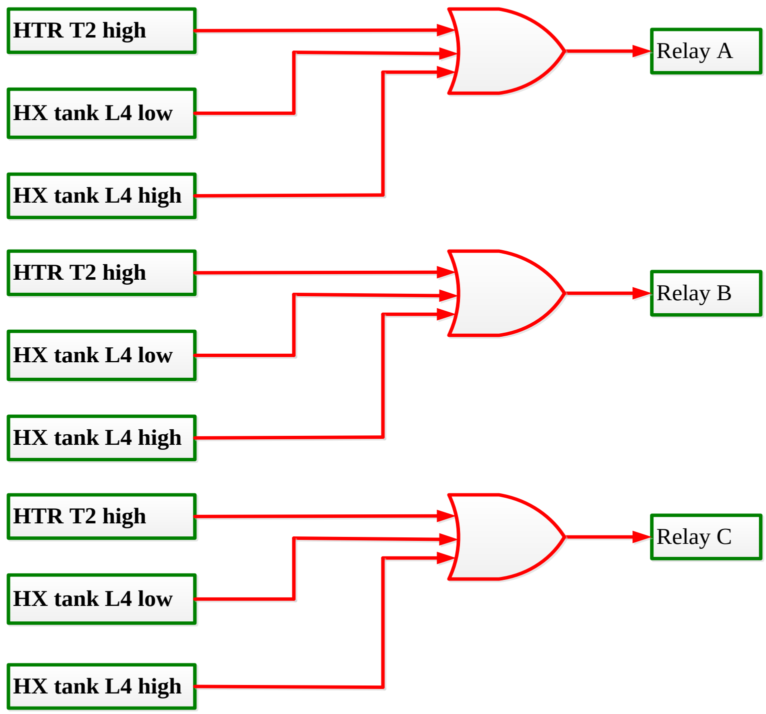

Figure 2.5 Logic function diagram of relays

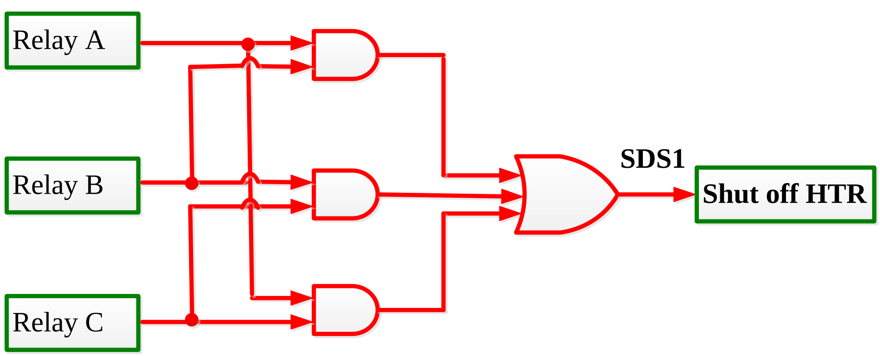

Figure 2.6 2oo3 logic function diagram of SDS1

2.4.2 Shutdown System Number 2 (SDS2)

SDS2 (Shutdown System Number 2) is an independent safety system. Three parameters are selected to simulate the potential risk of the Heater. If any of these parameters is higher than the threshold, SDS2 will act and shut down the Heater. The logic is shown in Figure 2.7.

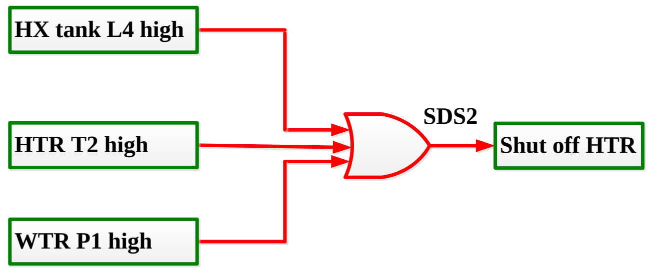

Figure 2.7 Logic function diagram of SDS2

2.4.3 Emergency Core Cooling (ECC)

The objective of an ECC system is to identify the potential risk of insufficient cooling of the Heater and then act to stop this situation, i.e. stopping the heat source (Heater) and feeding water from the Upper Tank to cool the Heater. As show in Figure 2.8, two conditions are simulated as Heater insufficient cooling conditions. Either SDS1 or SDS2 is active and Pump 1 is shut down. When this insufficient cooling condition is met, the ECC is created to shut off the heater and build the cooling circle of the Heater. As indicated in Figure 2.9 as seen using a highlighted green line, cooling water from the Upper Tank is fed to the Heater through the CV-19 and sent back to the Lower Tank. Pump 2 then feeds the water to the Upper Tank again. During this circling process, related valves are closed and opened to achieve the above functions, as shown in Figure 2.8.

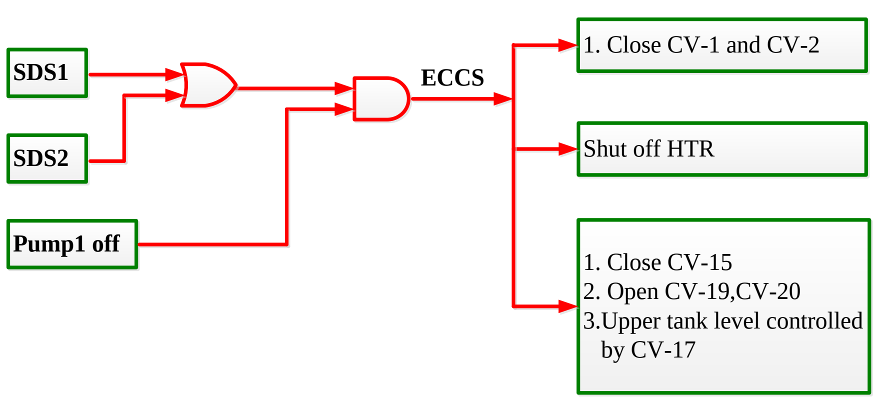

Figure 2.8 Logic function diagram of ECC

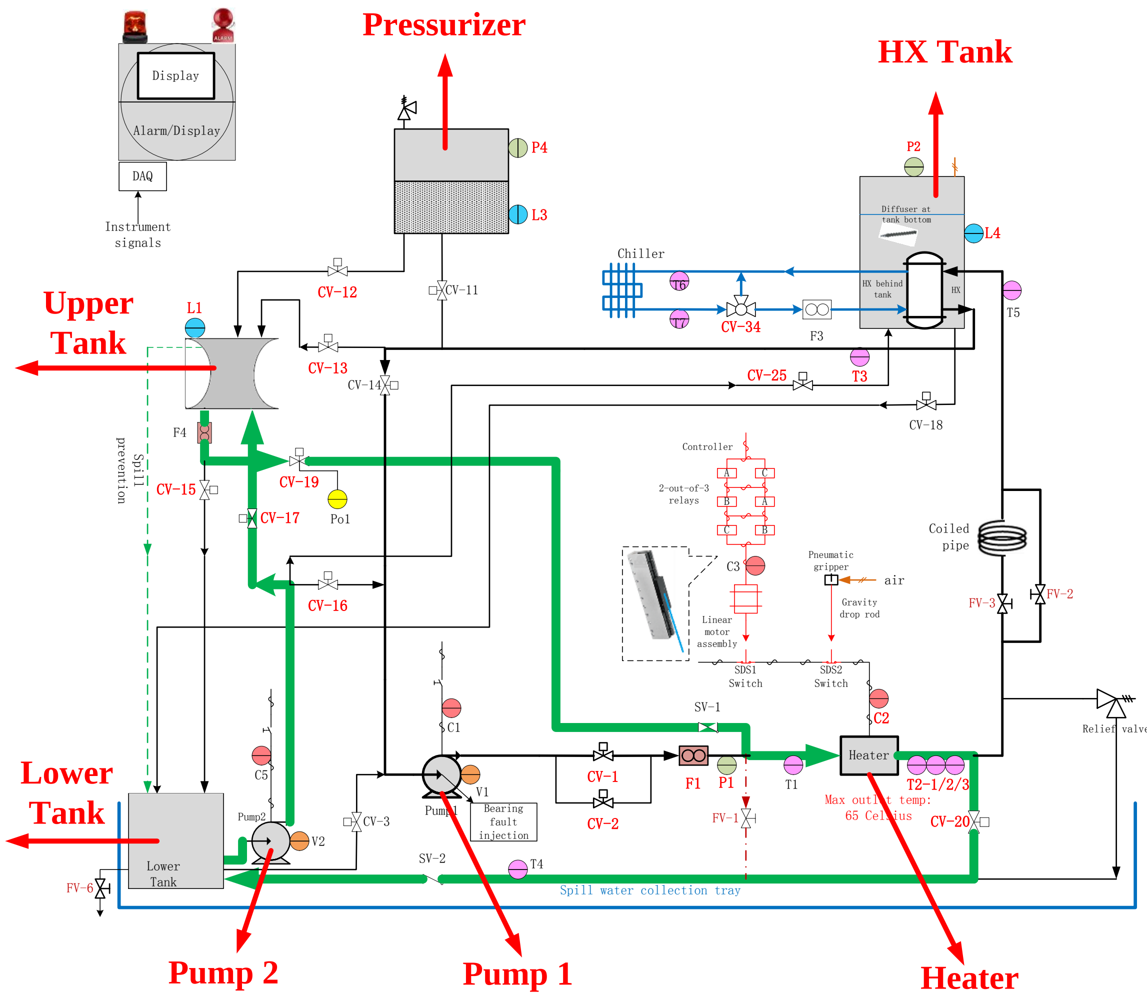

Figure 2.9 P&I diagram of NPCTF to show ECC

2.5 Operation Modes

The Pressurizer is used to maintain the pressure of the heat transport system. This operation mode is called Normal mode. However, under warm-up and cool-down conditions, the Pressurizer is isolated. In this situation, the feed and bleed system is used to maintain pressure and inventory of the heat transport system. This operation mode is named Solid mode. In NPCTF, two operation modes are also designed to simulate both

Normal mode and Solid mode. Moreover, an operation mode called Open mode is also designed in NPCTF for general I&C related research purposes such as advanced control theories, fault diagnostics, fault tolerance.

2.5.1 Normal mode of operation

Under Normal mode, the Pressurizer is used to maintain the primary water pressure (P1). The primary water circling is shown in Figure 2.10 using a highlighted brown line; water is fed by Pump 1 to Heater, cooled by Chiller and come back to inlet of Pump 1. As shown in Figure 2.10, the pressure of the Pressuriser is controlled by CV-9 and CV-10 and shown using a purple line while the level is controlled by CV-16 and CV-12. The related operation of valves and controllers in order to achieve above functions under Normal mode are shown in Figure 2.11.

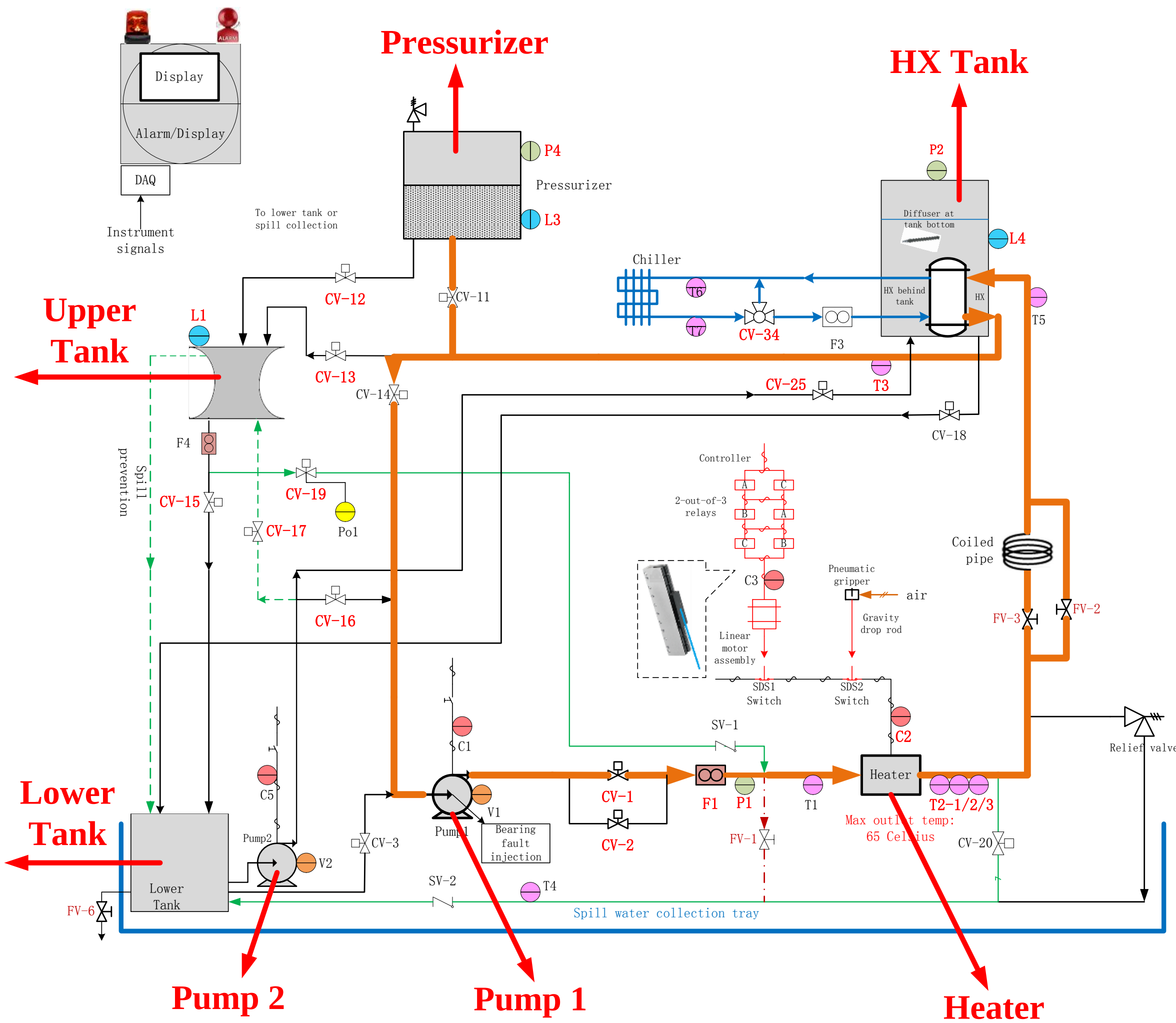

Figure 2.10 P&ID diagram of partial NPCTF in Normal mode

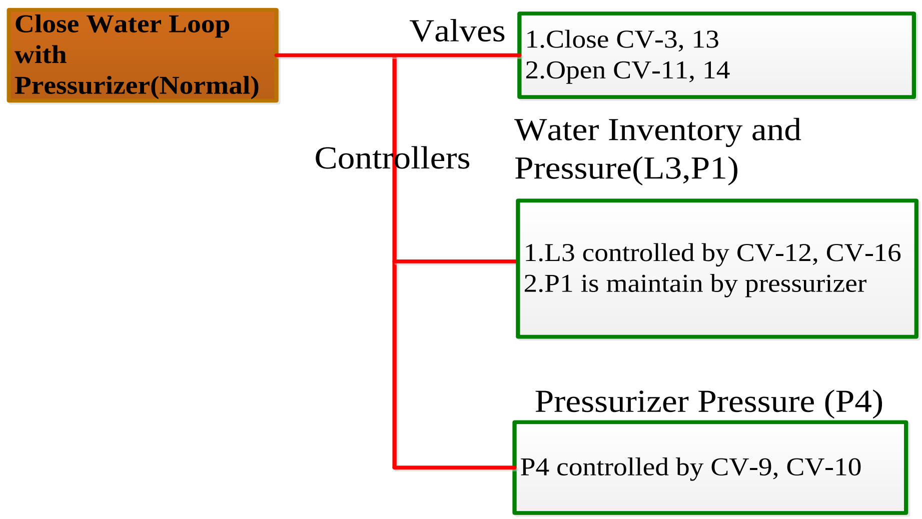

Figure 2.11 Logic function diagram of Normal mode

2.5.2 Solid mode of operation

Under Solid mode, the Pressurizer is isolated from water cycling. Primary water circling is shown in Figure 2.12 with a highlighted brown line: water is fed by Pump 1 to Heater, cooled by Chiller and come back to inlet of Pump1. As shown in Figure 2.12, the primary water pressure (P1) is controlled by CV-13 and CV-16 and is shown using a purple line. As shown in Figure 2.13, related valves and controllers are operated in order to achieve the above functions under Solid mode.

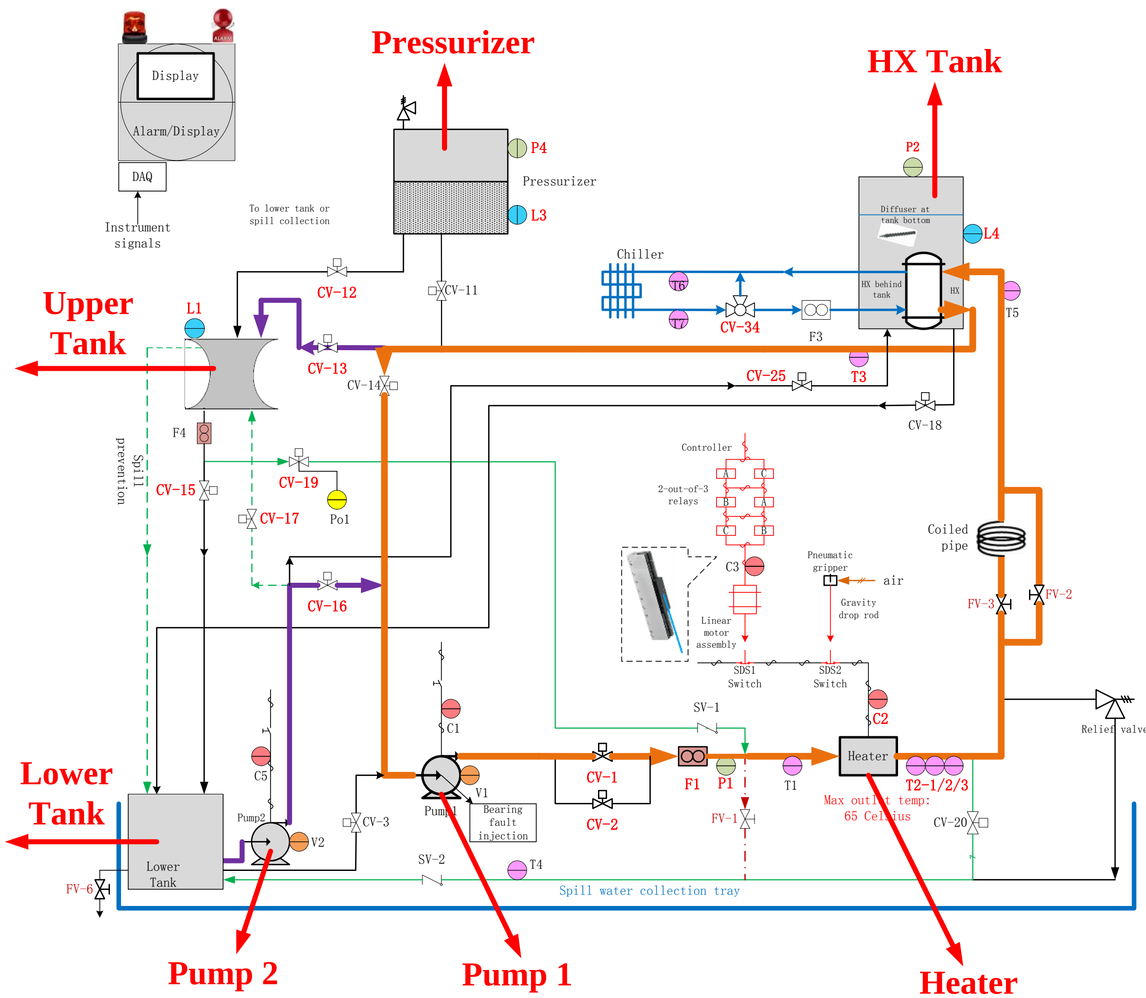

Figure 2.12 P&ID diagram of partial NPCTF in Solid mode

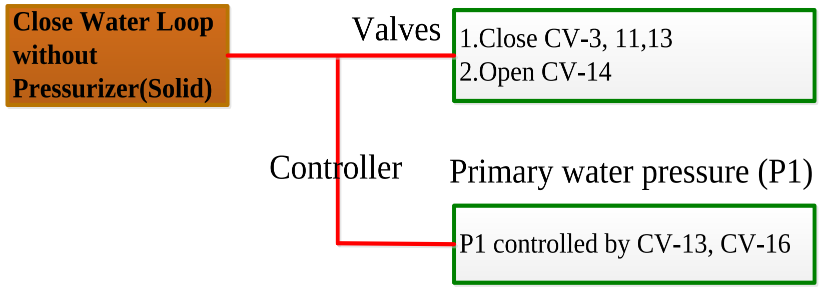

Figure 2.13 Logic function diagram of Solid mode

2.5.3 Open mode of operation

The NPCTF in the CIES lab at the University of Western Ontario is designed, not only for research of CANDU NPP, but for other general I&C related research as well. For example, if the research is only to investigate the nonlinear and adaptive control algorithm, and not related to CANDU, only a non-linear shaped Upper Tank and coiled pipe are used for this research. Therefore, NPCTF can be operated under the Open mode. Under Open mode, the Pressurizer is isolated from water cycling. Primary water circling is shown in Figure 2.14 using a highlighted brown line: water is fed by Pump1 from the Lower Tank via CV-3 to the Heater, cooled by the Chiller and then to Upper Tank. In order to build this cycling, valves CV-3 and CV-13 are open and valves CV-11 are CV-14 are close.

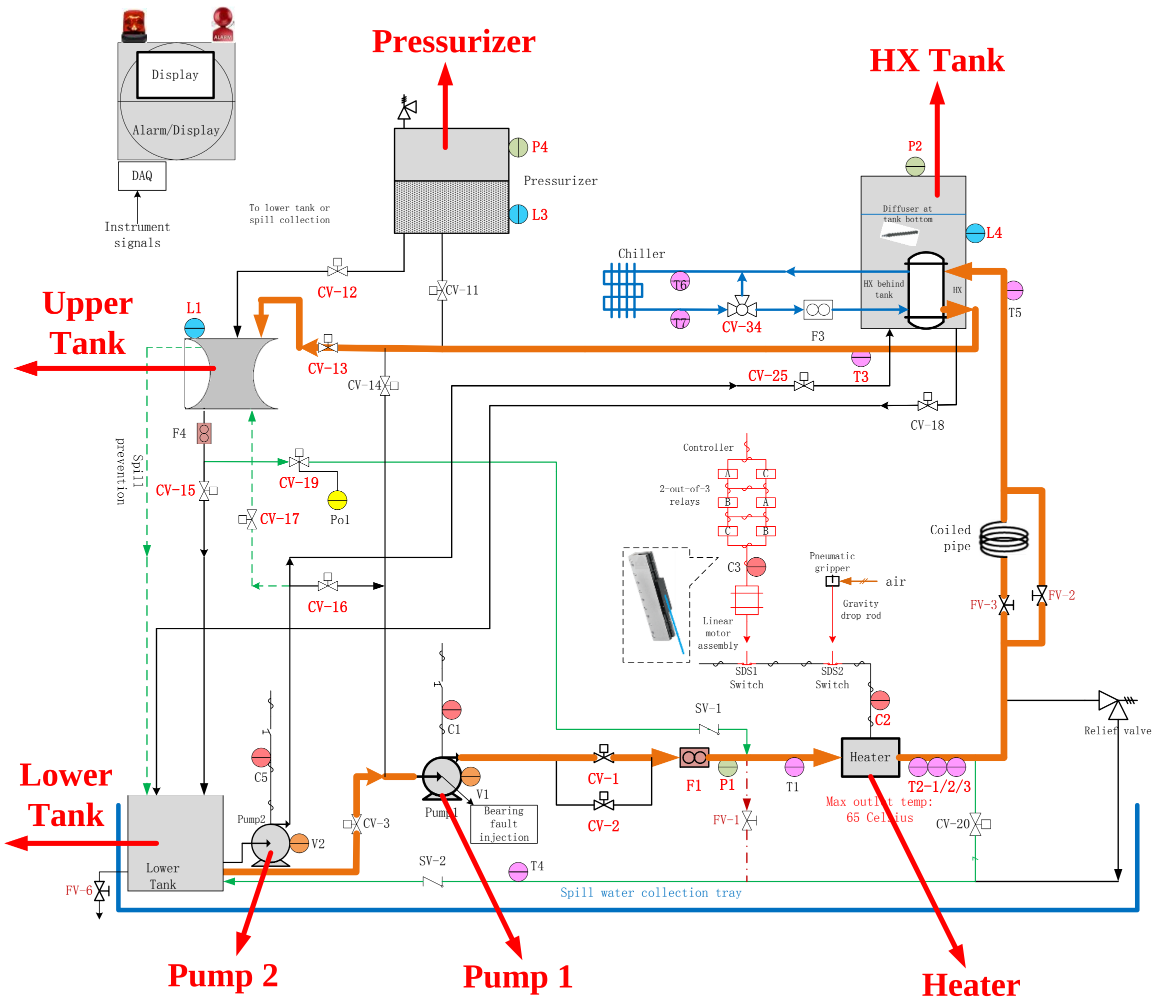

Figure 2.14 P&ID diagram of partial NPCTF in Open mode

2.6 Summary

In summary, in this chapter, four main functions of this NPCTF, are explained and described in detail. They are process control, sequence control, safety systems and operation modes. The construction and design of the DCS system to implement these functions are the topics of this research and are presented in Chapters 3 and 4. Their evaluations are also the topic of this research and are discussed in Chapter 5.

Chapter 3

Control System Design for NPCTF using DeltaV DCS

The Delta-V DCS system is one of the widely used DCS systems in industry [26] [27]. Therefore, DeltaV DCS system is selected to be used in this research to implement the functions of NPCTF described in Chapter 2: process control, sequence control, safety system, and operation modes.

3.1 Introduction to DeltaV DCS system

As a typical DCS system presented in Section 1.1.2, the DeltaV DCS system consists of devices at the control and supervisory level. As illustrated in Figure 3.1, the DeltaV DCS system in this research consists of workstation, two hubs, two system power supplies and two controllers for redundancy, and I/O cards. While the other components are standard design, I/O cards are different from project to project.

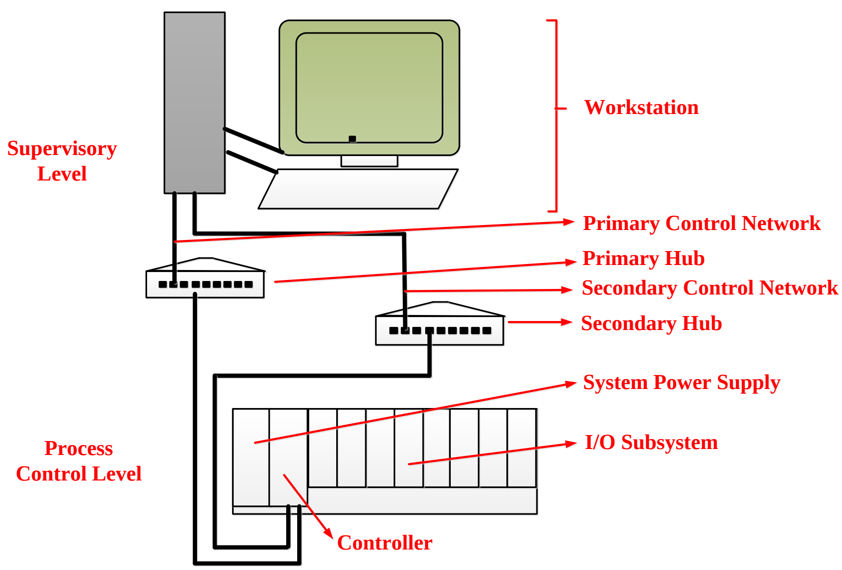

Figure 3.1 DeltaV system overview diagram

The workstation has all the software to configure the control strategies for the field device and HMI to monitor and operate process variables. The DeltaV system used in this research is Version 8 with the operating system Microsoft Windows XP and has OLE for Process Control (OPC) which stands for Object Linking and Embedding (OLE) for Process Control, to connect with third party software. The communication between PLC and computer is Ethernet with redundancy through two hubs.

At the control level, two power sources supply DC power to controllers. I/O modules function as the interface between field devices and CPU. Typically, there are four kinds of I/O cards: AI (analog input), AO (analog output), DI (digital input), and DO (digital output). I/O modules obtain data from field process and convert to discrete signal. Output cards send the signal from the process unit to drive field devices. One important function of an I/O card is to isolate high level real world signals from low level signals in the I/O bus with optical isolators to protect the I/O card. A/D converters in analog input cards convert analog signal to discrete signal for the/a process unit. D/A converters in input cards convert discrete signal to analog signal to drive field devices.

In this NPCTF system, there are 25 AI signals, 20 AO signals, 3 DI signals, and 12 DO signals. The details are listed in Appendix A. Each of the DeltaV I/O cards has 8 channels for 8 signals. Therefore, it needs 4 AI cards (4x8=32 channels), 3 AO cards (3x8=24channels), 1 DI card (8 channels), and 2 DO cards (2x8=16 channels).

3.2 System Configuration and Connection

3.2.1 DeltaV cabinet configuration

All devices mounted inside the DeltaV cabinet are shown in Figure 3.2. They mainly include system power supplies (P1 and P2) which provide 12VDC power for controllers and other cards mounted on carriers, carriers which hold all the modules, controllers (C1 and C2), I/O cards (AI, AO, DI, and DO), switches, terminals, relays (R_DO1_1~8) and power supply (PWR1 and PWR2) for all devices in the cabinet. They are the devices at the control level.

Two power supplies, PWR1 and PWR2, convert 120VAC line power to 24VDC and provide 24VDC power to the entire DeltaV system in the cabinet. A circuit breaker (CB) is mounted before two power supplies to protect the system. As indicated in the Figure 3.2, power supply number one (PWR1) supplies 24 VDC power to the system power supply P1 which converts 24VDC to 12VDC and carrier 3. Power supply number two (PWR2) supplies 24 VDC power to the system power supply P2 which converts 24VDC to 12VDC, other carriers, and DO intermediate circuit power through fuse F. Carriers provide power to all the cards mounted on the carriers. The two switches have separated AC power supply. R_DO1_1~8 are relays used for connecting AC powered field devices to DeltaV DO cards which can only connect DC powered field devices. A detailed DO intermediate circuit is discussed in section 3.2.2.

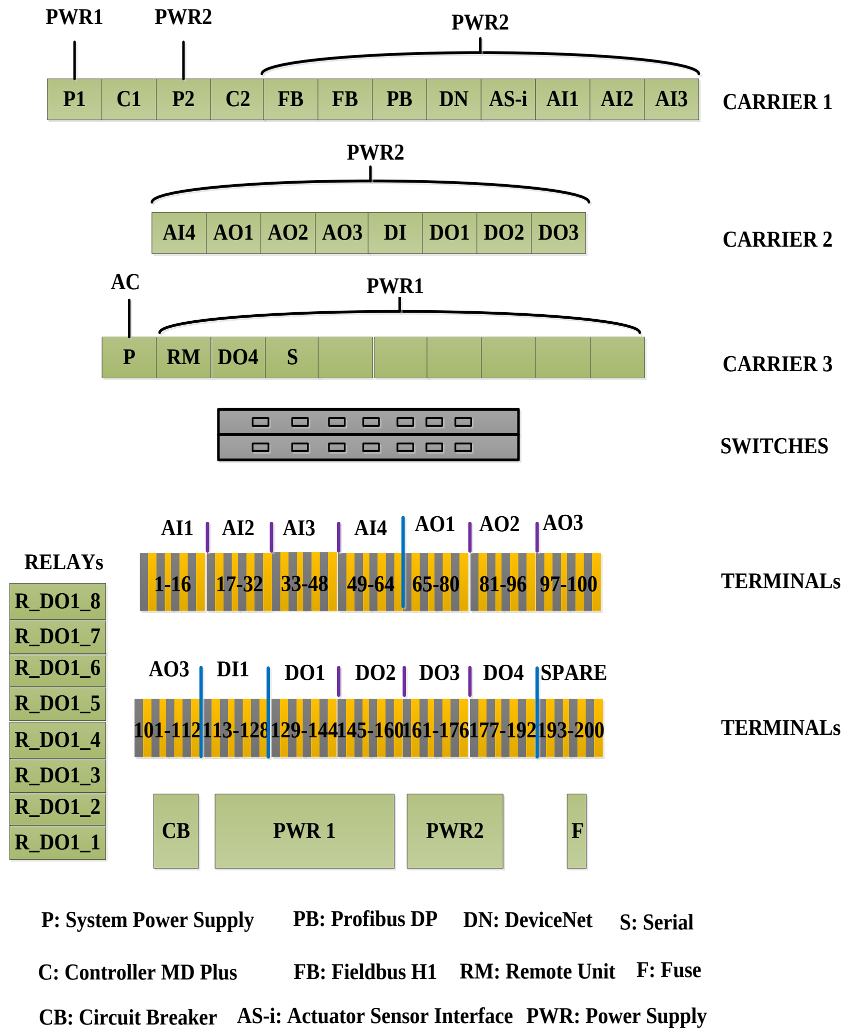

Figure 3.2 Device schematic diagram in DeltaV cabinet

3.2.2 I/O card wiring diagrams

I/O cards communicate with CPU through carriers which hold all the I/O cards. Meanwhile, all the I/O cards need to be wired to the field devices in order to communicate with field devices.

All the process variables, such as pressure, level, flow, temperature, are sent to the PLC in the format of 4-20 mA through analog input (AI) cards. Because the 4-20 mA analog input signals have power source, 4-wire AI cards are used to isolate the power source in the AI cards to prevent the conflict of the two power sources in the AI signal circuit. Wiring of the AI card is shown in Figure 3.3. Two wires of each signal from transmitter ("T") are connected to the AI card termination ("+" and "-"), as shown in Figure 3.3.

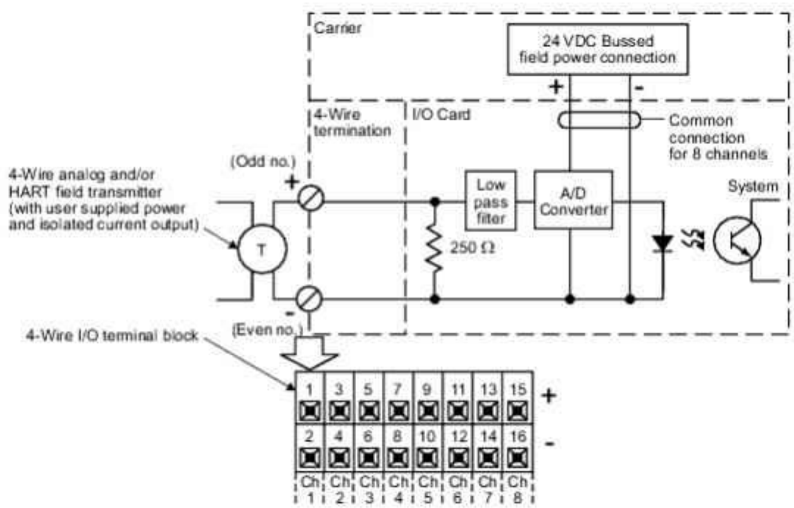

Figure 3.3 AI card wiring [28]

All the manipulated variables (AO), such as opening of valves and current of pumps, are in the format of 4-20 mA. For analog output (AO) signals to activate the relays of valves or pumps, the power sources come from AO cards. The two wires of each AO signals from field devices ("Load") are directly wired to AO cards termination ("+" and "-"), as shown in Figure 3.4.

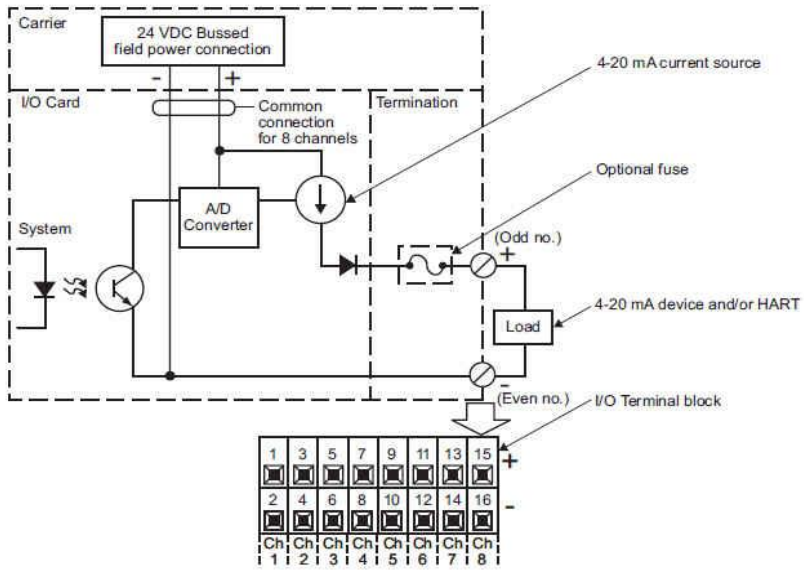

Figure 3.4 AO card wiring [28]

Digital input (DI) signals indicate the switch status. The power source comes from DI cards. The contacts of the switch are directly connected to the DI card termination ("+" and "-"), as shown in Figure 3.5.

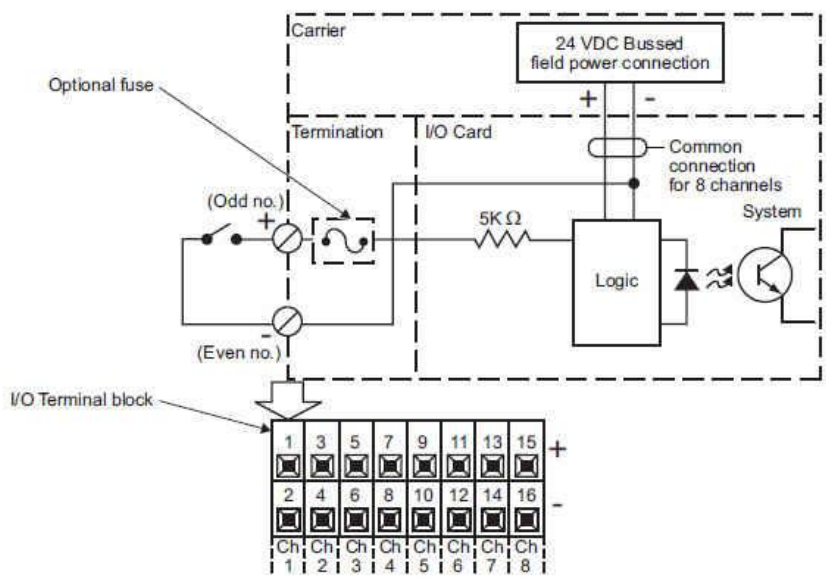

Figure 3.5 DI card wiring diagram [28]

For digital output (DO) signals, DeltaV DO cards only allow DC power source. Therefore, except signals of “ROD DOWN”, “DOWN”, “INTERM”, and “HOME” which are powered by DC, other digital output signals which are powered by AC cannot be directly wired to DeltaV DO card and, thus, an intermediate circuit is needed. Therefore, signals “ROD DOWN”, “DOWN”, “INTERM”, and “HOME”, are directly wired to DO card termination (“+” and “-”), as shown in Figure 3.6. Others, shown as “Load” in Figure 3.7, are wired to the contacts of the relays (“RELAYS R_DO1_1~8” in Figure 3.7) in the intermediate circuit and the relays (“RELAYS R_DO1_1~8” in Figure 3.7) are wired to DO cards, as shown in Figure 3.7.

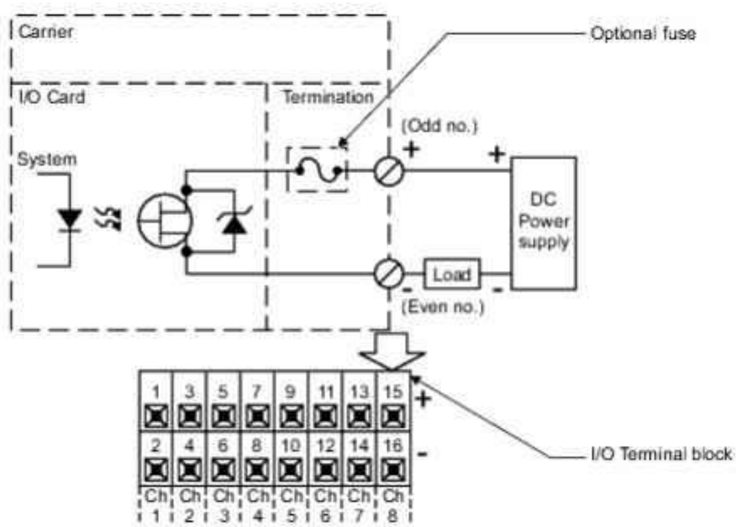

Figure 3.6 DO card wiring diagram [28]

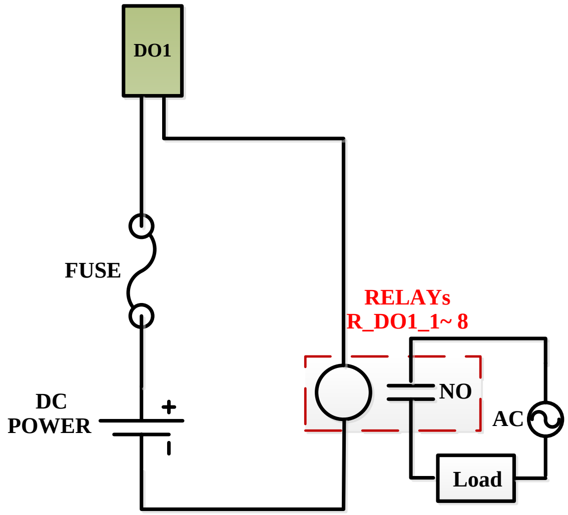

Figure 3.7 DO signal intermediate circuit

3.2.3 Connection to field devices

The connection includes connecting the server to the switches which are connected to PLCs with a Cat5e Internet cable, and connecting I/O cards to field devices using 22 gauge wires, as indicated in Figure 3.8.

The DelatV system has primary and redundant networks. Therefore, there are primary and secondary ports for two networks in the server. Accordingly, there are two switches for two networks. Each of two PLCs is connected to two switches: primary port of PLC will be connected to the primary switch, and the secondary port of the PLC will be connected to secondary switch. Figure 3.8 shows the wiring.

The wiring from I/O cards to the field devices are connected through terminals in the DeltaV cabinet and DCS cabinet, as shown in Figure 3.8. The wiring diagrams of each kind of I/O cards are presented in section 3.2.2.

All signals, their terminal numbers in DeltaV cabinet and DCS cabinet, their cable number, position in I/O cards, and their signal types are listed in the appendix A.

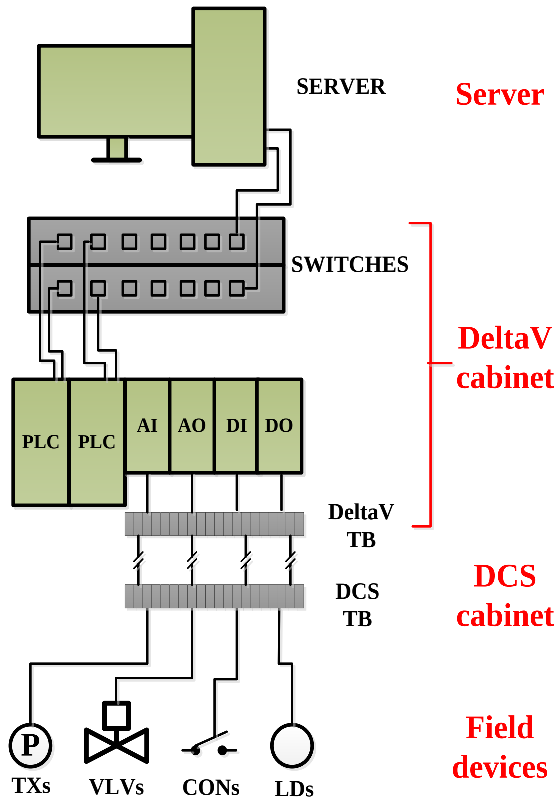

Figure 3.8 Schematic diagram of wiring system

The actual devices corresponding to the system indicated in Figure 3.8 are shown in Figure 3.9 to 3.12. Figure 3.9 indicates the server and the screen. Figure 3.10 shows the DeltaV cabinet where all the PLCs, I/O cards, switches, DeltaV terminals and powers are mounted inside the cabinet. Figures 3.11 shows the DCS cabinet where all of the signals from the DeltaV cabinet are connected to the field devices. Figure 3.12 shows all filed devices including control valves and transmitters.

Figure 3.9 DeltaV server and screen

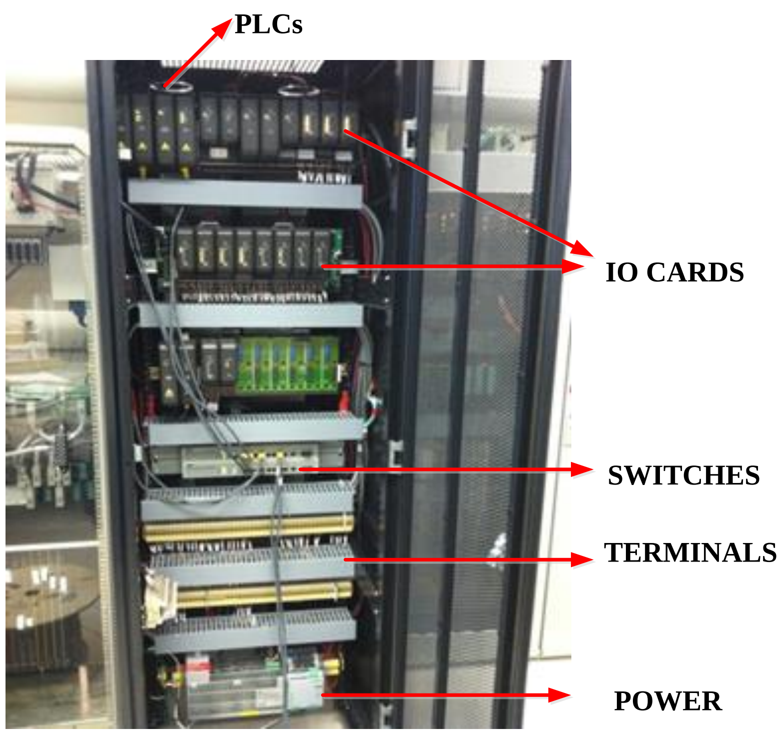

Figure 3.10 DeltaV cabinet

Figure 3.11 Wiring of DCS cabinet

\

Figure 3.12 Field devices of NPCTF

3.3 Programming Architecture Design

As discussed in Chapter 2, there are mainly three functions implemented in this research. The hardware of the DeltaV system, in order to implement these functions is discussed in Section 3.2. This section will present the structure to implement the functions.

The whole control structure involves many subsystems and algorithms. A clear and logical organization of the subsystems and algorithms facilitate not only the running of the program, but also design maintenance and troubleshooting.

As shown in Figure 3.13, a typical DeltaV system organizes the programming architecture as hierarchical structure: “Areas”, “Modules”, “Algorithm”, “Function Blocks or Composites”, and “Parameters”.

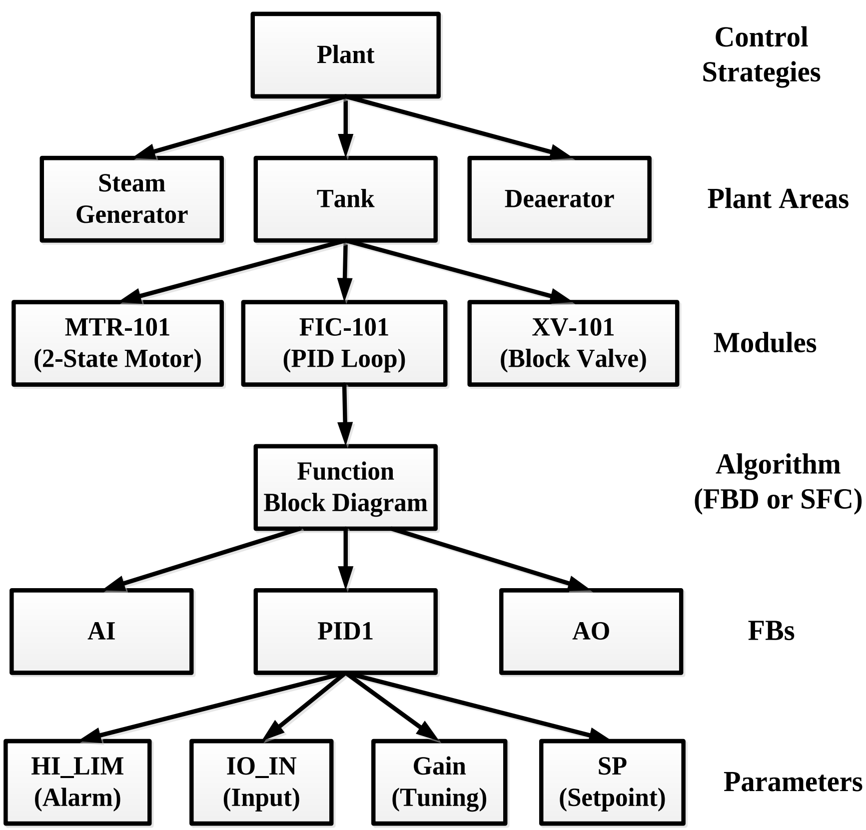

Figure 3.13 Programming architecture schematic diagram of typical DeltaV system

For NPCTF, control architecture of the whole system, named as “PCTF_SIMON”, is designed in the “Areas” level. Each sub-system is designed at the “Modules” level, as shown in Figure 3.14. Each sub-system includes the algorithm and logic which are programmed using languages of FBD (Function Block Diagram) or SFC (Sequential Function Chart). Each module consists of different kinds of function blocks with their own parameters, such as Proportional-integral-derivative (PID). In order to implement the functions (i.e. process control, safety, sequence, and operation modes) described in Chapter 2, twenty modules are designed for this NPCTF system, as shown in Figure 3.15. Each process control loop becomes one module and there are eleven modules for process control. Each type of I/O signals has one module and, thus, there are four modules for I/O signals. There are five other modules for safety, sequence, operation mode, stop system, and ratio.

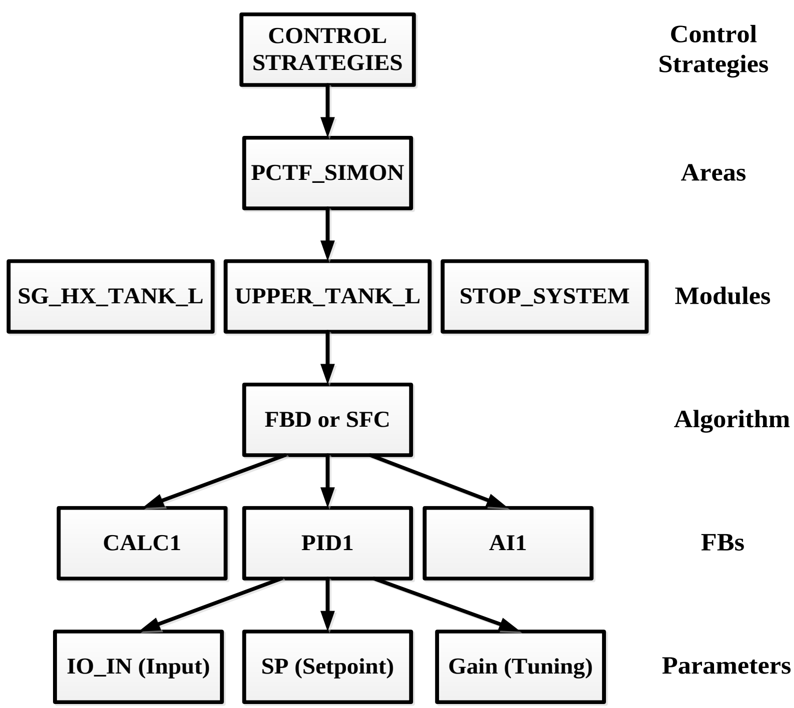

Figure 3.14 Programming architecture schematic diagram of NPCTF

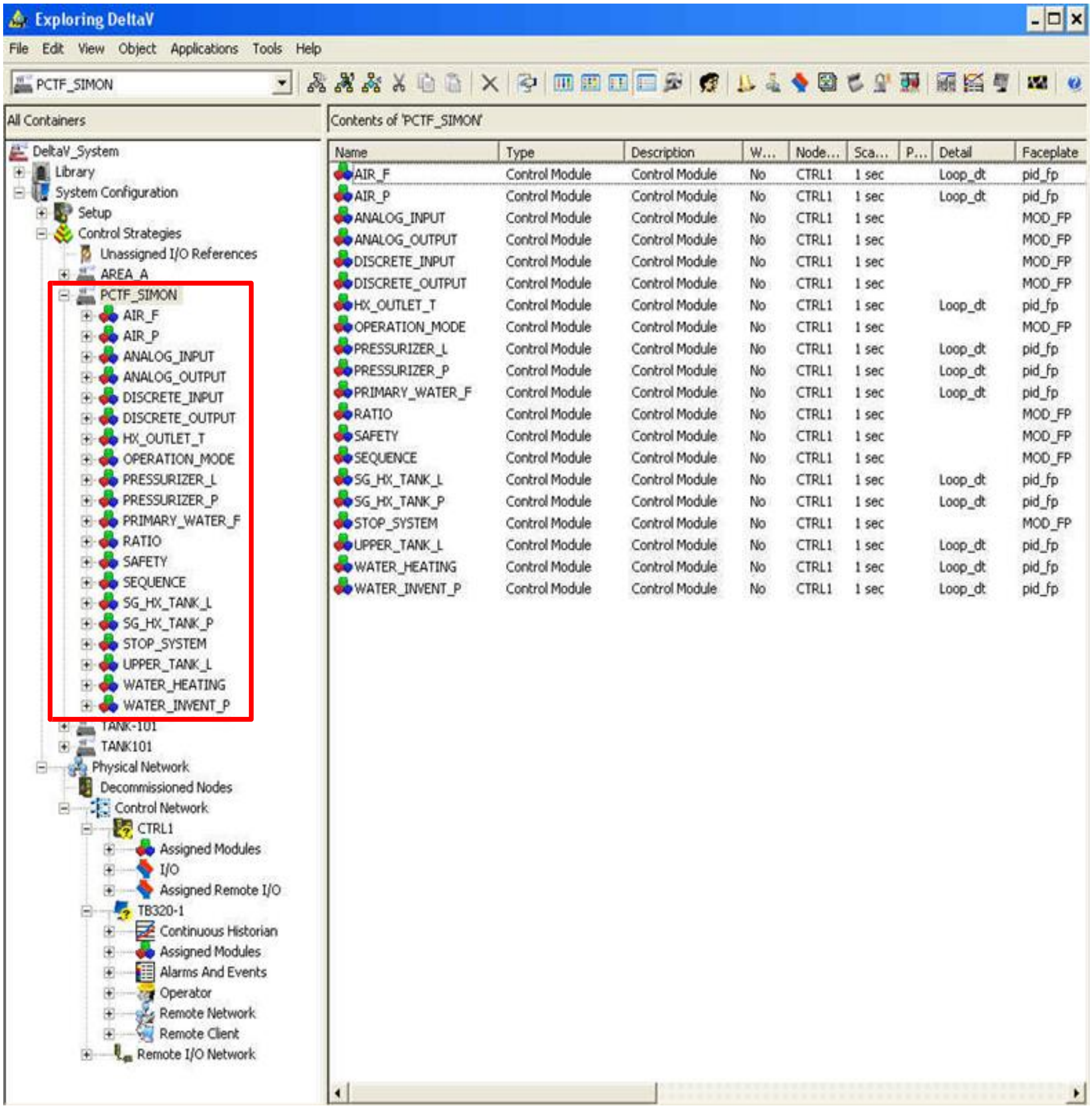

Figure 3.15 Programming architecture diagram of NPCTF

3.4 DCS Programming Languages (IEC 61131-3)

As discussed in section 3.3, the modules of the programming architecture are programmed with industrial languages prescribed in the standard of IEC 61131-3.

The programming of the early PLC was of ladder the diagram type. It resembled the circuit diagram and still remains one of the most popular PLC programming languages [29] [30]. However, with the integration of distributed control, sequential control, and factory control system, the controller is required to do multi-task operations. However, ladder diagram language is difficult to control application with multiple events occurring at the same time [31]. Other limitations includes: lack of software structure, problem in reusability, poor data structure definition, lack of support for describing sequential behavior, and difficulty in handling complex arithmetical operation [32]. Different PLC manufacturers developed their own programming languages to solve these problems. The multiplicities of languages and even proprietary programming languages or tools were too complex for customers to handle. Standardization of these practices became necessary. IEC adopted the prevalent programming practices of major PLC manufacturers and published the standard IEC 61131-3. In fact, language defined by IEC 61131-3 is also recommended to implement application functions of nuclear power plants [33].

IEC 61131-3 is the standard of programmable language and defines 5 kinds of languages: Instruction List (IL), Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), and Sequential Function Chart (SFC). FBD and SFC are the two main programming languages supplied by the DeltaV system and used in the research for programming.

3.4.1 Function block diagram (FBD)

FBD is one of the graphical languages in the IEC61131-3 for programmable logic controller design. It uses the following graphical conventions as shown in Figure 3.16 [32]:

- A function block is a rectangular bock with inputs entering from the left and outputs exiting from the right.
- The name of the function block and its instance are shown on top of the block.
- The inputs and outputs of the function block are shown within the block.

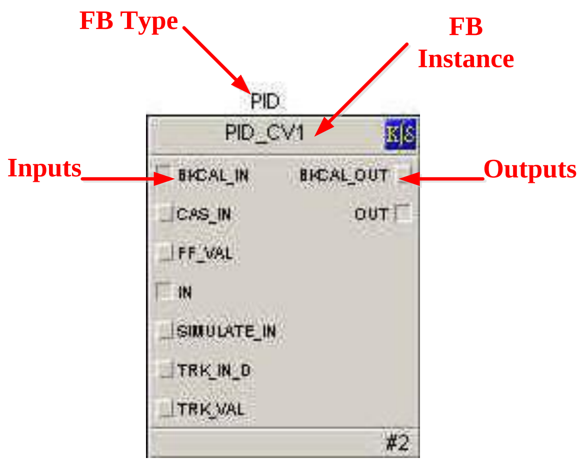

Figure 3.16 Individual function block

The function block diagram (FBD) network includes connections, graphical elements for execution control ("Jump" in Figure 3.17), and graphical elements to call function or function blocks. Some platforms of manufacturers include connectors which are used to construct long network. These connectors are not control or data flow elements [34].

Connections are horizontal or vertical lines which carry the signal and connect the functions and function blocks. Signals flow from left to right or from top to bottom along the connections. Left or top is input, and right or bottom is output. In special cases, the signals can be fed back, as shown in Figure 3.17. It is possible to split a line into several connections. For example, in Figure 3.17, the connection from “OUT” of “AI1” is split into two connections. However, it is forbidden to connect more than one output to one input. For example, in Figure 3.17, it is not allowed to connect more than one connection to “CAS_IN” of “AO1”. Different data type of variables, such as a floating-point double word or Boolean value, can flow along the connections. However, the output parameter must be the same data type as the connected input parameter.

In an FBD network, the execution could transfer from one part of network to another. As shown in Figure 3.17, “PV” points to the signal from “AI1” to other part of FBD network.

The graphical representations for calling an FB or function are similar. They have input and output parameters which are indicated in the block, as “PID” in the Figure 3.17.

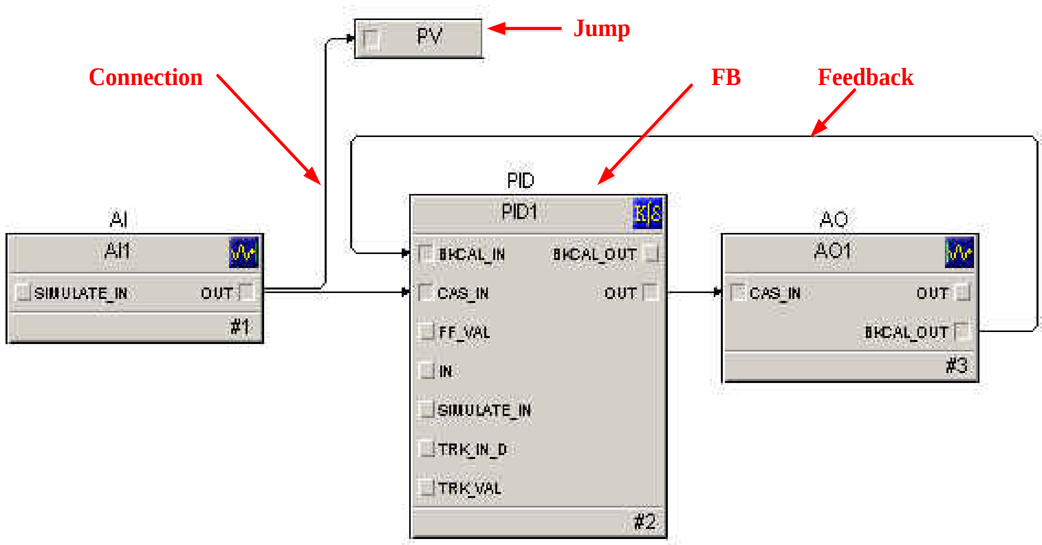

Figure 3.17 FBD Network

The FBD network is evaluated by PLC according the following rules:

- Before the functions or function blocks in the network are evaluated, all the inputs values from other elements must be available.

- The outputs of a function block cannot be considered available until all its outputs have been evaluated.
- The FBD network is not complete until all the outputs of its functions and function blocks have been evaluated.

3.4.2 Sequential Function Chart (SFC)

Sequential function chart (SFC) is a graphical language for programmable logic controllers (PLCs). It is one of the five languages defined by IEC 61131-3 standard. It can be used to program processes that can be split into steps.

As shown in Figure 3.18, the sequential function chart (SFC) network consists of Steps (rectangular boxes: S1 to S4), Transitions (bars with identifiers: T1) and Links. Steps are associated with actions, Transitions are logic conditions, and Links connect Steps and Transitions. A Step is either active or inactive. When a Step is active, the associated actions are executed until the Step becomes inactive. The status of active or inactive is decided by Transition. The Transition is programmed by Transition condition which is a Boolean expression. When the Transition becomes true, the Step immediately before it becomes inactive and the one immediately after it becomes active. Therefore, the processes with a step by step state behavior are suitable for programming with SFC.

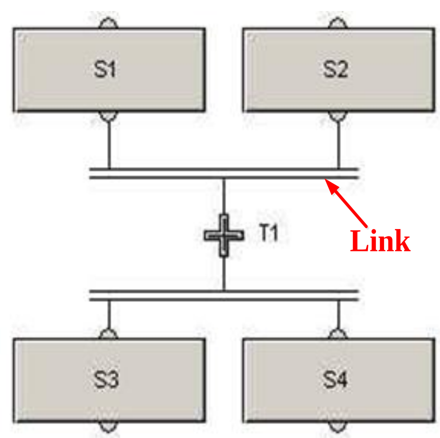

Figure 3.18 SFC network

3.5 Programming Algorithms for Modules

The purpose of the programming is to implement the functions described in Chapter 2.

Section 3.3 discussed the programming architecture of this project. Modules of the programming architecture are implemented by programming. Under the structure, as shown in Figure 3.19, modules of STOP_SYSTEM and SEQUENCE are step by step with sequence in nature and, thus, are programmed using SFC language. For the STOP_SYSTEM, when the operator wants to stop the system, the system is reset to original status, i.e. all valves to normal status: open for NO (normal open) valves and closed for Normal Close (NC) valves; all the heaters, and pumps are stop; and all the tanks are empty. The operation of these devices is sequential. The main purpose of the SEQUENCE is to fill up the pipes and tanks. The filling procedure is step by step. Refer to Figure 2.4 in Section 2.3 for the logic procedure and Section 2.3 for more function details. The Pressurizer is first filled up and then the HX Tanks. The Upper Tank is the last one to fill up; therefore, they are programmed using SFC language. Detailed programming is shown in appendix D.

Other modules are programmed using FBD. AI, AO, DI, and DO modules are the interfaces between signals from and to local devices and programming algorithm. The SAFETY module is used to implement the safety functions of SDS1, SDS2, and ECC described in section 2.4 (referring Figure 2.5 to 2.8 in section 2.4 for their logic diagrams).

The module of OPERATION_MODE implements three kinds of operation modes described in section 2.5, Normal, Solid, and Open mode (referring to Figure 2.11 and 2.13 in section 2.5 for logic diagrams). Other modules, such as HX_OUTLET_T, are the controller algorithm for process control described in section 2.2.

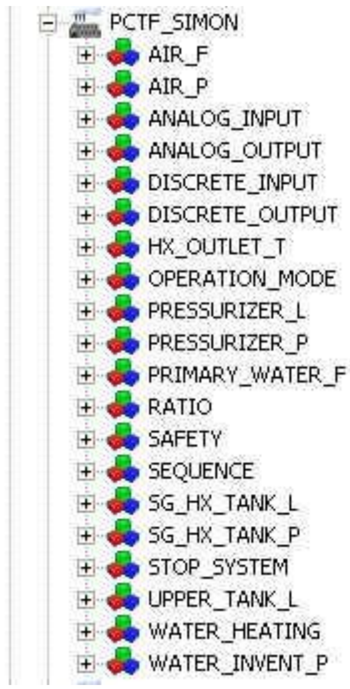

Figure 3.19 Programming architecture of NPCTF

3.6 Human Machine Interface (HMI) Configuration: Graphics and Faceplates

The Human Machine Interface (HMI), as indicated in Figure 3.20, is created mainly to monitor and control the process in order to implement the functions introduced in Chapter 2: process control, sequence control, safety shutdown, and operation modes. It is similar with the P&ID diagram of NPCTF of Figure 2.2 in Chapter 2.

There are two kinds of graphics of HMI: dynamic pictures and static pictures. Static pictures do not change their properties while dynamic pictures communicate with the data base to change their properties or database. Static pictures show the process system: devices and pipes, tanks. Dynamic pictures indicate the process status and are used to operate the process through faceplates and buttons. For example, when press the button

“STOP SYSTEM” in Figure 3.20, the button becomes brown and the system is also stopped.

All the valves, except manual valves, indicate the full open status as red color, full closed status as green and between full open and closed as yellow. If the safety system, “Safety” in Figure 3.20, is active, the HMI will show the active one in red: SDS1, SDS2, or ECC. From HMI, as show in Figure 3.20, the operator can operate the Pumps, Heater, and Chillers, stop the system, start the sequence, and select the operation modes through related buttons, as indicated in Figure 3.20. For process control, faceplate is the interface for operator to control the process. By clicking the “Faceplate” icon in HMI, the operator can open the faceplate window as shown in Figure 3.21 to perform the operation of process controllers: select values of set-point or output, manual or auto. By click the “trend” in the faceplate, operator can open the trend window as shown in Figure 3.22 to monitor the current trend of three parameters of controller: process variable (PV), set-point (SP), and controller output (OUT). The controller is also allowed to change the controller tuning parameters through “detail” icon in faceplate, as shown in Figure 3.23.

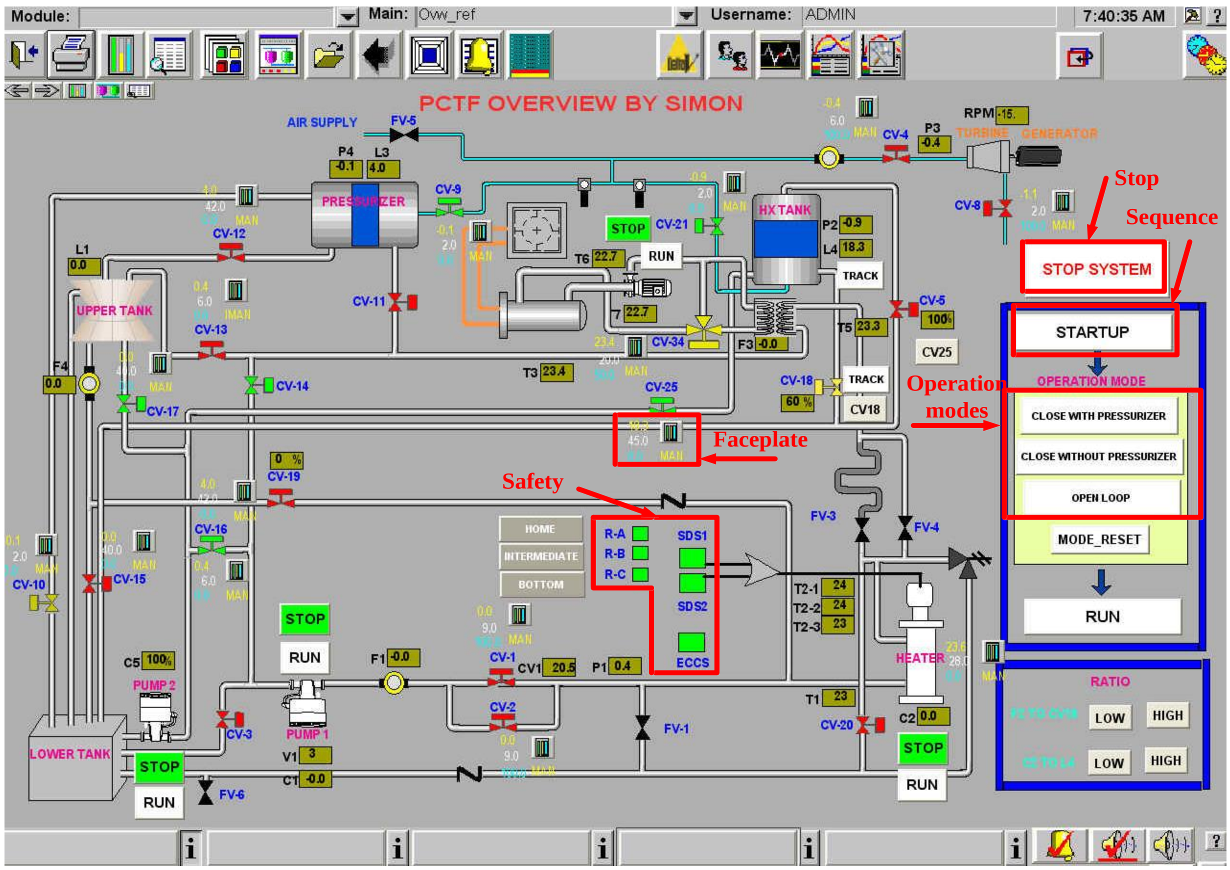

Figure 3.20 HMI of NPCTF

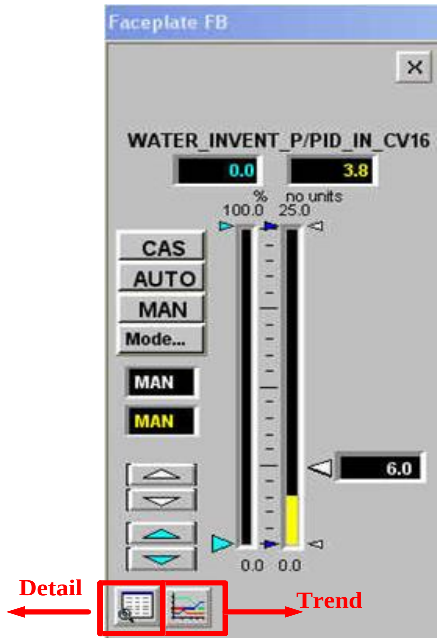

Figure 3.21 Typical faceplate

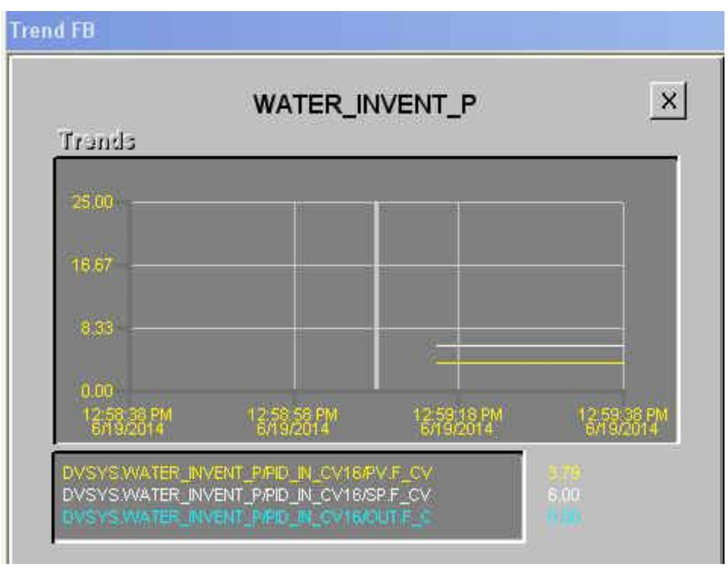

Figure 3.22 Current trend of controller parameters

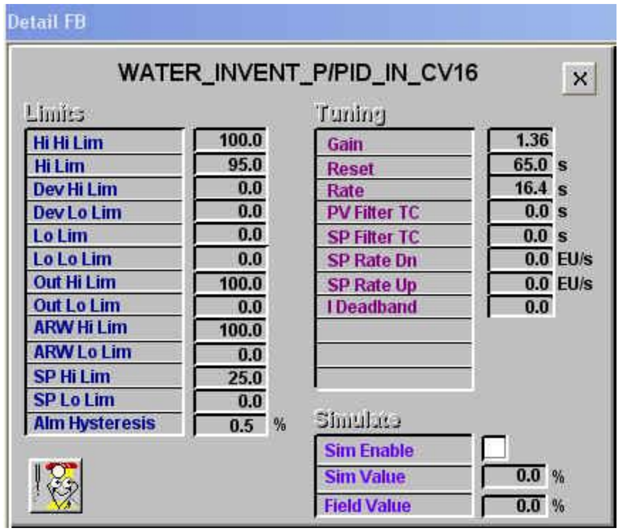

Figure 3.23 Controller tuning parameters

3.7 Summary

In summary, this chapter has described the construction and design of the DeltaV DCS system which is used to implement the functions explained in Chapter 2. The programming to implement these functions in this DCS system is presented in this chapter. The main industrial languages, FBD and SFC, used to make the program are explained. The design of HMI used to operate the NPCTF is also discussed in the chapter.

Chapter 4

Process Controller Design in DeltaV DCS

As described in Chapter 2, process control is part of the functions designed and evaluated in this research. This chapter describes the design of controller for process control. There are 11 process control loops which are given in Appendix B. Focus is given to the SG Tank level control to illustrate the detail analysis and design process. Other loops are tuned based on experience.

4.1 Development of Control Strategies

For process control described in Chapter 2, the feedback control system shown in Figure 4.1 is designed to control the process variables (PVs). The process variables, such as Heater outlet temperature (T2), are measured and sent to the controller. The controller compares the measured variable with the value set by operator (SP) and calculates the action (M) needed to change the process in order to make the process variable equal to the value designed by the operator.

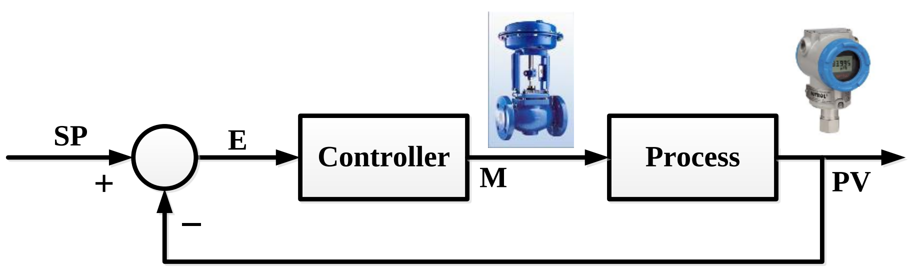

Figure 4.1 Process control diagram

For controller algorithm in DCS system, such as DeltaV system, the control strategies could be advanced control strategies. The NPCTF in this research is tested with PID controllers which are widely used in variable industries.

4.1.1 Proportional-integral-derivative (PID) algorithm

Error (*E*):

Error (*E*) is defined as the deviation from set-point [35]. For different DCS systems, the definition of error may be different. In DelatV system, *E* is defined as [28]:

$$E = SP - PV \tag{4.1}$$

where

SP is an input variable that sets the desired value of the controlled variable manually, automatically, or by means of a program in the same units as the controlled variable.

PV is the process variable to be controlled, such as water temperature, pressure, level, etc.

Proportional Control (P):

Proportional action keeps the output of the controller in proportion to the error:

$$M = K_c E + C \tag{4.2}$$

K_c is the controller gain. E is the error between set-point and process variable. *M* is the output of the controller. The term *C* is necessary since it is hard to be the case that zero error coincides with zero controller output.

In order to obtain the value of the controller output, it also needs to know *C*. Therefore, another form of controller output is often used. Assume current error is E_n, and previous error is E_{n-1}, then we have:

$$M_n = K_c E_n + C \tag{4.3}$$

$$M_{n-1} = K_c E_{n-1} + C \tag{4.4}$$

Subtracting (4.3) by (4.4) gives:

$$\Delta M = K_c(E_n - E_{n-1}) = K_c \Delta E \tag{4.5}$$

Using this version to calculate the controller output, it needs to calculate the current and previous errors between set-point and process variable.

Integral Control (I):

Integral control eliminates the offset of the process variable from the set-point. It keeps changing the controller output as long as the error exists:

$$\frac{dM}{dt} = \frac{K_c}{\tau_I} E \tag{4.6}$$

τ_I is integral time. Integrate both sides, and the following can be obtained.

$$M = \frac{K_c}{\tau_I} \int E dt \tag{4.7}$$

If the scan interval is T_s, the discrete form of the equation becomes:

$$\Delta M = K_c \frac{T_s}{\tau_I} E_n \tag{4.8}$$

Derivative Control (D):

Very few industrial controllers use derivative action because of noise [35]. The principle behind derivative action is if the error changes quickly, it can become large in the future. Derivative action tries to prevent this change by changing the output in proportion to the change rate of error *E*:

$$M = K_c \tau_D \frac{dE}{dt} \tag{4.9}$$

τ_D is the derivative time which can be adjusted. The discrete form of the above equation is:

$$M_n = K_c \tau_D \frac{E_n - E_{n-1}}{T_s} \tag{4.10}$$

Previous scan interval is:

$$M_{n-1} = K_c \tau_D \frac{E_{n-1} - E_{n-2}}{T_s} \tag{4.11}$$

Subtracting (4.10) by (4.11)

$$\Delta M = \frac{K_c \tau_D}{T_s}(E_n - 2E_{n-1} + E_{n-2}) \tag{4.12}$$

For PI control:

Conventional analog form is:

$$M = K_c \left[E + \frac{1}{\tau_I} \int E dt \right] \tag{4.13}$$

Discrete form is:

$$\Delta M = K_c \left[(E_n - E_{n-1}) + \frac{T_s}{\tau_I} E_n \right] \tag{4.14}$$

For PID control:

Conventional analog form is:

$$M = K_c \left[E + \frac{1}{\tau_I} \int E dt + \tau_D \frac{dE}{dt} \right] \tag{4.15}$$

Discrete form is:

$$\Delta M = K_c \left[(E_n - E_{n-1}) + \frac{T_s}{\tau_I} E_n + \frac{\tau_D}{T_s}(E_n - 2E_{n-1} + E_{n-2}) \right] \tag{4.16}$$

4.1.2 Control loops with two actuators

In an NPCTF system, there are a couple of loops which process variables are adjusted by two valves – one mainly to increase the process variable and another to decrease the PV.

For example, valves CV-13 and CV-16 are used to control the water inventory pressure P1, as shown in Figure 4.2. While CV-16 is mainly used to increase the pressure P1 and CV-13 is used to decrease pressure P1, CV-16 can be closed to decrease the pressure P1 and CV-13 can be closed to increase the pressure P1. The question is that, if the pressure P1 is increased, should it increase CV-16 or decrease CV-13? In order to solve this conflict, the opening of one of the valves is fixed and another valve controls the process variable. The two actuator loops and the fixed valves are shown in Table 4.1.

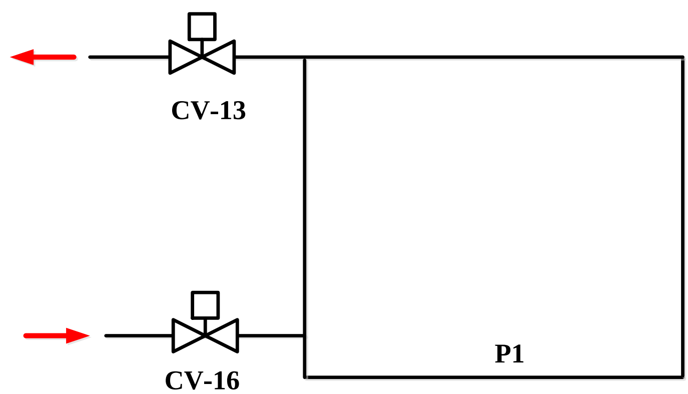

Figure 4.2 Diagram of water inventory control process

Table 4.1 Control loops with two actuators

Pressurizer pressure	CV-9(Main)	P4 (air pressure)
	CV-10(Fixed)	
Pressurizer level	CV-12(Fixed)	L3 (water level)
	CV-16(Main)	
Water pressure	CV-16(Main)	P1 (water pressure)
	CV-13(Fixed)	
Upper Tank level	CV17(Main)	L1 (water level)
	CV15(Fixed)	

4.2 Controller Analysis and Design for Steam Generator Level Control

For the functions of process control in the DCS system, there are 11 loops. This section takes the example of the SG tank level as process controller analysis and design.

4.2.1 Process model

4.2.1.1 Method of constructing model

Model identification is the process of quantifying the process dynamics. The model can be used for simulation or control strategy analysis and design. Different purposes require different accuracy of the models of the real process.

In control engineering, FOPDT (first-order-plus-dead-time) and SOPDT (second-order-plus-dead-time) are used to represent the main features of various process responses. For the model structure, FOPDT and SOPDT are widely used in controller design and controller tuning in various process industries [36]. In this research, a model is created mainly for the purpose of control strategy analysis and design. Therefore, the process can be modelled as first order plus dead time or second order plus dead time. Models constructed here are FOPDT, given by equation 4.17, or SOPDT, given by equation 4.18:

$$G(s) = \frac{K_p}{1+\tau_p s} e^{-\theta s} \tag{4.17}$$

$$G(s) = \frac{K_p}{a_2 s^2 + a_1 s + 1} e^{-\theta s} \tag{4.18}$$

where K_p is process static gain; τ_p is time constant; and θ is process delay.

The techniques available can be categorized into two approaches, open loop test and close loop test. For the open loop test, the controller (if existing) is put into manual mode and the disturbance is applied to the process through changing the output of the controller (*M*). The response of the system is then measured. If the controller exists and provides some level of control, the close loop test can be used to identify the process model. This test is performed by putting the controller in Auto mode and changing the set-point of the controller.

For this research, all the models are identified by the open loop test since the system is new and no previous controller existed. The MATALB System Identification Tool is used to identify the process model.

The identification of a process model involves three basic steps [37].

1. The data record. In this step, it should be decided what variables are to be measured and what variables are to be applied to the process. How much data should be recorded?

What is the sampling time? The result is the input-output data for model developed. This step is illustrated by “Experiment Design” and “Data” in Figure 4.3.

2. The set of models or the model structure. This step is the most important. The model structure should be combined with the purpose of the model. If there is information about the model structure and physical process, this information should be combined to construct the model. The parameters are adjusted to reflect physical consideration. This model set is called grey box. If there is no information about the physical process, the parameter is adjusted to fit the data. This model set is called black box. This step includes “Choose Model Set,” “Choose Criterion to Fit,” and “Calculate Model” in Figure 4.3.

3. The determination of the “best” model in the set, based on available data. This step assesses the model quality. The basic approach is to compare the data produced by the model with the experiment data. In Figure 4.3, this step includes the “Validate Model.”

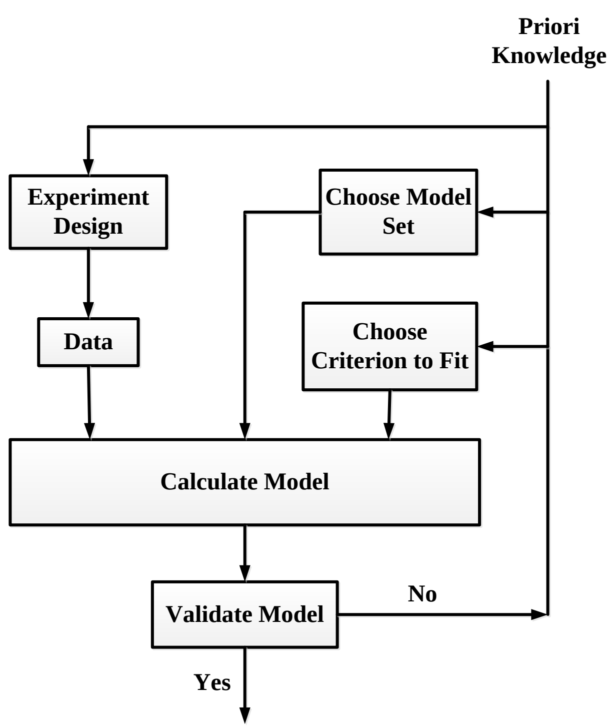

Figure 4.3 Diagram for system identification procedure [37]

In this research, the controller is put into Manual mode and output of the controller (MV) is applied to the process. The process variable is measured. The output of the controller is recorded as input data and the process variable as output data. By this method, the data for modelling is recorded. The MATLAB System Identification tool is used to identify the process model.

4.2.1.2 SG Tank level process model

The level of SG HX Tank (L4) is controlled by valve CV-25, as shown in Figure 4.4. However, CV-25 cannot be used to decrease the tank level. Valve CV-18 is required to decrease the level. Experiments show the opening of 40% of CV-18 is the best. Therefore, the opening of CV-25 is the input, and tank level L4 is the output of this process model. When the CV-25 opens at 17%, the level is balanced and stable. In order to develop the model of SG HX Tank level control, the CV-25 is open from 17% to 27%, and the level L4 is measured. CV-18 keeps 40% opening as normal running status.

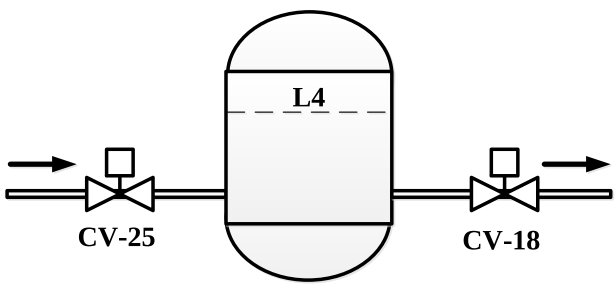

Figure 4.4 Diagram of SG HX Tank level control

The data was drawn from experiment described above, and the relationship between input and output is shown in Figure 4.5. From the Figure 4.5, the response has time delay.

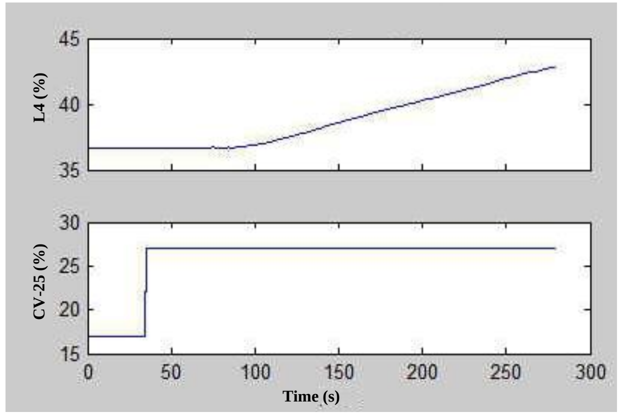

Figure 4.5 Input and output curve from experiment data

The MATLAB System Identification Tool and the PEM (Prediction Error Methods) are used to estimate the model. It is an SOPDT (second-order-plus-dead-time) system. The model is as following:

$$G(s) = \frac{0.007272}{s(1+29.883s)} e^{-18s} \tag{4.19}$$

This model has a 94.87% fit to the model. The Final Prediction Error (FPE) is 0.0073, and the Mean Square Error (MSE) is 0.01152. Therefore, the designed model fits the experiment's data.

From Eqn. (4.19), the process has a time delay of 18 seconds. The delay is produced because the control valve CV-25 is small, and the volume of the tank is big. It takes time for level change becoming significant enough for the transmitter to be able to read the change.

4.2.2 Stable analysis

The controller design should guarantee the stability of the system. In order to judge the stability of a system, general stability criterion is: A necessary and sufficient condition for a feedback system to be stable is that all the poles of the system transfer function have negative real parts [38].

Bode Stability Criterion:

Definitions [39]:

Critical Frequency ω_c is defined to be the value of ω for which

$\phi_{OL}(\omega)$ = -180^0. This frequency is also referred to as a phase crossover frequency.

Gain Crossover frequency ω_g is defined to be a value of ω for which $AR_{OL}(\omega) = 1$.

Consider an open-loop transfer function $G_{OL} = G_c G_P$ (as shown in Figure 4.6) that is strictly proper (more poles than zeros) and has no poles located on or to the right of the imaginary axis, with the possible exception of a single pole at the origin. Assume that the open-loop frequency response has only a single critical frequency ω_c and a single gain cross frequency ω_g, then the closed-loop system is stable if amplitude ratio $AR_{OL}(\omega_c) < 1$. Otherwise, it is unstable [39].

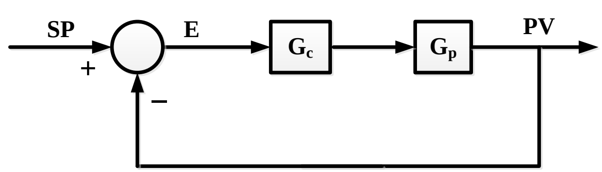

Figure 4.6 Block diagram of closed-loop

4.2.3 Controller parameters tuning using Simple-IMC method

Numerous tuning techniques have been reported in the literature for setting the parameters of a PID controller. In 2000, O'Dwyer did a survey from literature regarding the tuning techniques for PI and PID controllers [40]. For self-regulating process, there were at least 81 methods for tuning PI controller and 117 methods for tuning PID. For integrating processes, at least 22 tuning method for PI controller and 15 for PID controller have been published. However, each one of these methods has at least one flaw [41].

Ziegler Nichols (ZN) method is the basic and one of the most popular tuning techniques mentioned in the literature. However, it is aggressive and easy to cause oscillation. The IMC PID tuning method by Rivera [42] causes poor disturbance response for integrating processes [43] [44]. The Simple Internal Model Control (SIMC) has good performance for both integrating and pure time delay processes and for both setpoints and load disturbances [45].

The SIMC method starts from the IMC PID tuning rules by Rivera [42], and is closely related with direct synthesis tuning rules by Smith and Corripio [46]. It modifies the integral term to improve disturbance rejection for integrating processes.

As indicated in [45], for integrating process with delay and lag with transfer function as following:

$$G_p(s) = K_p \frac{e^{-\theta s}}{s(\tau_2 s + 1)} \tag{4.20}$$

the SIMC method uses the following tuning rule to calculate the parameters for cascade PID (controller G_c) in the closed loop as shown in Figure 4.6:

$$K_c = \frac{1}{K_p(\tau_c + \theta)} \qquad \tau_I = 4(\tau_c + \theta) \qquad \tau_D = \tau_2 \tag{4.21}$$

where K_p is process static gain; τ_2 is time constant; θ is process delay; K_c is gain of controller; τ_I integral time of controller ; and τ_D derivative time of controller.

τ_c is tuning parameter and determined by the trade-off between fast response (small value of τ_c) and stability (large value of τ_c). For fast response, this value is suggested as $\tau_c = \theta$.

For ideal PID, the tuning rule is

$$K_c^{'} = aK_c \qquad \tau_I^{'} = a\tau_I \qquad \tau_D^{'} = \frac{\tau_D}{a} \qquad a = (1 + \frac{\tau_D}{\tau_I}) \tag{4.22}$$

The SIMC tuning rules as indicated in equations (4.21) and (4.22) are used to calculate the parameters of controller of SG Tank level control. For SG Tank level process, as constructed in section 4.2.1, the model is

$$G(s) = \frac{0.007272}{s(1+29.883s)} e^{-18s}$$

For this model, the process gain (K_p), time delay (θ) and time constant (τ_2) are as following:

K_p = 0.007 θ = 18 seconds τ_2 = 29.88 seconds

SMIC tuning technique is used to calculate the parameters as following:

$\tau_c = \theta$ = 18 seconds

Therefore,

$$K_c = \frac{1}{K_p}\frac{1}{\tau_c + \theta} = \frac{1}{0.007}\frac{1}{(18+18)} = 4.0$$

$\tau_I = 4(\tau_c + \theta)$ = 144 seconds

$\tau_D = \tau_2$ = 29.88 seconds

For ideal PID controller, the coefficient is

$$a = (1+\frac{\tau_D}{\tau_I}) = 1+\frac{29.88}{144} = 1.21$$

Therefore, the parameters for ideal PID controller are:

$K_c^{'} = aK_c$ = 1.21x4.0 = 4.84

$\tau_I^{'} = a\tau_I$ = 1.21x144 = 174 seconds

$$\tau_D^{'} = \frac{\tau_D}{a} = \frac{29.88}{1.21} = 24.6 \text{ seconds}$$

After tuned by SIMC with the controller parameters of proportional gain of 4.84, integral time of 174 seconds and derivative time of 24.6 seconds, the Bode diagram is shown in

Figure 4.7. As shown in the Figure 4.7, the gain margin is reduced from 8.31 (18.4dB) to 2.83 (9.04dB) which is between 1.7 and 4 suggested by Seborg [39]. The system is still stable because the gain margin is greater than zero.

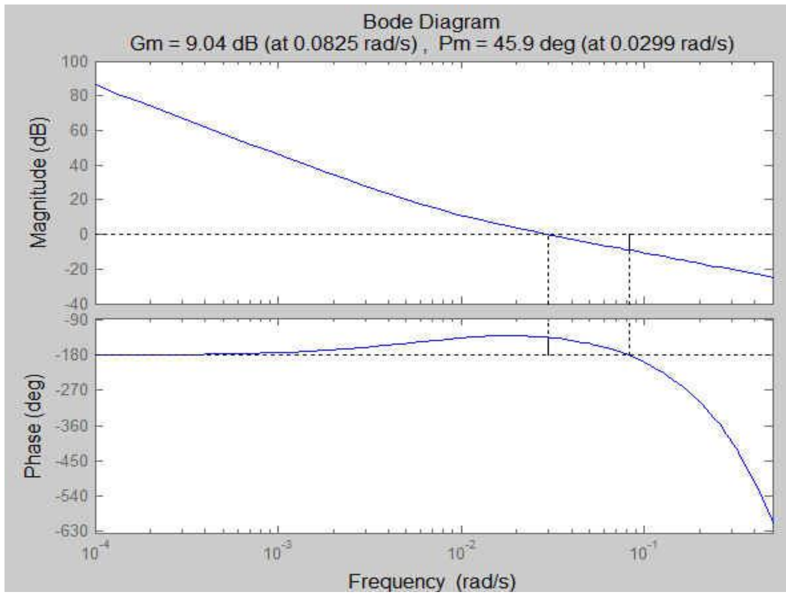

Figure 4.7 Gain and phase margins on a Bode plot

4.2.4 Comparison of tuning results in MATLAB and on NPCTF

In order to compare the tuning results between MATLAB and NPCTF, the control system is built in MATLAB Simulink as shown in Figure 4.8. In the Figure 4.8, the “signal 1” which stands for set-point of the controller is built to change from 30% to 33%, 33% to 36%, and then 36% to 39%, as did in the experiment in NPCTF. Function of “Saturation” is added between PID and process transfer function to limit the opening of valves within 100%. Transfer function created in Section 4.2.1 is used to represent the SG Tank level process. PID is tuned as described in Section 4.2.3.2. Performance from Simulink is shown in Figure 4.9. The same PID parameters are applied in DelatV system for SG Tank level PID controller. To obtain the performance of the SG Tank level controller, its PID controller in the DeltaV system is set in Auto and then change set-point from 30% to 33%, 33% to 36%, and 36% to 39%, respectively. The level performance is shown in Figure

4.10. The overshoot and rise time (time to reach set-point) from MATLAB and NPCTF are calculated in Table 4.2. It can be observed from the Table 4.2 that, the overshoots between results from MALAB and NPCTF are similar. However, with the NPCTF system, it takes more times to reach set-point because of the stiction and hysteresis of the valves [35].

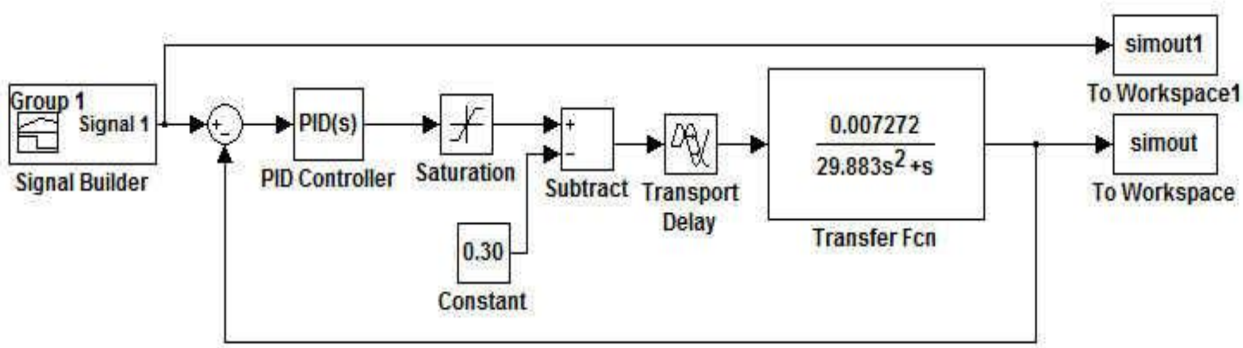

Figure 4.8 Function block diagram of close loop control

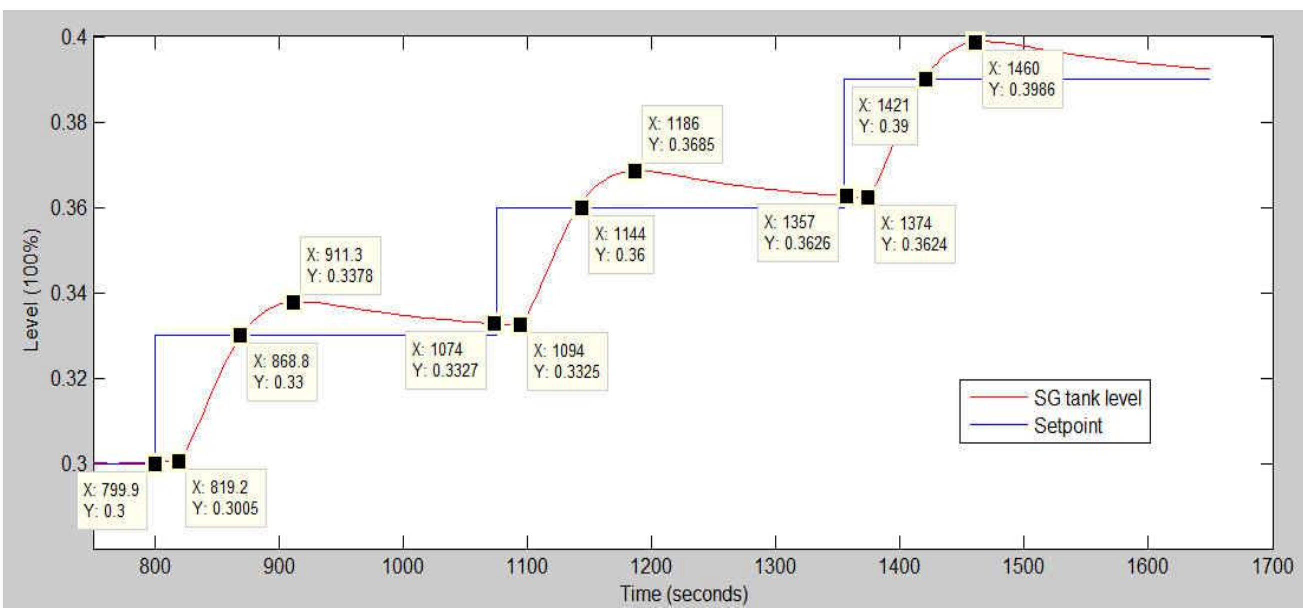

Figure 4.9 SG Tank level responses from Simulink

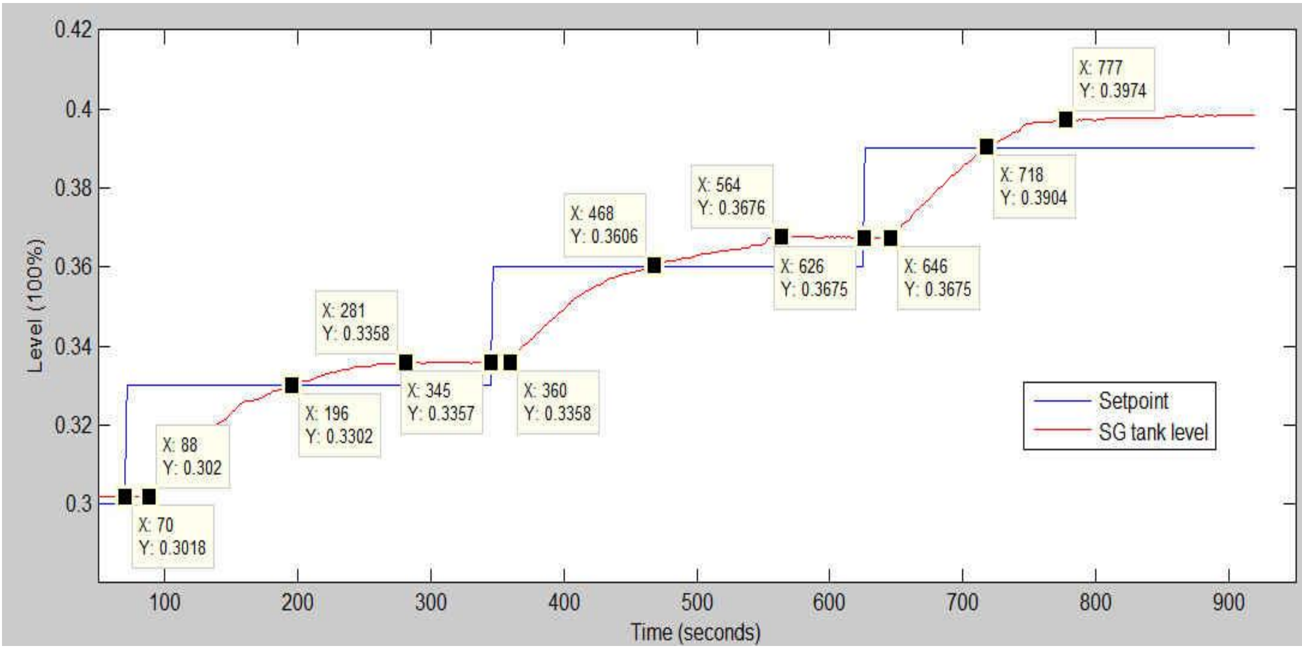

Figure 4.10 SG Tank level responses from NPCTF experiment

Table 4.2 Overshoot and rise time from simulation of MATLAB and NPCTF experiments

	30% - 33%		**33% - 36%**		**36% - 39%**	
	Overshoot (%)	**Rise time (seconds)**	**Overshoot (%)**	**Rise Time (seconds)**	**Overshoot (%)**	**Rise Time (seconds)**
MATLAB	26.40%	50	30.90%	50	31.10%	86
Experiments	20.70%	108	31.40%	108	32.80%	131

4.3 Empirical Tuning Methods for Process Loops

Empirical methods are applied to tune process controllers, except SG Tank level controller discussed in the previous sub-section.

In industry, the process model is often hard to obtain and the controller parameters are generally obtained from a similar process and, thus, adjustment is necessary. Moreover, there is always mismatch between the model and real process. Therefore, the controller parameters, based on the model, are required to be adjusted. The PID parameters are adjusted based on the effect of P (proportional), I (integral), or D (derivative) to the performance of the process controller.

The general rule is that, for proportional only controller, increasing controller gain K_c reduces the rise time and offset at steady state, but demonstrates more oscillation behavior. The Proportional controller always has an offset at steady state. The rule is shown in Figure 4.11.

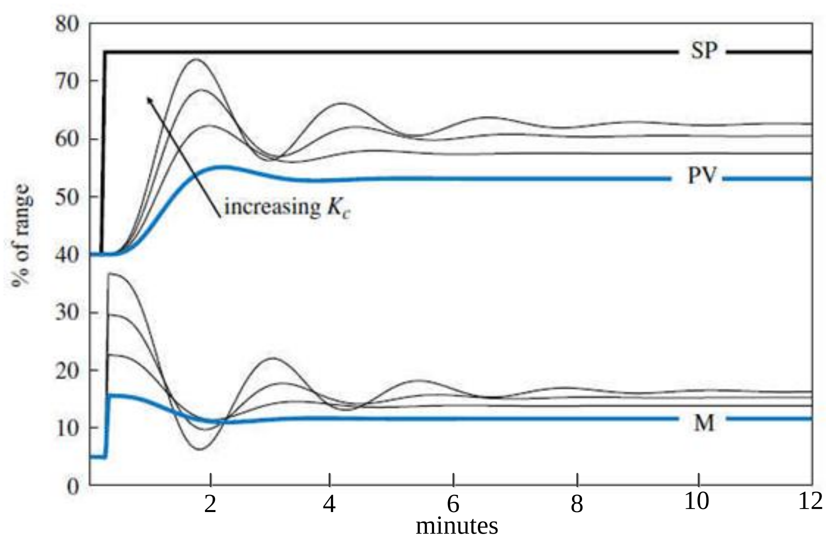

Figure 4.11 Effect of changing K_c [35]

Adding integral action can cancel the offset, but increase the overshoot. More integral action, less integral time T_i, will reduce rise time, but increase overshoot, as shown in Figure 4.12. Less integral action, more integral time T_i, takes a long time to achieve at set point.

In fact, PI controller is adequate for many processes and simplifies the tuning procedure when using empirical method.

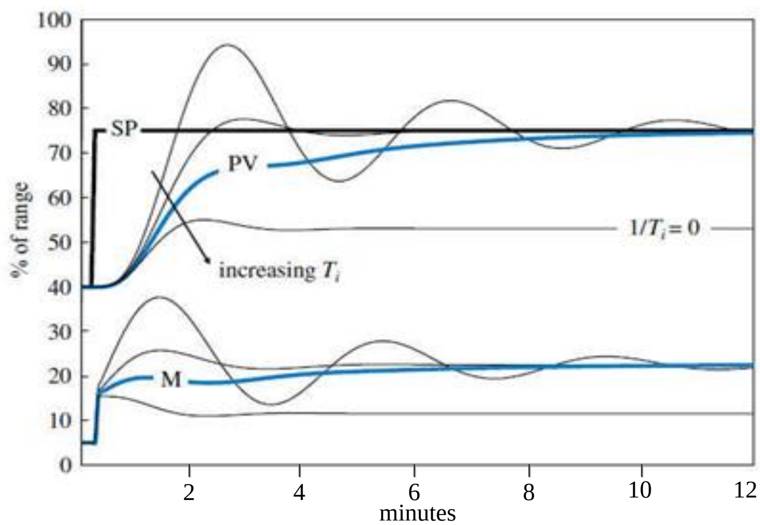

Figure 4.12 Effect of changing T_i [35]

Main function of derivative action is to anticipate the changing trend. For the process with maintaining the rate of change, derivative action can be added in the controller.

Flow process will not maintain the current rate of change for any appreciable time in the future and, thus, derivative action is not recommended in the flow control loop. Temperature process responds slowly and tends to maintain the rate of change, so derivative action is recommended for the temperature control loop [47].

The loops tuned by empirical method and their PID parameters are listed in Table 4.3.

Table 4.3 PID parameters of process controller tuned by empirical method

Loops	**P (Gain)**	**I (Reset: seconds per repeat)**	**D (Rate: seconds)**	**MV**
PRESSURIZER_L	4.0	100.0	15.0	CV-16
PRESSURIZER_P	1.3	21.3	0.0	CV-9
PRIMARY_WATER_F	0.6	16.8	0.0	CV-1
SG_HX_TANK_P	0.8	45.0	20.0	CV-21
UPPER_TANK_L	3.0	250.0	10.0	CV-17
WATER_HEATING	3.0	30.0	0.0	C2
WATER_INVENT_P	1.36	65.0	16.4	CV-16

4.4 Summary

In brief, this chapter explains the controller algorithm of all process control loops. SG Tank level is given as an example to illustrate the model construction, analysis of the model, and controller design and tuning. The design is compared in MATLAB environment and NPCTF, and shows similar results. Other loops are tuned on empirical method and parameters of the controller are given in this chapter.

Chapter 5

Evaluation and Testing of the DCS Operated NPCTF

The purpose of this chapter is to evaluate if the DeltaV system designed can perform the functions described in Chapter 2 which mainly include: process control, sequence control, safety systems and operation modes. The DelatV system is configured and programmed in Chapter 3, and the process controllers are designed in Chapter 4. For safety systems, safety conditions are created and checked if the safety systems in the DelatV system can respond as described in Chapter 2. In operation modes, different operation modes are selected and examined if the related actions, such as the opening or closing of the valves, are performed as described in Chapter 2. Sequence control is involved during the start-up of the system. During the start-up of the system, related tanks are inspected if they are filled up sequentially. For process control, the set-points of controllers are changed from different operating points and checked if there are oscillation for the responses and there are too many overshoots.

5.1 Process Control

In this Chapter, functions such as primary water flow control, water heating control, HX Tank pressure control, Pressurizer pressure control, Pressurizer water level control, Upper Tank level control, and water inventory pressure control are evaluated. The controller is tuned at one operating point and also tested if it works properly at other operating points.

5.1.1 Evaluation methods

In order to evaluate the dynamic performance of a control system in time domain, the following specifications are widely used [38], as shown in Figure 5.1.

In Figure 5.1, percentage overshoot is the amount by which the system output response proceeds beyond the desired response, given by the formula (5.1):

$$P.O. = \frac{M_{pt} - fv}{fv} \times 100\% \tag{5.1}$$

where

$P.O.$ denotes percentage overshoot;

M_{pt} denotes peak value of the time response; and

fv denotes the final value of the response.

Rising time is defined as the time for a system to respond to a step input and attain a response equal to a percentage of the magnitude of the input. For under-damped systems, the 0% to 100% rise time is normally used, and for over-damped systems, the 10% to 90% rise time is commonly used [38].

Settling time is the time required for the system output to settle within a certain percentage of the input amplitude, such as 5% of the final value. In this research, 5% is used to evaluate the performance [48].

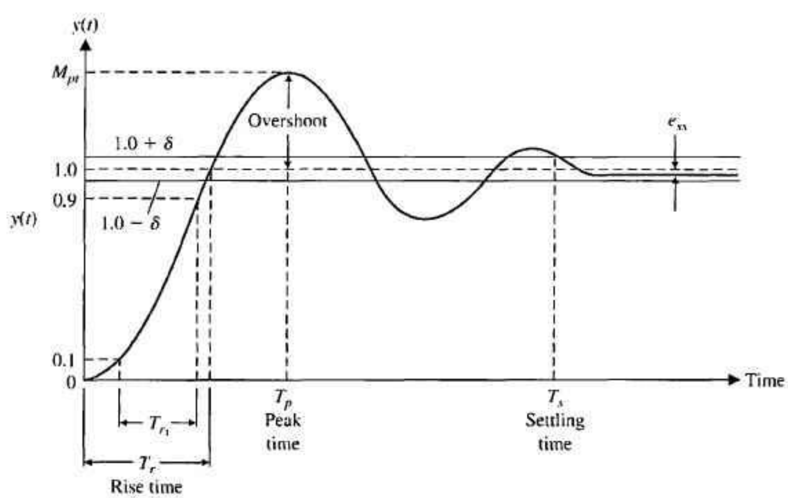

Figure 5.1 Performance specifications in time domain

5.1.2 Primary water flow control

The evaluation of the primary water flow control is performed by setting the controller of primary water flow to Auto mode, and the set-point is changed from 9 to 6 and then 6 to 9. The response of primary water flow is shown in Figure 5.2 in red line. The blue line shows the change of set-point and the green line shows the output of the controller. It can be observed that the process variable, primary water flow, can track the set-point without oscillation.

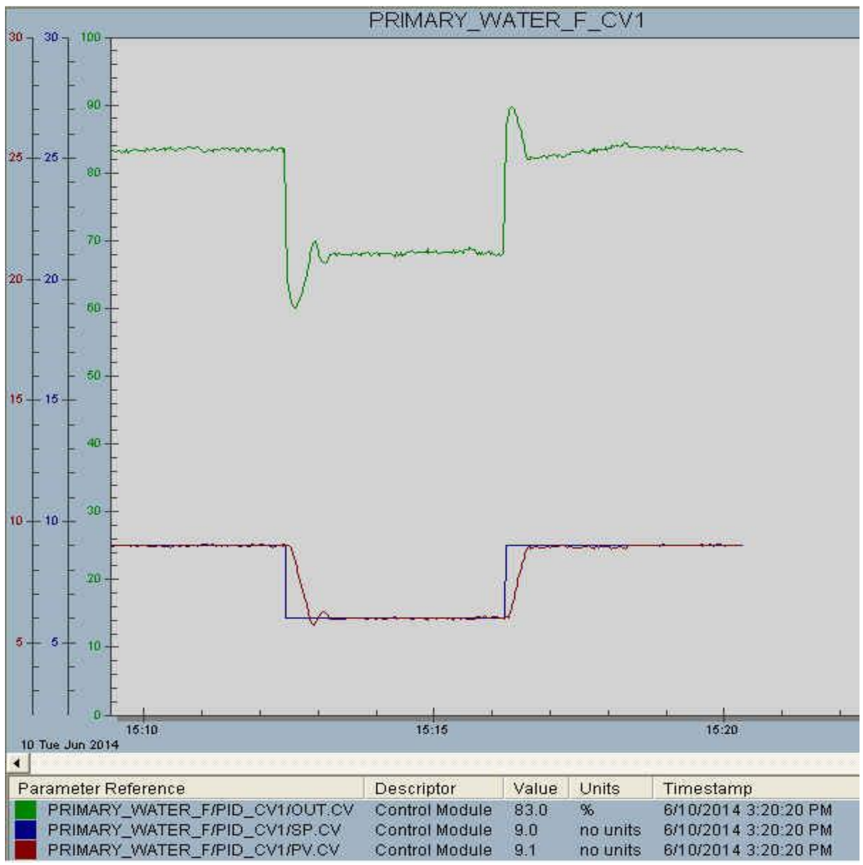

Figure 5.2 Primary water flow performance

When the set-point is dropped from 9 to 6, the response is shown in Figure 5.3. In Figure 5.3, there are four straight red lines which indicate the beginning and final set-point values and error band of the final value. The blue line is the response curve. From Figure

5.3, it can be seen that the response has an overshoot of (6-5.71)/(9-6) = 9.7%, a rising time of (52-30) = 22 seconds and a settling time of (70-30) = 40 seconds.

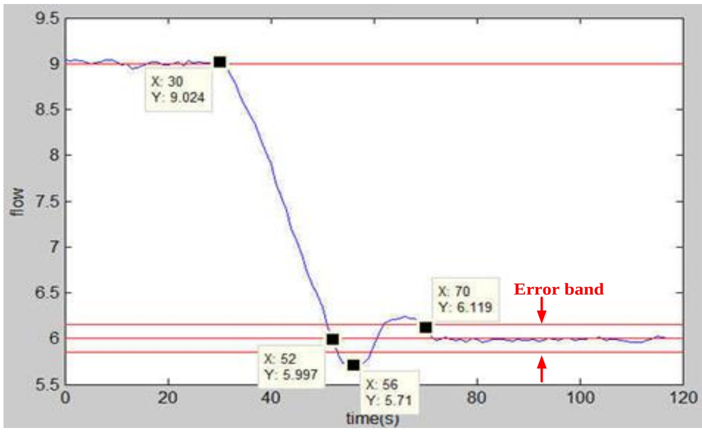

Figure 5.3 Performance characteristics of step response of decrease primary water flow

The increase of the set-point from 6 to 9 shows the response in Figure 5.4. In Figure 5.4, six straight red lines indicate the beginning and final set-point, 10% and 90% total change of set-point, and error band. From the curve, it can be found that the response is over-damped and there is no overshoot. It has rising time of (63-49) = 14 seconds and a settling time of (65-45) = 20 seconds.

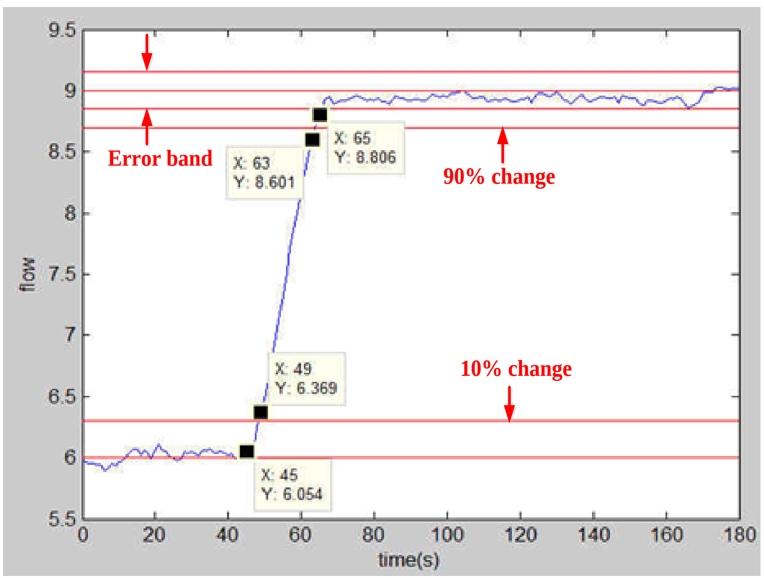

Figure 5.4 Performance characteristics of step response of increase primary water flow

The step response is over-damped to increase the set-point and under-damped to decrease the flow. Moreover, it responds faster when increasing the set-point. Therefore, characteristics of increasing the flow and decreasing the flow are different and, somehow, nonlinear.

5.1.3 Water heater control

In order to evaluate the water heating control, the controller of water heating is set to Auto mode. The set-point is changed from 30℃ to 25℃ and then 25℃ from to 30℃; the responses are shown in Figure 5.5. As indicated at the bottom of Figure 5.5, "Parameter Reference", blue line denotes the temperature changes, green for set-point change, and red line for output of the controller. The process variable and temperature can track the set-point.

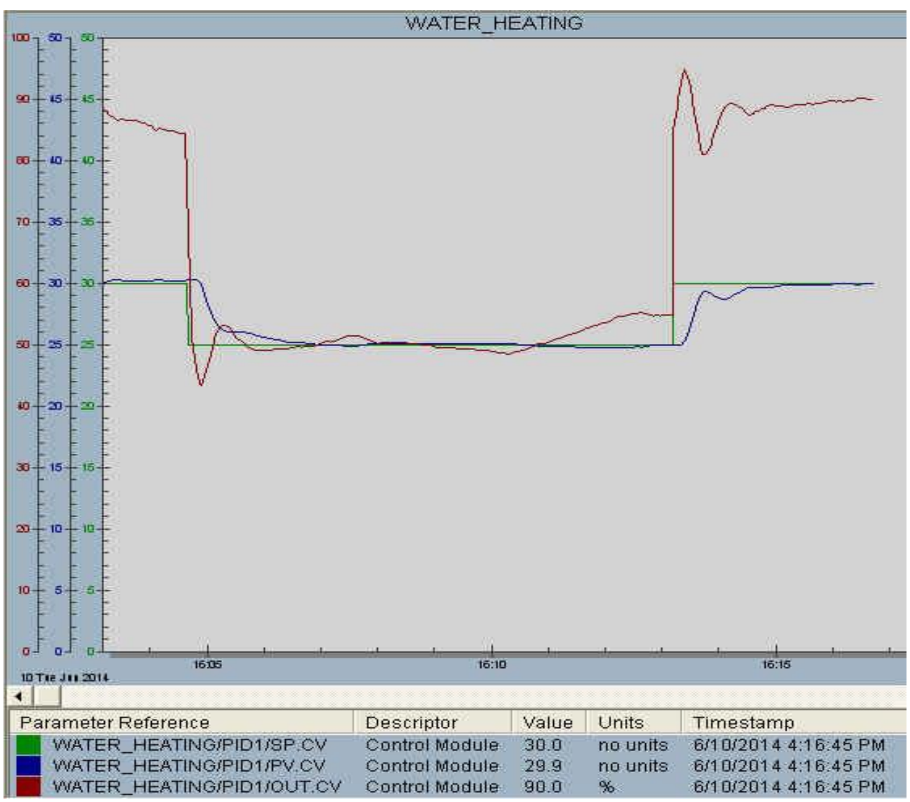

Figure 5.5 Heater outlet temperature performance

When drop heater outlet temperature set-point goes from 30°C to 25°C, the temperature response is shown in Figure 5.6. The blue line shows the temperature response with time. Four straight lines indicate the beginning and final set-point values and error band of the final value. From the curve, it can be seen that the overshoot is (25-24.85)/(30-25) = 3%, rising time is (177-52) = 125 seconds, and settling time is (143-52) = 91 seconds.

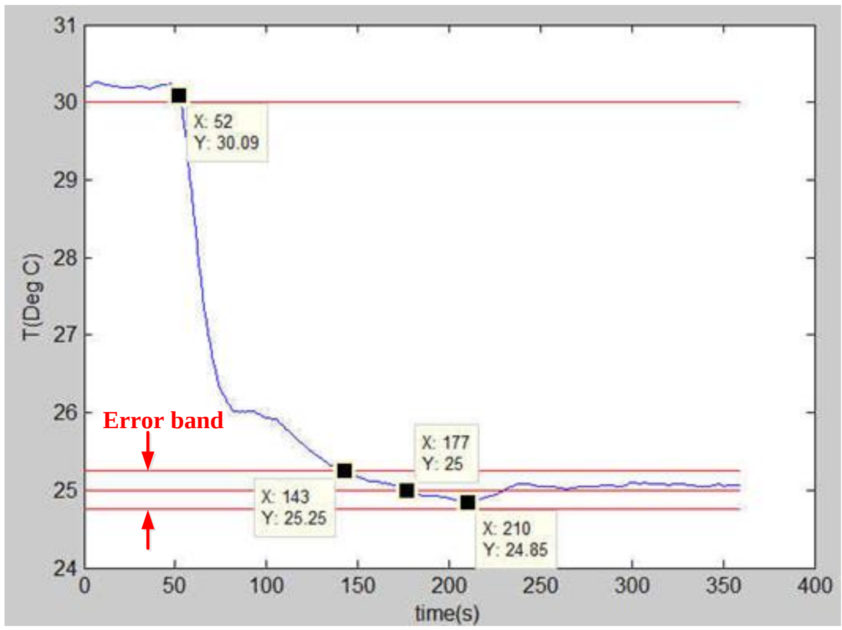

Figure 5.6 Performance characteristics of step response of decrease heater outlet temperature

When the temperature set-point is increased from 25℃ to 30℃, the transient response is shown in Figure 5.7. The rise time is calculated from 10% to 90%. It has a rise time of (148-86) = 62 seconds and a settling time of (185-82) = 103 seconds. The system is a poor damping system and difficult to describe the system using the second-order model giving the oscillatory performance.

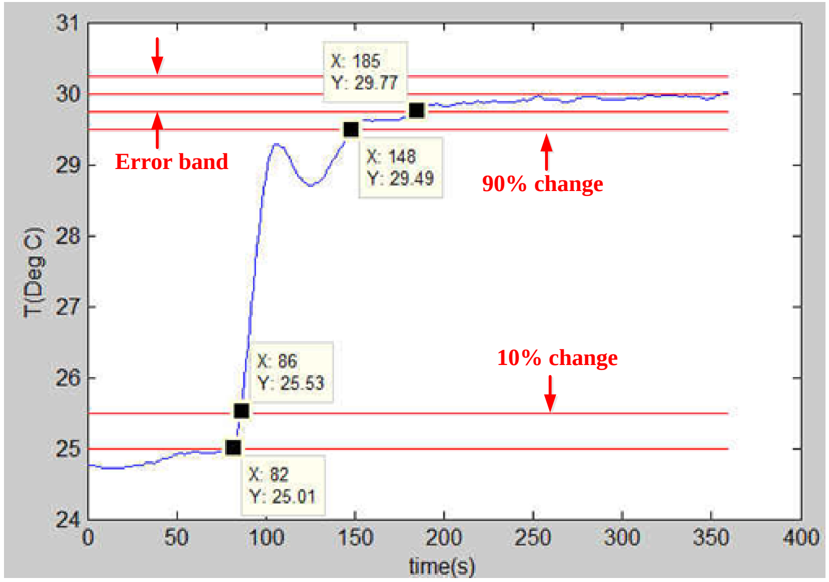

Figure 5.7 Performance characteristics of step response of heater outlet temperature

Comparing the responses between increasing and decreasing the temperature setpoint, it responds faster when increasing the set-point. Therefore, the characteristics of increasing the temperature and decreasing the temperature are different and nonlinear.

5.1.4 HX Tank pressure control

To evaluate the HX tank pressure control, the HX tank pressure controller is set to AUTO and the set-point is changed from 0 to 2psig, 2psig to 3psig, and then to 4psig. The pressure response is shown as a red line in Figure 5.8. The blue line and green line show the set-point and controller output respectively. From the Figure 5.8, the pressure tracks the set-point better from 0psig to 2psig and worse from 3psig to 4psig.

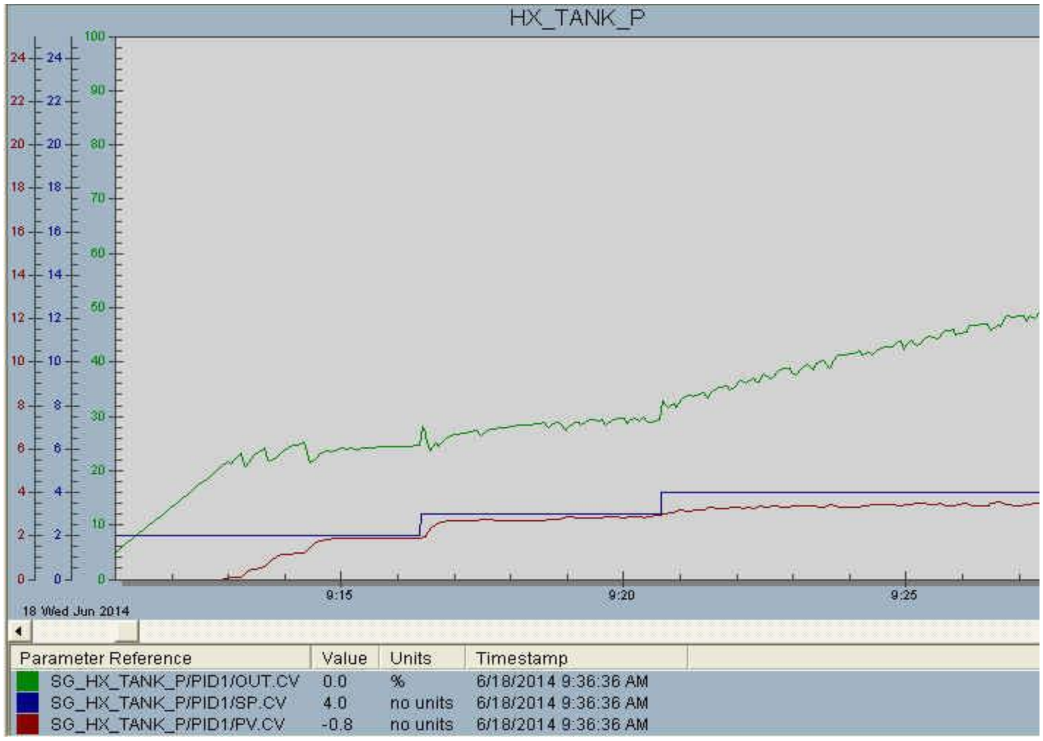

Figure 5.8 HX tank pressure performance

When the set-point of tank pressure is changed from 0psig to 2 psig, the pressure response is shown as a blue line, as in Figure 5.9. . Its response is over-damped, and the rise time is calculated from 10% to 90%. It has an offset which is (2-1.866)/2 = 7.25% and rise time of (104-17) = 87 seconds.

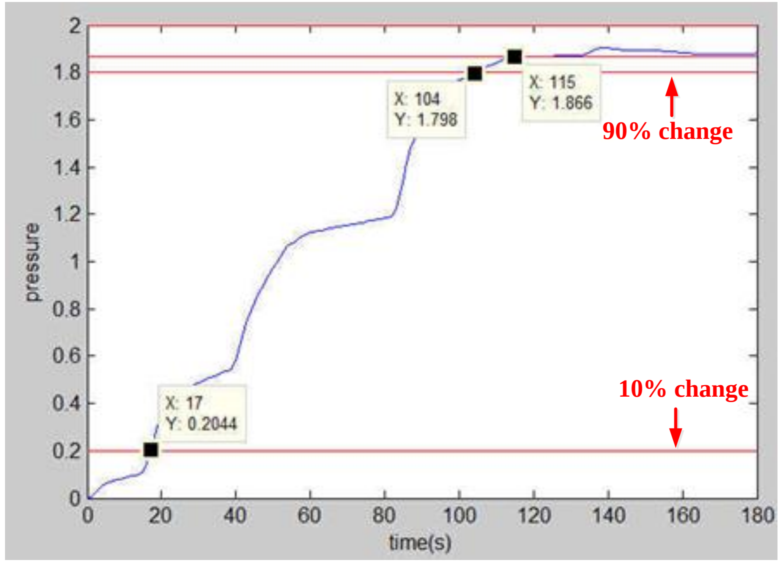

Figure 5.9 Performance characteristics of step response of HX tank pressure (0psig-2psig)

The response from 2psig to 3psig is shown in Figure 5.10. It has a bigger offset which is (3-2.8)/(3-2) = 20%. It has a settling time of (176-31) = 145 seconds when a 20% offset is acceptable.

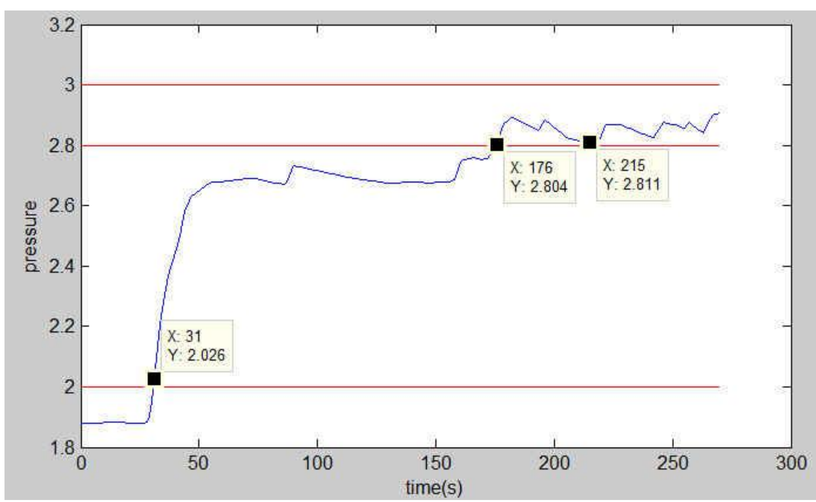

Figure 5.10 Performance characteristics of step response of HX Tank pressure (2psig-3psig)

The response from 3psig to 4psig is shown in Figure 5.11. It has a bigger offset which is (4-3.36)/(4-3) = 65%.

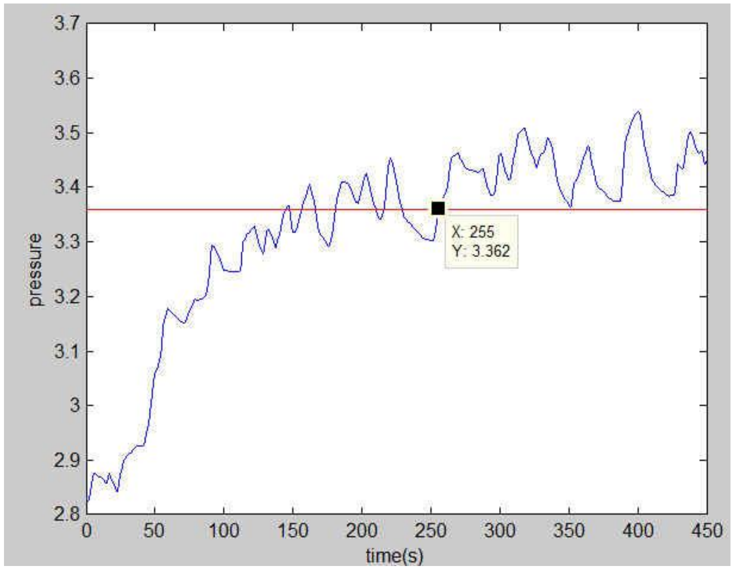

Figure 5.11 Performance characteristics of step response of HX Tank pressure (3psig-4psig)

Therefore, the step response from 0psig to 2psig, 2psig to 3psig, and 3psig to 4psig is different and the process is nonlinear. The controller can control the HX Tank pressure.

5.1.5 Pressurizer pressure control

In order to assess the Pressurizer pressure control, the Pressurizer pressure controller is set in AUTO and its set-point is changed from 2psig to 3psig, 3psig to 4psig, and 4psig to 5psig, respectively. The pressure response is shown as a red line in Figure 5.12. The blue line indicates the set-point and a green line for the output of the controller. From the curve, the pressure of the pressurizer can tack the set-point without oscillation.

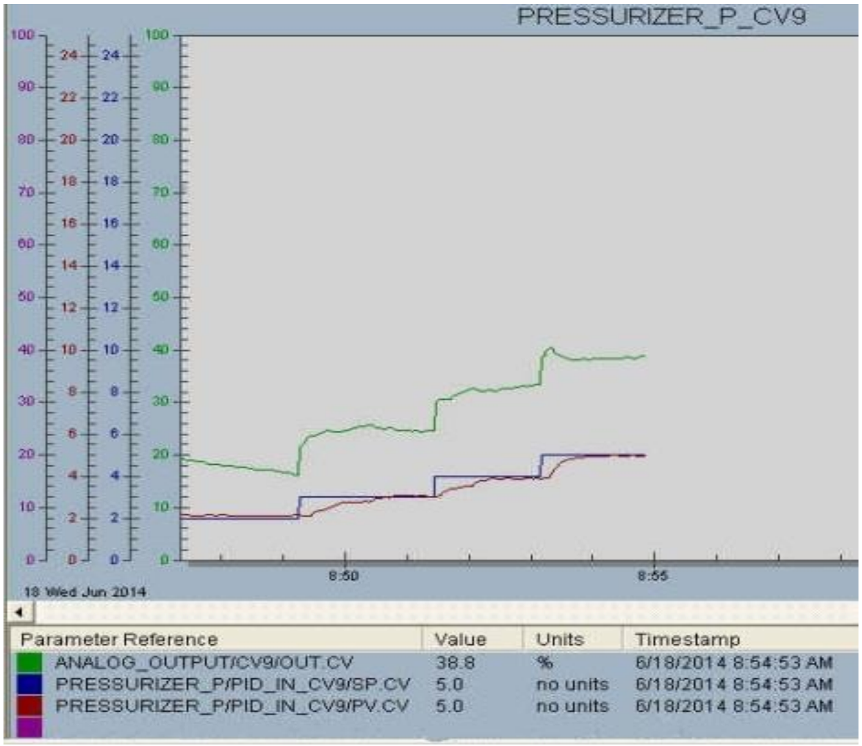

Figure 5.12 Pressurizer pressure control performance

When increasing the set-point of the pressure of the pressurizer from 2psig to 3psig, the response is shown in Figure 5.13. The response has an overshoot of (3.066-3)/(3-2.108) = 7.4% and a rise time of (138-55) = 83 seconds.

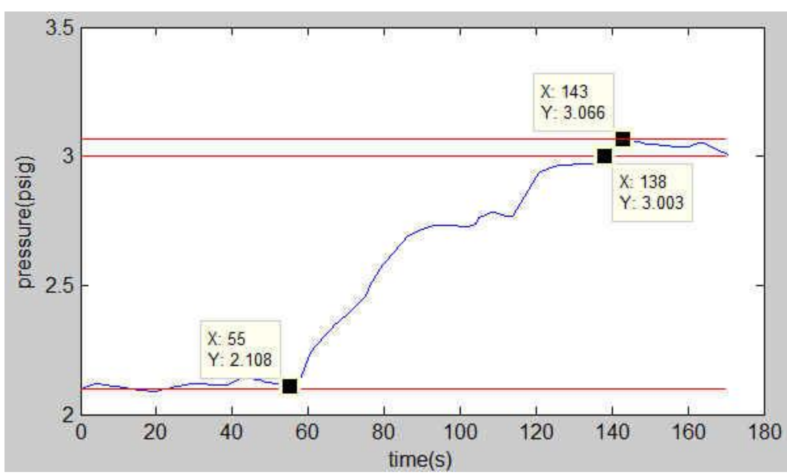

Figure 5.13 Performance characteristics of step response of pressurizer pressure (2psig-3psig)

If the set-point is changed from 3psig to 4psig, its response is shown in Figure 5.14. It is over-damped and has an offset of (4-3.907)/(4-3.008) = 9.4% and rise time of (117-34) = 83 seconds.

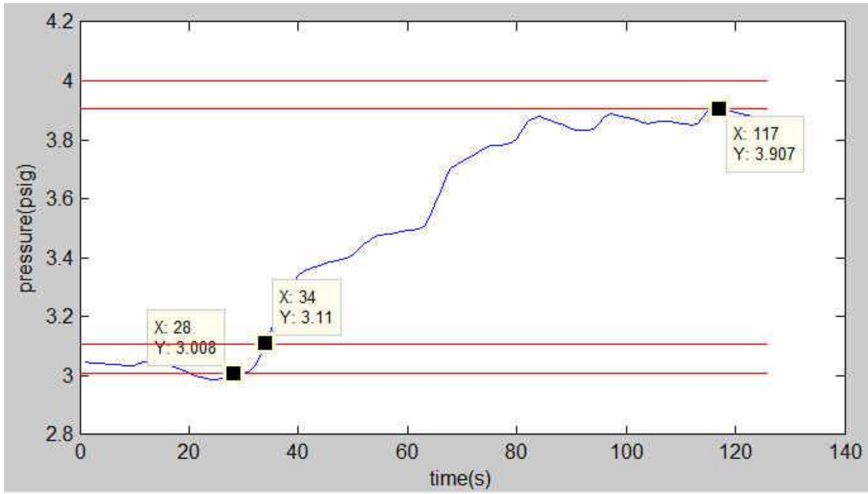

Figure 5.14 Performance characteristics of step response of pressurizer pressure (3psig-4psig)

The response in Figure 5.15 shows an increase in the set point from 4psig to 5psig. It is over-damped and has an offset of (5-4.975)/(5-3.911) = 2.3% and rise time of (42-25) = 17 seconds.

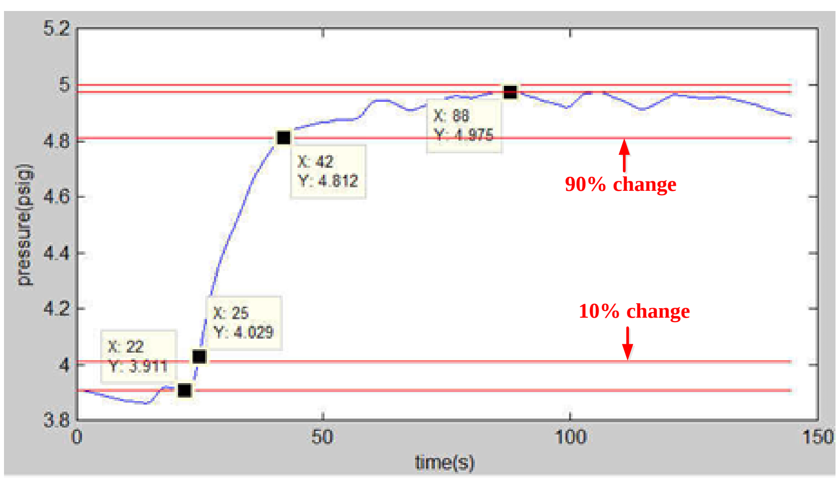

Figure 5.15 Performance characteristics of step response of pressurizer pressure (4psig-5psig)

Therefore, the response from 2psig to 3psig is under-damped and 3psig to 4psig and 4psig to 5psig are over-damped. The process is nonlinear. The controller can control the pressure of the Pressurizer.

5.1.6 Pressurizer water level control

To evaluate the Pressurizer level control, the Pressurizer water level controller is set to AUTO and its set-point is changed from 42% to 45%, then from 45% to 50%. Its response is shown in Figure 5.16. The red line, blue line, and pink line indicate pressurizer pressure, set-point, and controller output respectively.

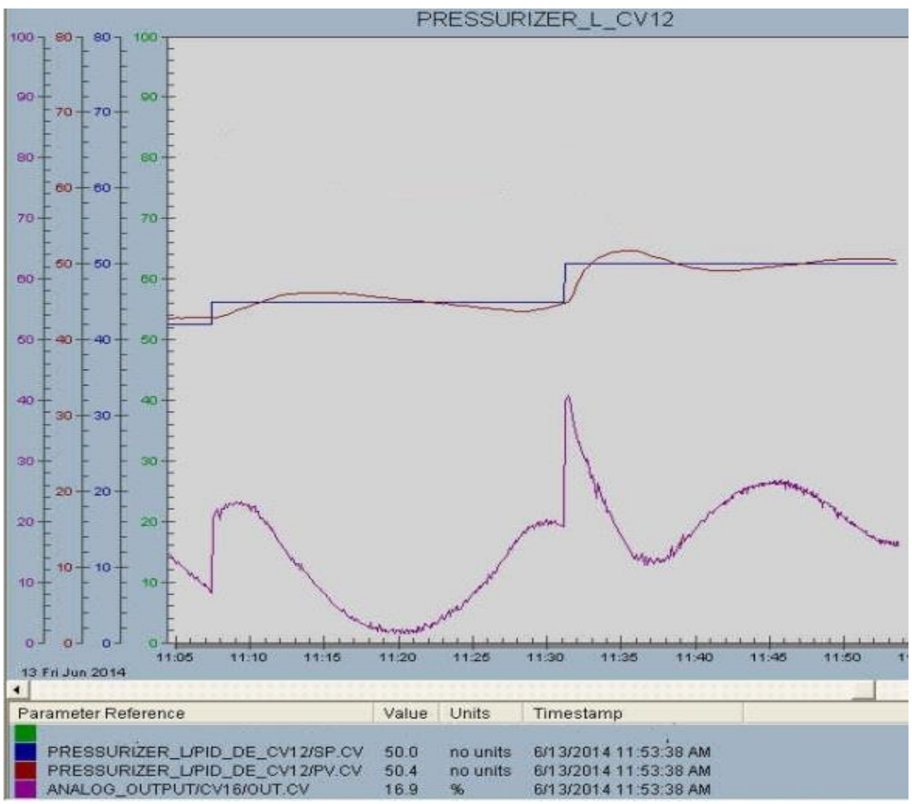

Figure 5.16 Pressurizer water level control performance

When the set-point of the level is increased from 42% to 45%, in fact 42.85% to 45%, it takes (397-59) = 338 seconds to reach the maximum. It has an overshoot of (46.13-45)/(45-42.85) = 52.6% and a rise time of (245-59) = 185 seconds.

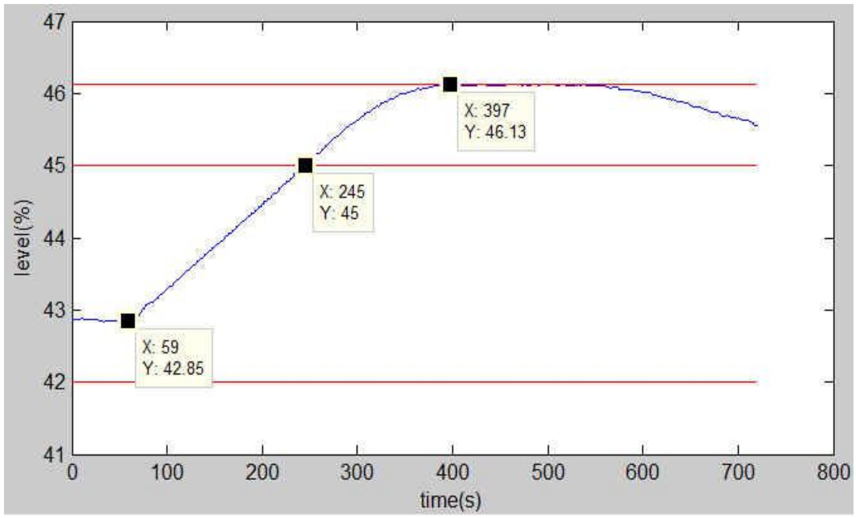

Figure 5.17 Performance characteristics of step response of pressurizer level (42%-45%)

When the set-point of the level is increased from 45% to 50%, it takes 289-42=247 seconds to reach the maximum. It has an overshoot of (51.77-50)/(50-45) = 35.4% and a rise time of 132-42 = 90 seconds.

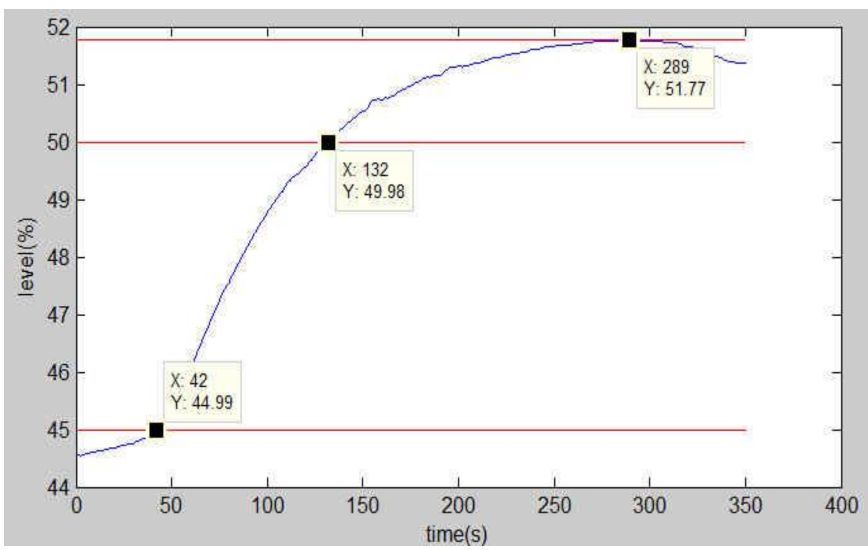

Figure 5.18 Performance characteristics of step response of pressurizer level (45%-50%)

In conclusion, the response from 42% to 45% and 45% to 50% are different and, thus, nonlinear. The controller can control the process without oscillation.

5.1.7 Upper Tank level control

In order to assess the Upper Tank level control, the Upper Tank level control is set to AUTO and its set-point is changed from 30% to 35%, 35% to 40%, and then 40% to 45%. The response is shown in Figure 5.19. The red line, blue line and green line indicate the tank level, controller set-point and output, respectively.

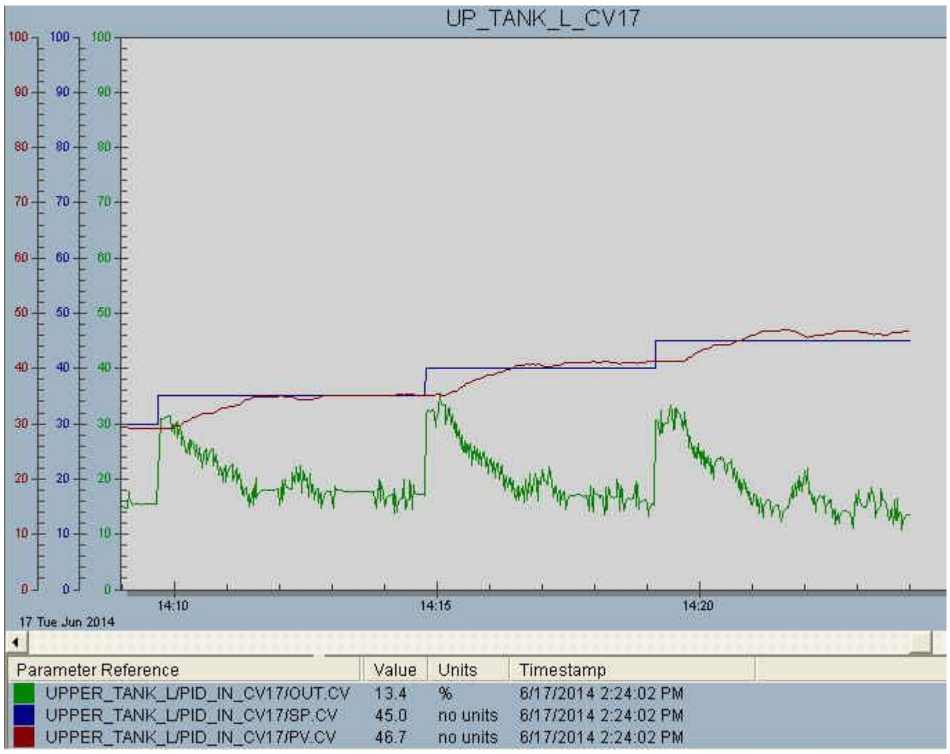

Figure 5.19 Upper tank level control performance

When the set-point of the level is increased from 30% to 35%, the response is shown in Figure 5.20. It takes (210-54) = 156 seconds to reach the maximum. It does not have an overshoot and rise time of (126-54) = 72 seconds.

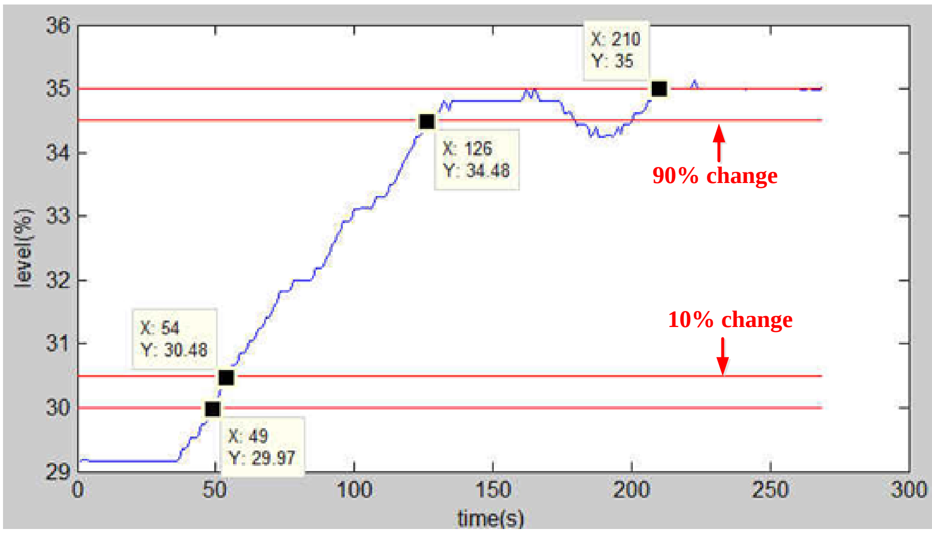

Figure 5.20 Upper Tank level step response from 30% to 35%

When the set-point of the level is increased from 35% to 40%, the response is shown in Figure 5.21. It takes (189-49) = 140 seconds to reach the maximum. It has an overshoot of (41.03-40)/(40-35) = 20.6% and a rise time of (125-49) = 76 seconds.

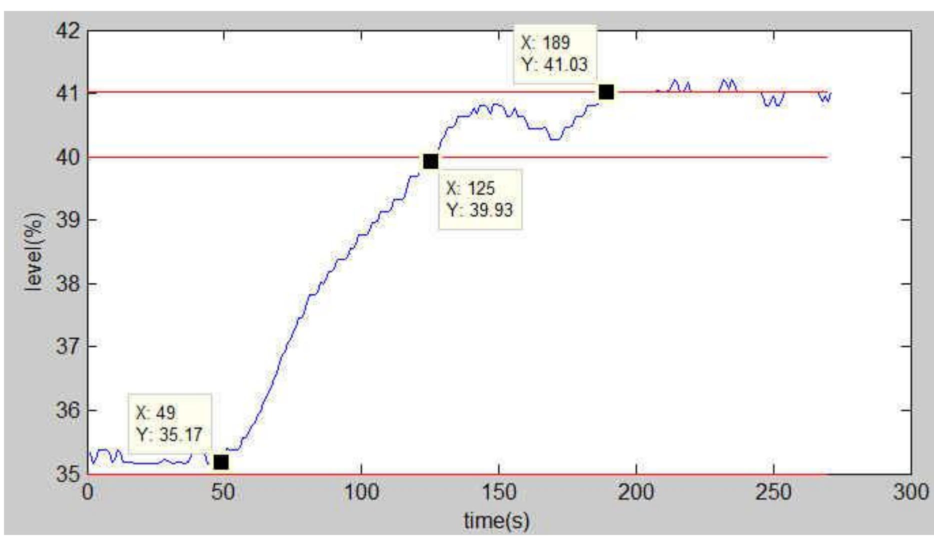

Figure 5.21 Upper tank level step response from 35% to 40%

When the set-point of the level is increased from 40% to 45%, the response is shown in Figure 5.22. It takes (189-49) = 116 seconds to reach the maximum. It has an overshoot of (47.04-45)/(45-41.21) = 53.8% and a rise time of (117-51) = 66 seconds. The overshoot at an operating point of 40% is high since the Upper Tank is nonlinear tank and PID parameters are tuned to an operating point of 30%.

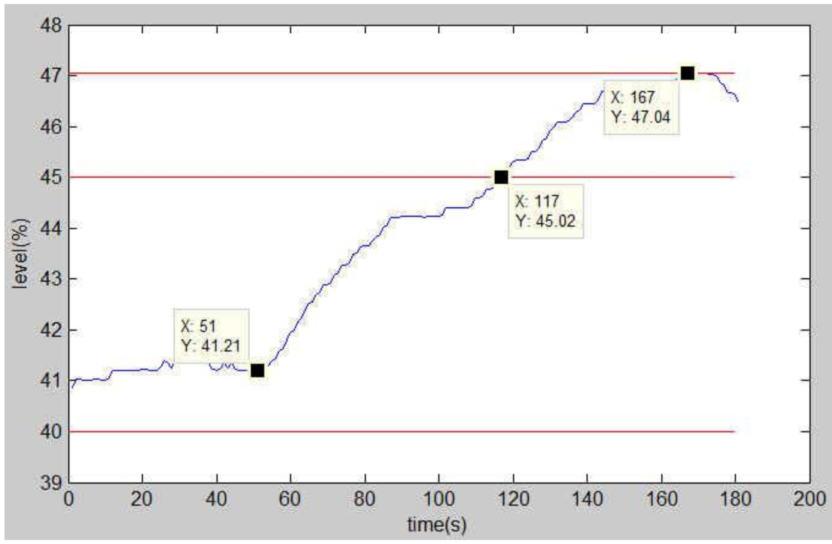

Figure 5.22 Upper tank level step response from 40% to 45%

In conclusion, the responses from 30% to 35%, 35% to 40%, and 40% to 45% are different and nonlinear. The figures show that the controller can control the process without oscillation.

5.1.8 Water inventory pressure control

To evaluate the water inventory pressure control, the water inventory pressure controller is set to AUTO and its set-point is changed from 3psig to 5psig. The response is shown in Figure 5.23. The brown line, blue line, and pink line indicate pressure, controller set-point, and controller output, respectively.

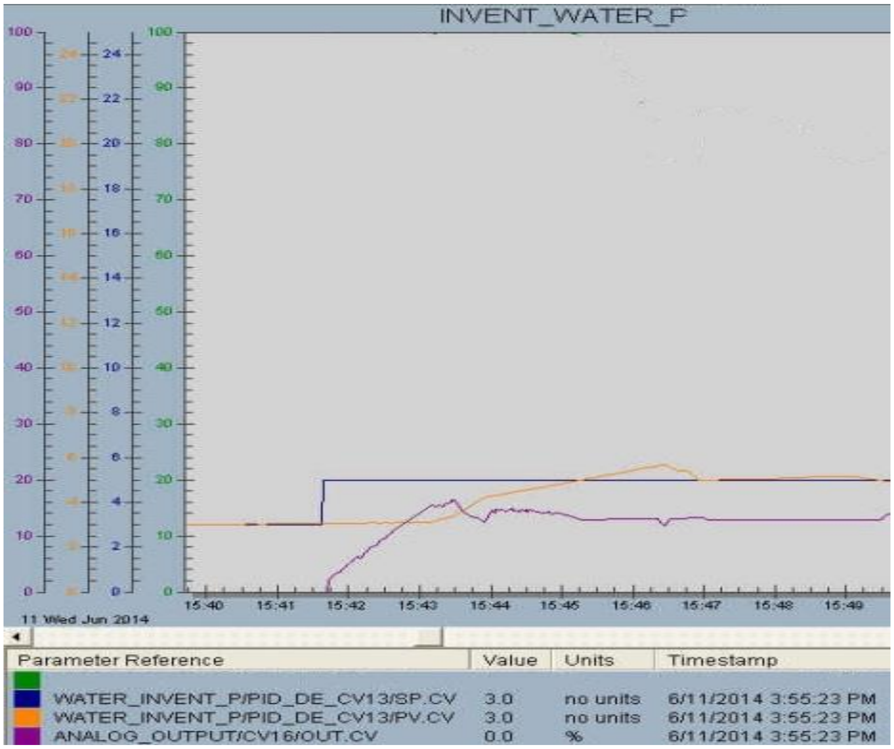

Figure 5.23 Primary water pressure P1 step response

When the set-point of the pressure controller is increased from 3psig to 5psig, its response is shown in Figure 5.24. It takes (296-125) = 171 seconds to reach the maximum. It has an overshoot of (5.715-5)/(5-3.076) = 37.2% and a rise time of (247-125) = 122 seconds. Therefore the overshoot is high, but only takes 2 minutes to reach the set-point. The controller can control the pressure.

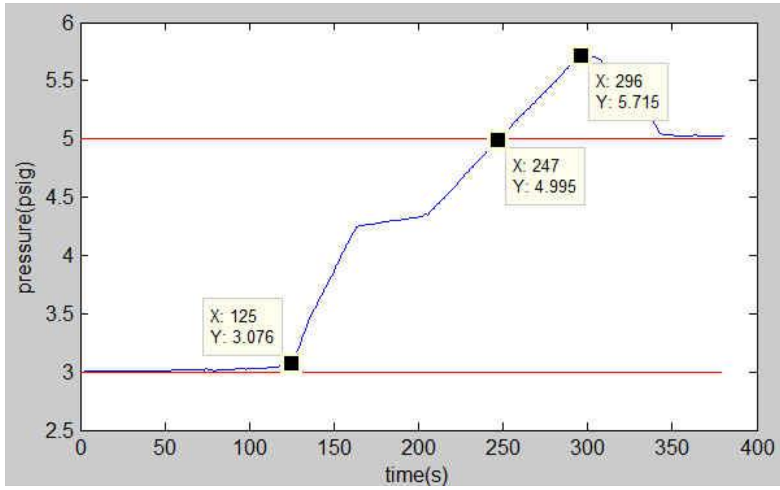

Figure 5.24 Primary water pressure P1 step response from 3 psig to 5 psig

In conclusion, most of the processes are, somehow, nonlinear. The PID tuning that has good performance at one operating point does not have perfect performance at other operating points. Some processes have very nonlinear characteristics and it is better to develop nonlinear control techniques in future research. However, the controllers designed can reasonably control the process.

5.2 Sequence Control

When the "STARTUP" button, as shown in red box in Figure 5.26, is pressed, the system begins the procedure of filling up the tanks to get the system ready for running. The procedure is that the Pressurizer Tank is filled up first and then the HX Tank, and the Upper Tank is the last one to fill up. The level change of each tank is shown in Figure 5.25. As shown in Figure 5.25, the pressurizer level L3 begins to increase. When the Pressurizer level reaches its designed value, the system stops filling the tank and begins to fill in HX Tank. After the level of the HX Tank reaches its designed value, the fill-up process is stopped and begins to fill in the Upper Tank. However, Figure 5.25 shows the tank's level still increases after stopping and changing to fill in the next tank. From Figure 5.25, it is obvious that before the pressurizer level L3 begins to increase, Upper Tank level L1 increases a little bit and keeps stable. Before starting the filling procedure, valve CV-13 is in NO (normal open) status. When the system begins to fill the first tank -

Pressurizer, it sends an order to close CV-13. However, it takes time for CV-13 to close. During this time, water is fed into the Upper Tank and, thus, its level increases until the CV-13 closes as indicated in Figure 5.26. After the process switches to fill in the HX Tank and valve CV-11 is requested to close, it takes time for CV-11 to close. Water is still fed to Pressurizer and thus Pressurizer level L3 is still increasing, as indicated in Figure 5.26

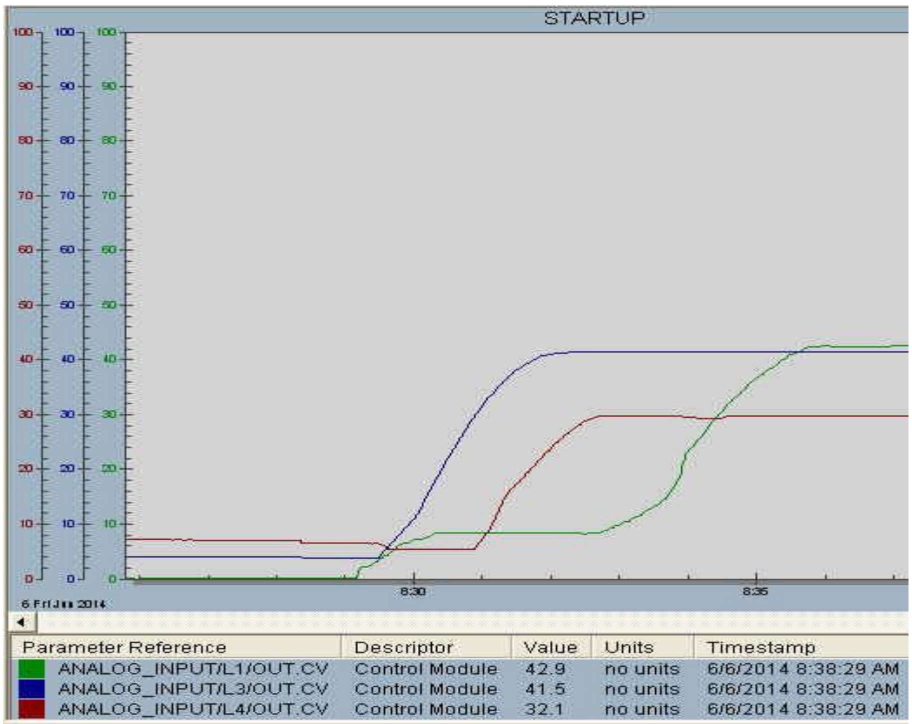

Figure 5.25 Tanks level change during start-up

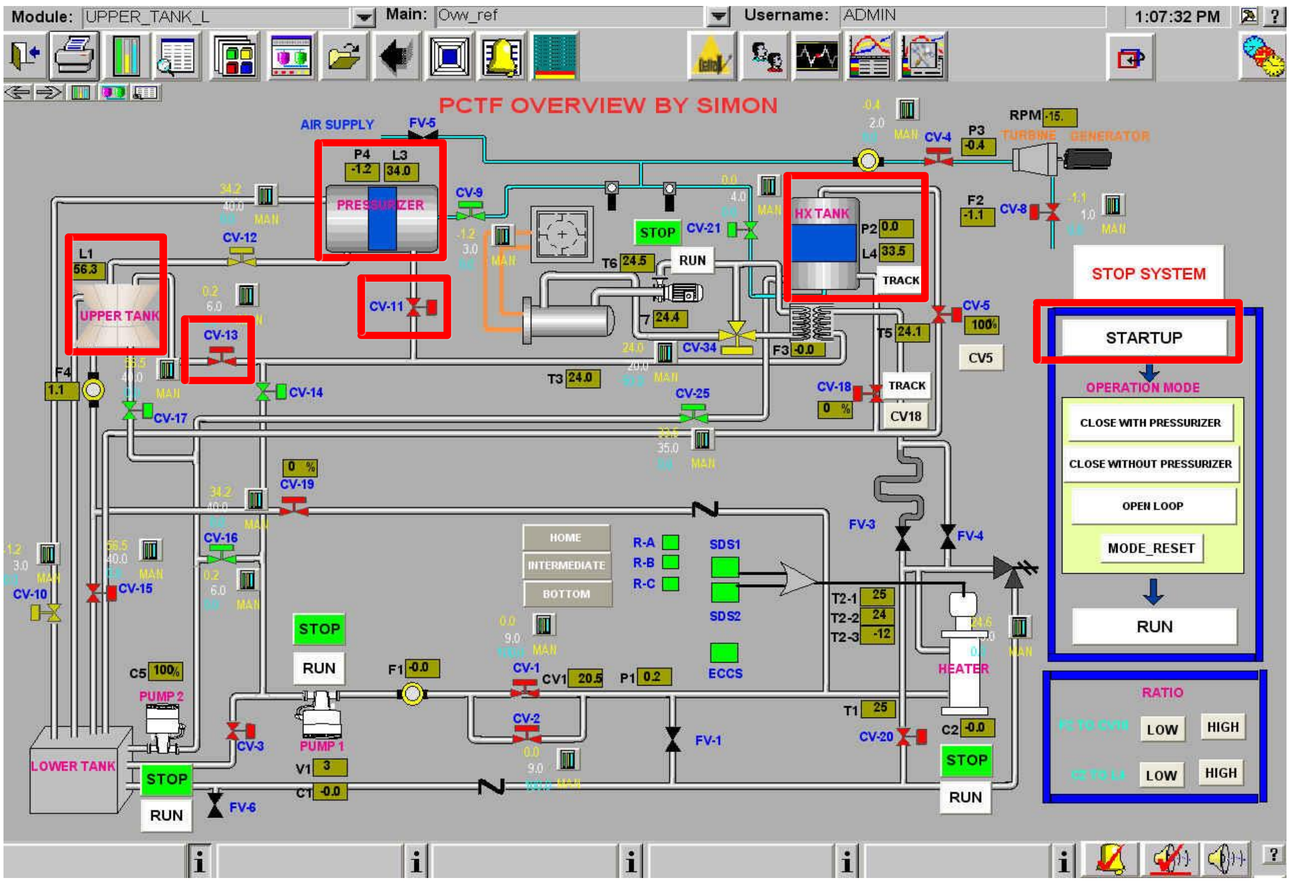

Figure 5.26 NPCTF system diagram during start-up

5.3 Safety Systems Evaluation

As described in chapter 2, this DCS system implements three safety systems: Shut-down System Number 1 (SDS1), Shut-down System Number 2 (SDS2), and Emergency Core Cooling (ECC). This section evaluates these three safety systems.

5.3.1 Shut-down System Number 1 (SDS1)

As discussed in Chapter 2, SDS1 can be activated when RELAY_A (R-A in Figure 5.28) and RELAY_A (R-A in Figure 5.28) are all activated, and RELAY_A and REALY_B all can be activated by the average temperature of the Heater outlet, T2. SDS1 function is tested by increasing the average temperature of T2 to 32 (℃) and 34 (℃) to activate

RELAY_A and RELAY B, respectively. When the average temperature of T2 reaches 32 (℃), the RELAY_A (R-A in Figure 5.28) icon turns red, as shown in Figure 5.28. When T2 arrives at 34 at 10:45:54AM, as shown in Figure 5.27, RELAY_B (R-B in Figure 5.28) is activated and the corresponding icon becomes red, as shown in Figure 5.28. At this time, the condition of SDS1 is satisfied and activated to trip the Heater, and the SDS1 icon becomes red and the Heater icon turns grey, as shown in Figure 5.28. “SAFETY” and time of “10:45:54AM” are also shown at the bottom of Figure 5.28. The Alarm of “SAFETY/SDS1_ALARM” shows the time of “10:45:54AM”, as shown in Figure 5.29. Therefore, the HMI can indicate the alarm of SDS1 and the system activates to stop the Heater when condition of SDS1 is satisfied. Other conditions that can trigger relays of A, B, and C are listed in appendix C.

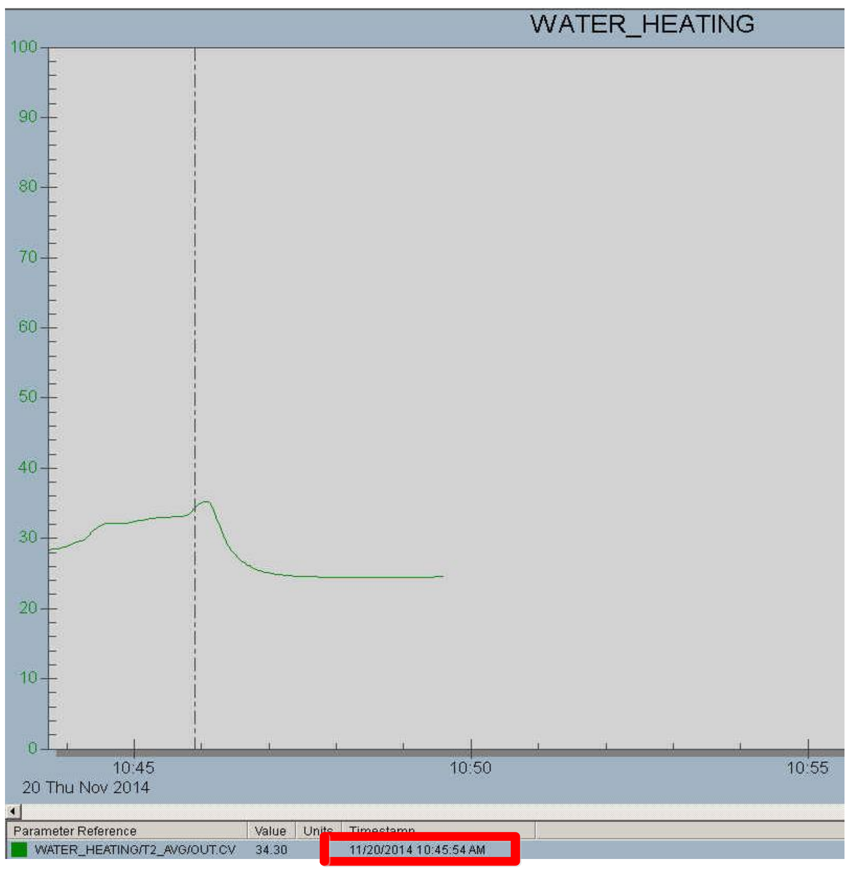

Figure 5.27 T2 average curves for SDS1

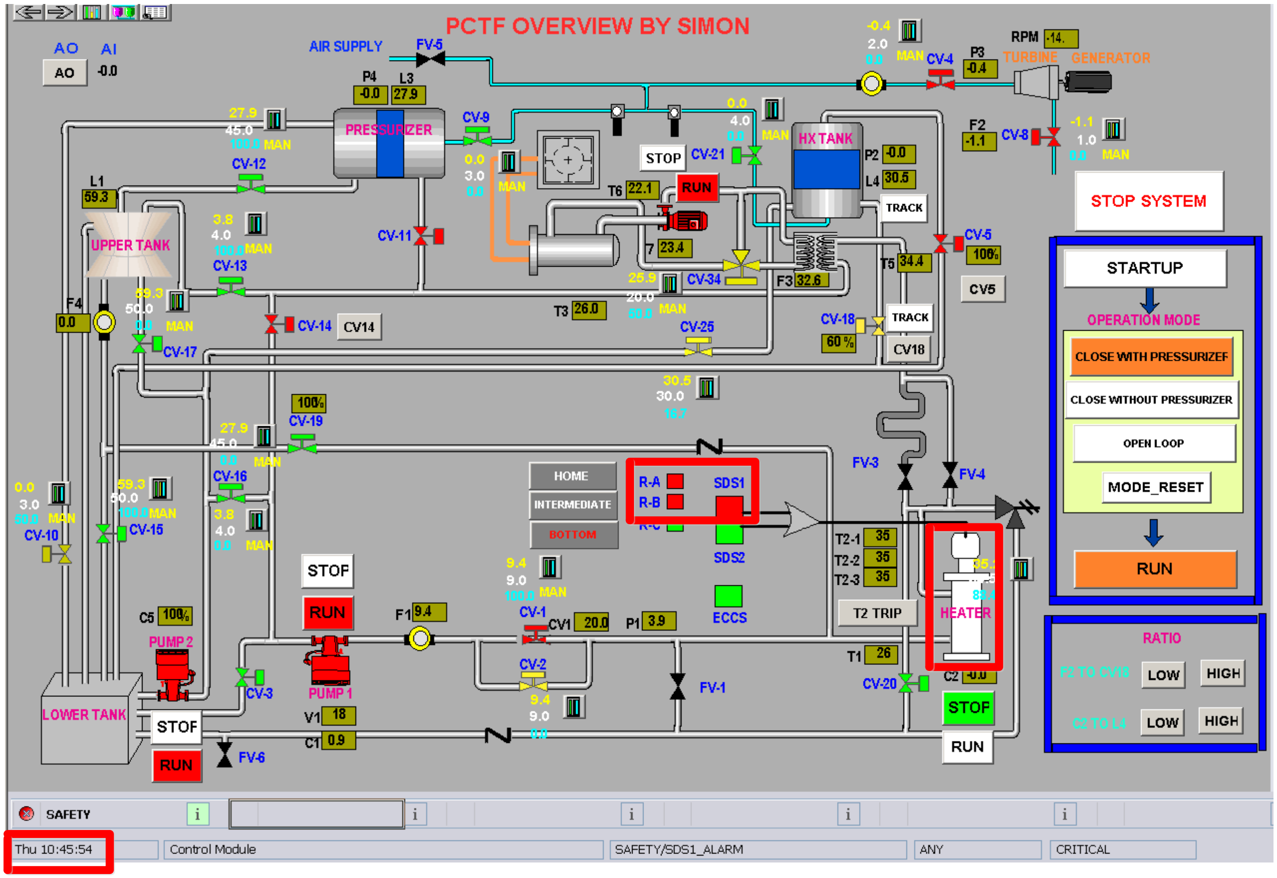

Figure 5.28 NPCTF system diagram for SDS1

Figure 5.29 Alarm of SDS1

5.3.2 Shut-down System Number 2 (SDS2)

Water primary pressure P1 is used to test SDS2 function. Since the maximum pressure after the regulator is 18psig, 17psig is used to test the SDS2 function. Increase the primary water pressure P1 to reach 17psig to activate SDS2. From Figure 5.30, the primary pressure reaches 17psig at 9:01:54AM. In Figure 5.31, HMI shows there is a "SAFETY" alarm at 9:01:54AM at the bottom, the SDS2 icon becomes red and the Heater becomes grey. The alarm of "SAFETY/SDS2_ALARM" at 9:01:54AM is shown in Figure 5.32. Therefore, when the SDS2 condition is satisfied, the HMI indicates the alarm and related action is taken. The Other conditions to trigger SDS2 are listed in appendix C.

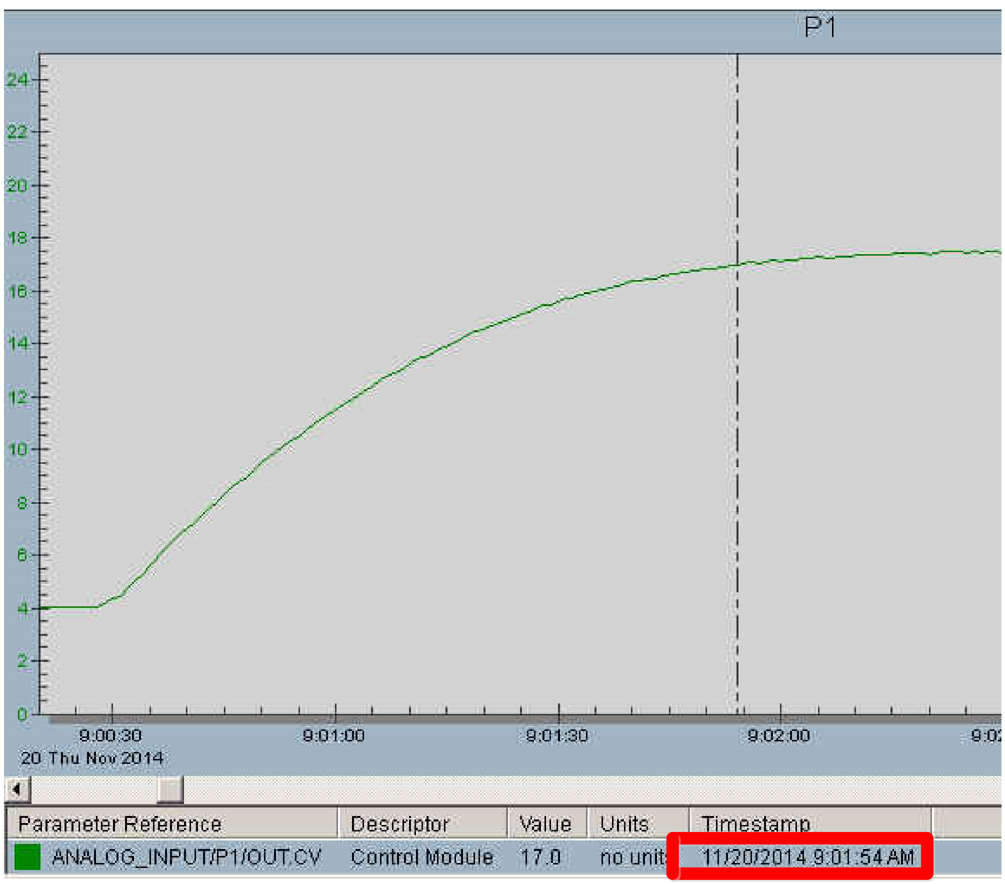

Figure 5.30 Primary water pressure P1 curve for SDS2

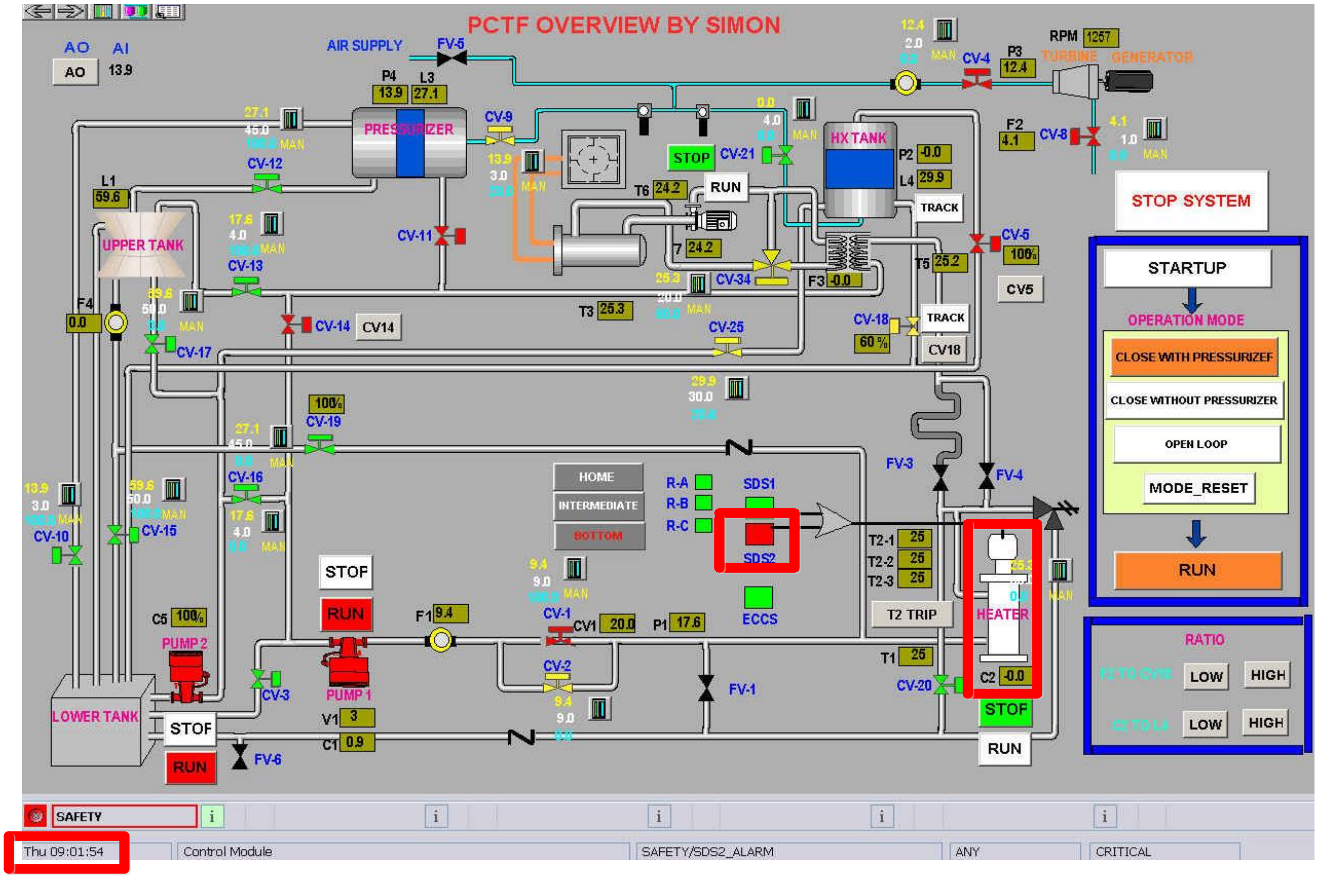

Figure 5.31 NPCTF system diagram for SDS2

Figure 5.32 Alarm of SDS2

5.3.3 Emergency Core Cooling (ECC)

As discussed in section 2.4.3, ECC is activated by two conditions: either SDS1 or SDS2 and Pump 1 off. Therefore, ECC is tested by increasing primary pressure P1 to 17psig to activate SDS2 and stopping Pump 1. The pressure P1 reaches 17psig at around 11:08:12AM, as shown in Figure 5.33. This condition can activate SDS1. In fact, there is a “SAFETY” alarm at the bottom of the HMI, as shown in Figure 5.34. Detail of the alarm is shown in Figure 5.35 as “SAFETY/SDS2_ALARM” at the same time of 11:08:12AM. At 11:08:15AM another condition is met by stopping Pump1. When the two conditions of ECC are satisfied at 11:08:15AM, the alarm of ECC is activated. The alarm is shown in Figure 5.34 as “SAFETY,” and in Figure 5.35 as “SAFETY/ECC_ALARM” at 11:08:15AM. When both conditions are met, the system acts to shut off the Heater which becomes grey and cools the Heater by opening valves CV-19 and CV-20, and closing valves CV-15, CV-1, and CV-2. As shown in Figure 5.34, the color of valves CV-19 and CV-20 becomes red and that of valves CV-15, CV-1, and CV-2 becomes green. When valves CV-19 and CV-20 open, and valves CV-15, CV-1, and CV-2 close, water flows from the Upper Tank to the Heater and goes back to the Lower Tank by natural circling, as shown in Figure 5.34 as a red line.

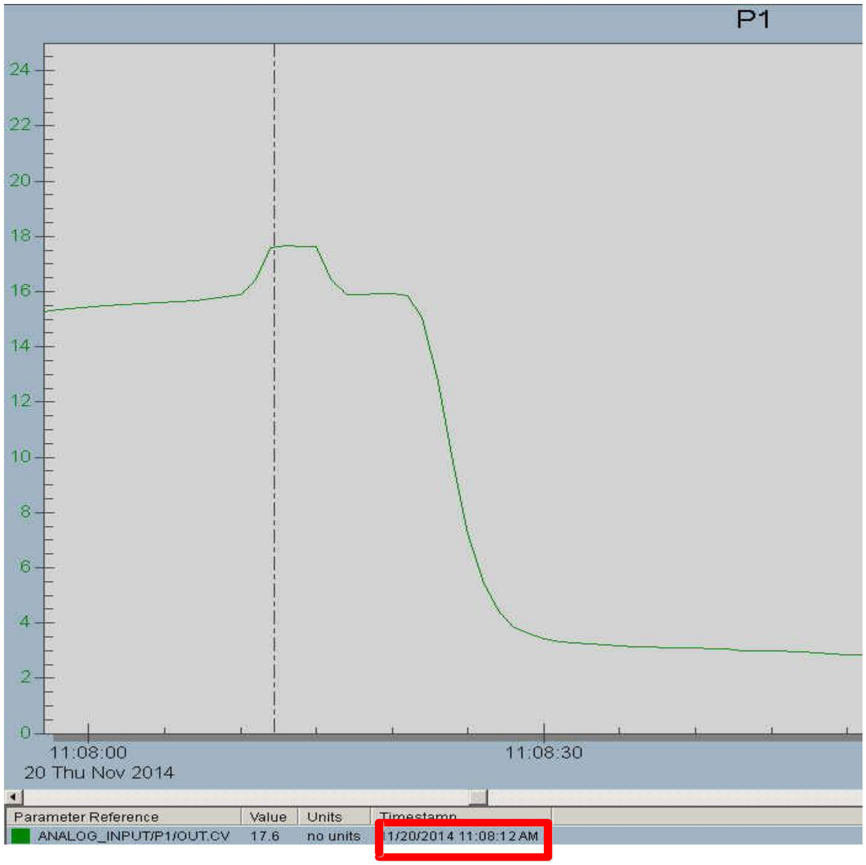

Figure 5.33 Primary water pressure P1 for ECC

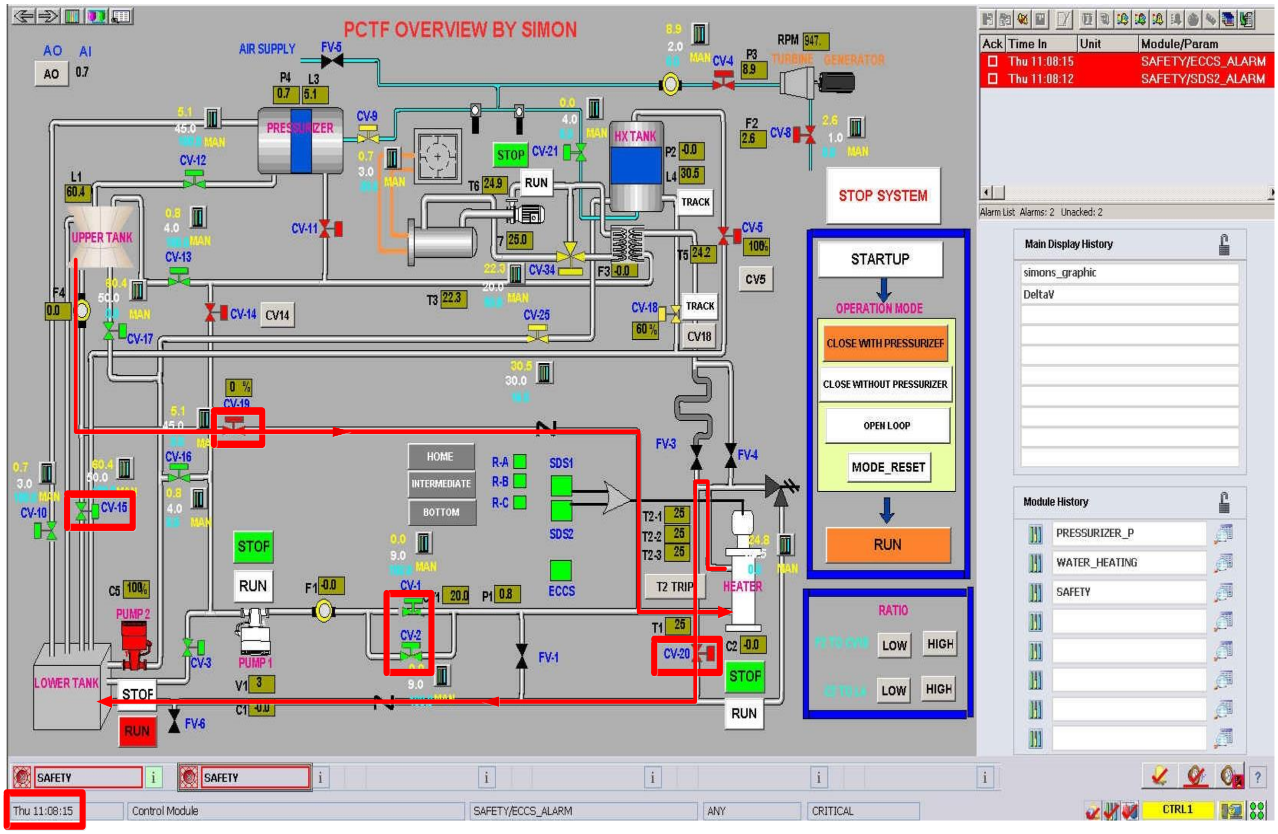

Figure 5.34 NPCTF system diagram for ECC

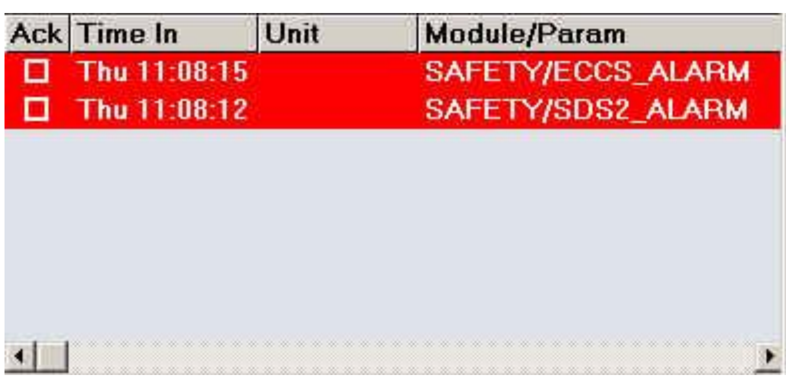

Figure 5.35 Alarms of ECC and SDS2

5.4 Operation Modes Evaluation

5.4.1 Normal mode evaluation

Under Normal mode, the primary water pressure P1 is controlled by the Pressurizer as well, as shown in Figure 5.36. Therefore, when the pushbutton “CLOSE WITH PRESSURIZER”, on the right side of Figure 5.36 is pressed, the Normal mode is selected. The control valve CV-11 is open to let the Pressurizer control the pressure of the system. Meanwhile, CV-3 and CV-13 are closed and CV-14 is opened to build the cycling loop. In Figure 5.36, CV-3 and CV-13 are shown in green which illustrates that the valves are closed and CV-11 and CV-14 are shown as red which indicates the valves are open.

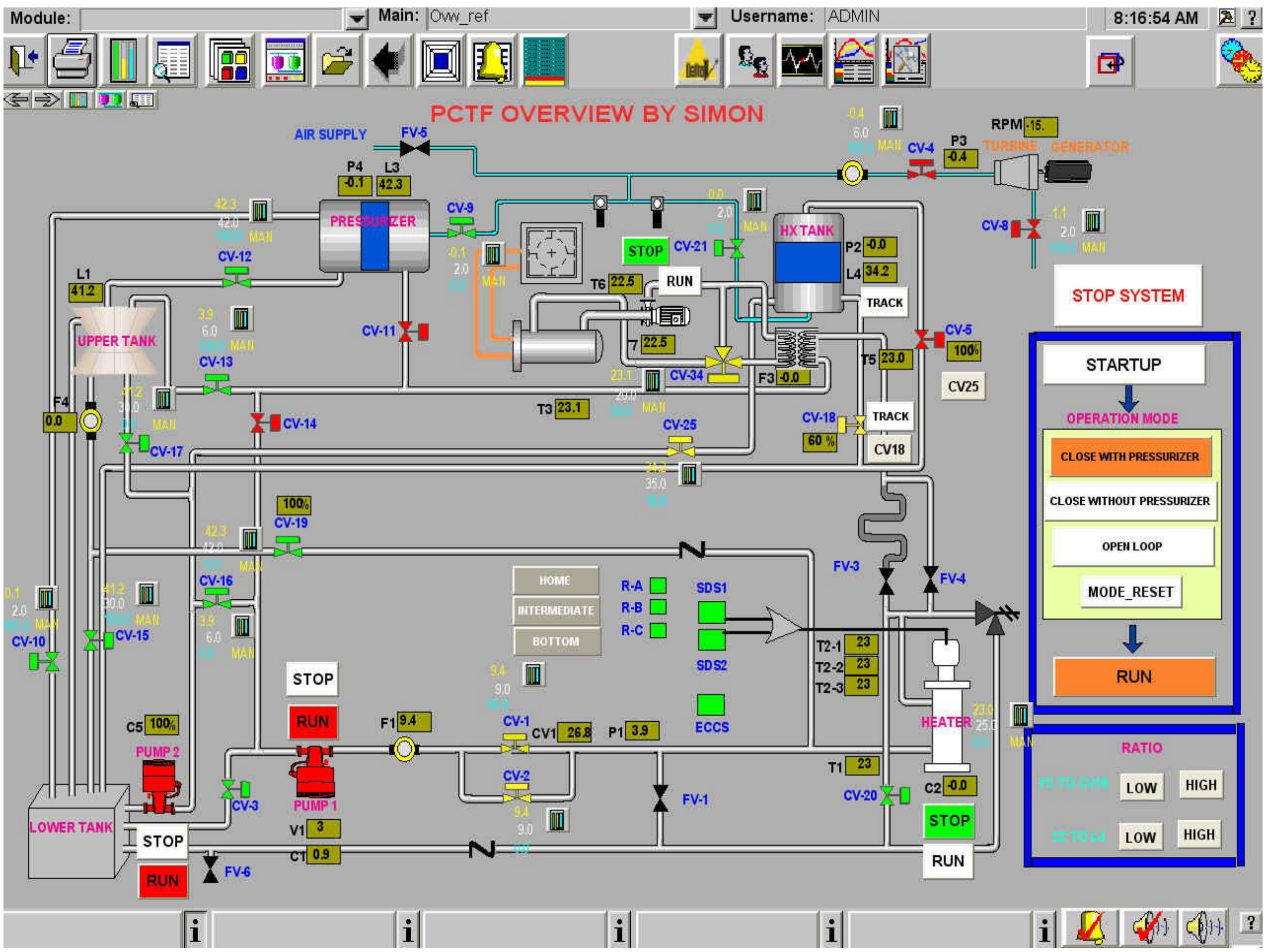

Figure 5.36 NPCTF system diagram during “Close With Pressurizer Mode”

5.4.2 Solid mode evaluation

Under Solid mode, the Pressurizer is isolated and the cooling water pressure is controlled by “feed and bleed”. Under Solid mode in NPCTF, the primary water pressure P1 is controlled by valve CV-16 which functions as the “feed” to increase the water inventory, and CV-13 which functions as the “bleed” to decrease the inventory. As shown in Figure 5.37, when the “CLOSE WITHOUT PRESSURIZER” button is pressed, the Solid mode is selected. CV-11 is closed to isolate the Pressurizer. CV-13 and CV-3 are closed and CV-14 is open to build the cycling loop. In Figure 5.37, CV-11, CV-3, and CV-13 are green to indicate close status and CV-14 is red to show open status. The water pressure P1 is controlled by CV-16 and CV-13.

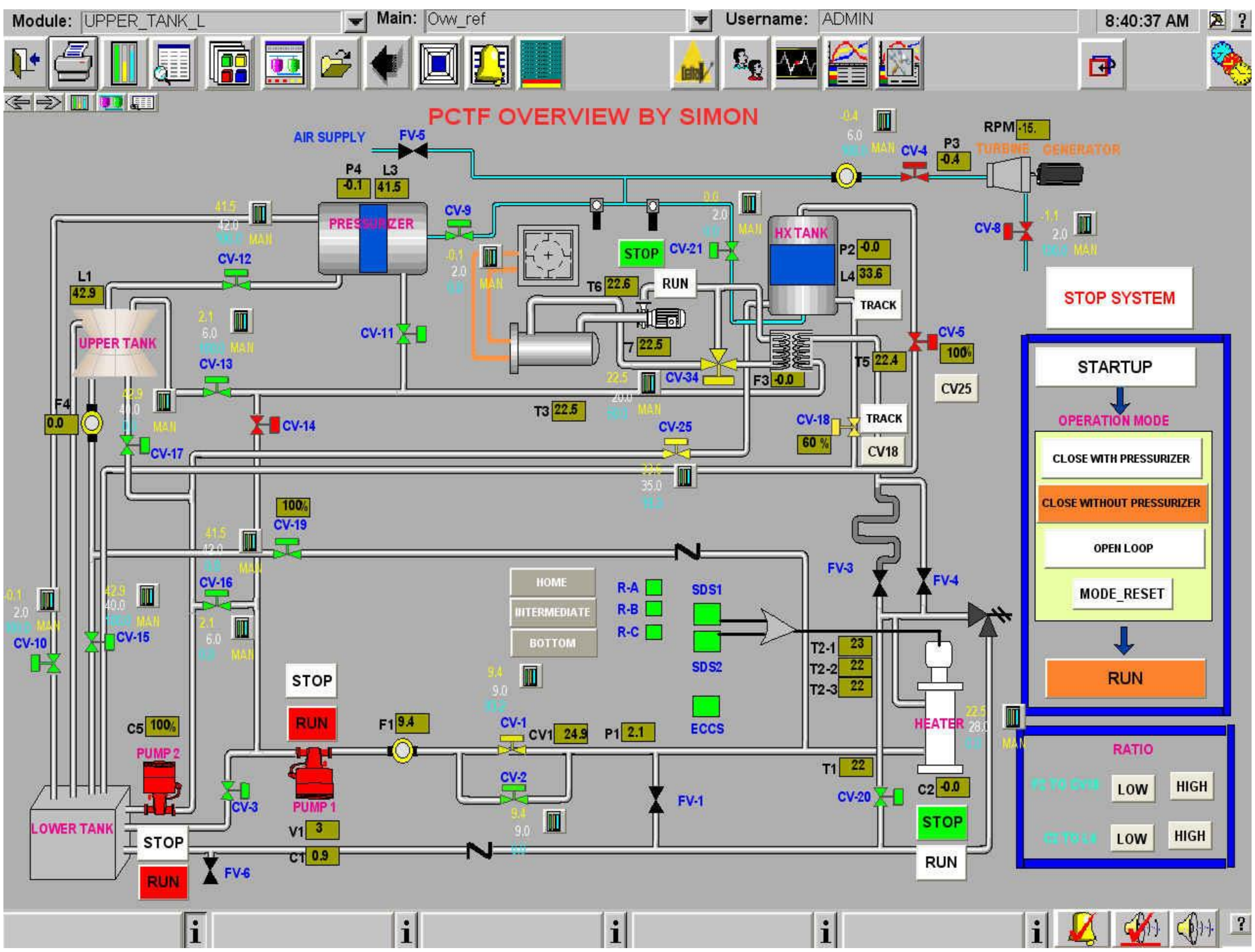

Figure 5.37 NPCTF system diagram during “Close Without Pressurizer Mode”

5.4.3 Open mode evaluation

When the “Open Mode” button is pushed, as indicated in the right side of Figure 5.38, the operation of Open mode is selected. Under Open mode, CV-3 and CV-13 are open, and CV-11 and CV-14 are closed. As indicated in Figure 5.38, CV-3 and CV-13 becomes red to indicate an open status and CV-11 and CV-14 becomes green show closed status.

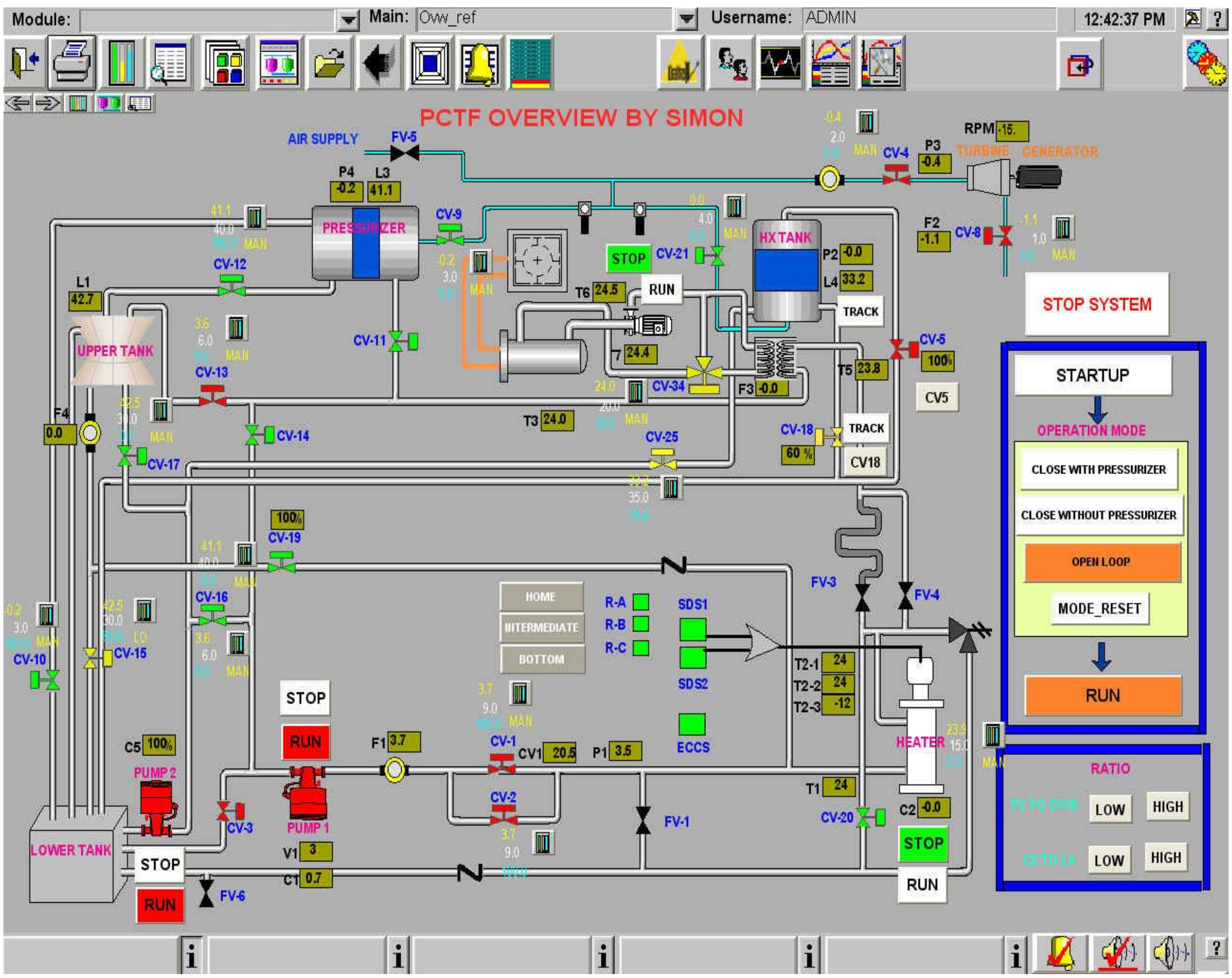

Figure 5.38 NPCTF system diagram during “Open Mode”

5.5 Summary

In summary, the implemented four control functions on NPCTF are evaluated in this Chapter, including process control, sequence control, safety systems and operation modes, which are designed in Chapters 3 and 4. From the evaluation results shown above, it can be concluded that the DeltaV DCS can perform all of these functions.

Chapter 6

Conclusions

This thesis focuses on the programming and implementation of a DCS platform for NPCTF. The main functions of this NPCTF are discussed. The functions are also presented in logic diagrams. These functions are implemented in DCS system. Therefore, a popular industrial DCS system, DelatV, is designed and built to operate the NPCTF. The DelatV is setup and configured. Programming architecture is designed to organize these functions under one “plant” in DCS. Industrial languages are used to program the algorithm of these functions. HMI is created to operate and monitor the NPCTF. The SG Tank level control loop is selected to demonstrate the process of design, a controller starting from the model, model analysis and controller design. Using the MATAB system identification tool, the model of the SG Tank level control is created and analyzed. Then the controller parameters are calculated from the model and applied to controller of DeltaV DCS. The response in the NPCTF proves that the method of design of the controller based on the model can control the process. Response comparison between MATLAB and NPCTF is similar. After the system is implemented, it is commissioned to perform the functions.

The evaluation of the system concludes that this DeltaV DCS can implement the functions of the NPCTF. When the designed safety conditions are satisfied, the safety systems of SDS1 or SDS2 or ECC can perform designed actions, such as shut off heater. For the process control, designed controllers can control the process variables through manipulating related variables. Under start-up sequence control, the DCS operates to satisfy the fill up requirement of pipes and tanks. The operator or researcher can select the different operation modes to satisfy their requirements via HMI. Not only can HMI supervise the process, observe and record parameter trends, but it can also operate the NPCTF system and change the controllers’ parameters to satisfy the superficial research requirements. In conclusion, the objective of the research is satisfied.

6.1 Summary of Contributions

The major contributions of the research presented in this thesis have included the following:

- The control system requirements of NPCTF are discussed. The requirements are summarized as process control, safety systems, sequence control, and operation modes. They are expressed in logic diagrams.

- An industrial DCS system, DeltaV, is selected to implement these requirements. A cabinet which holds the DelatV hardware is designed and built. They are connected to field devices with wires and to workstation through Ethernet. The DelatV workstation and HMI is configured and setup to be able to communicate with field devices and process variables can be monitored and be recorded.

- Programming architecture is designed to organize the requirements with the principle of easy troubleshooting and understanding. All of the functions are grouped under one “Plant”. Each process control loop, safety systems, sequential control, operation modes and each type of I/O cards is organized under one module respectively. Functions are programmed with industrial languages prescribed under the standard of IEC 61131-3. The system is programmed with the functions of safety operations which protect devices, and the system can stop safety at anytime. HMI is created to supervise the process of NPCTF, force the devices for purpose of research, and operate the system. The devices are indicated as red when in operating mode, such as running pumps, and grey when stopped. The pushbutton becomes brown when pushed. The valves are shown as red or if open, green when closed and yellow when in the middle. Real value and history values the process variables can be observed and recorded.

- The DeltaV system is commissioned for proper communication and correct signals between field devices, PLC and workstation, HMI. The signals to and from field devices are calibrated to indicate the real device status. They are also checked to be shown correctly in the HMI and workstation.

- The SG Tank level control process model is developed within operating range of 36% to 44% using the MATLAB system identification tool. The model is used to analyze the stability of the process and design the controller parameters. The parameters calculated from the model are applied to the NPCTF system. Its responses prove that the controller designed from the model can control the process within operating tange of 36% to 44%. Also, the SG Tank level control system is compared between MATLAB and NPCTF and show that the response is similar.

- Other process loops, except SG Tank level control, are tuned with the empirical method to have as better responses as possible.

- The designed control system is evaluated to function as designed. To evaluate three safety shutdown systems (SDS1, SDS2, ECC), the conditions which activate the designed safety systems are created and actions are observed from HMI and the NPCTF. The main purpose of the sequential control is fill up the pipes and tanks and, thus, related valves and levels of the tanks can be observed the change of status. Three operation modes are selected separately from HMI, and related valves are observed to have corresponding changes. All process control loops, except the SG Tank level control, are tuned with empirical method to have the best possible performance.

6.2 Suggestions for Future Work

Based on the research, the following are the future subjects suggested for further research:

- Even though the current design of controllers can control the system, the evaluation results show different characteristic responses at different operating points. Some loops, such as the Upper Tank level control, show more nonlinear characteristic since the Upper Tank is designed as a nonlinear system. Nonlinear controller or methods are suggested to control such processes.

References

[1] G. L. Brooks, A Short History of the CANDU Nuclear Power System, January 1993.

[2] Ontario Power Generation, Unit Operational Control, pp. 136-138, April 1993.

[3] CANDU 6 Program Team, Candu 6 Technical Summary, pp. 62, June 2005.

[4] The Canadian Nuclear Industryand its Economic Contributions. Accessed: March 5, 2015. http://www.nrcan.gc.ca/energy/uranium-nuclear/7715

[5] CANDU Reactors Worldwide. Accessed: March 5, 2015. http://teachnuclear.ca/all-things-nuclear/canadas-nuclear-history/candu-reactors-worldwide/

[6] G. T. Bereznai, *Nuclear Power Plant Systems and Operation*, University of Ontario Institute Technology, 2011.

[7] ANSI/ISA-S5.1-1984(R 1992), Instrumentation Symbols and Identification.

[8] K. Kant, *Computer-Based Industrial Control*, 2^{nd}, PHI Learning Pvt. Ltd, 2013.

[9] S. K. Singh, *Computer-Aided Process Control*, PHI Learning Pvt. Ltd, 2004.

[10] D. Popovic and V. Bhatkar, *Distributed Computer Control for Industrial Automation*, New York, 1990.

[11] T. Samad, "System architecture for process automation: Review and trends," *Journal of Process Control*, v 17, n 3, pp. 191-201, March 2007.

[12] S. K. Singh, *Industrial Instrumentation and Control*, 3^{rd}, New Delhi, 2012.

[13] IDE Technologies, *Practical Distributed Control Systems for Engineer and Technician*, Revision 6.1, 2012.

[14] L. F. Wang and K. C. Tan, *Modern Industrial Automation Software Design*, Wiley-IEEE Press. Febuary 2006.

[15] M. Beharrell and R. Barillere, "OPC Evolution toward UNIX," *10^{th} ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control Systems*. Geneva, 10-14 October, 2005.

[16] K. C. Kwon and M. Lee, "Technical Review on the Localized Digital

Instrumentation and Control Systems," *Nuclear Engineering and Technology* , Vol.41, No.4, May 2009.

[17] S. F. Gray and S. Basu, "Turbine Control System Upgrade for Bruce Nuclear Plant Units 1 and 2," *IEEE International Conference on Electro/Information Technology*, June 2009.

[18] Combined Reliability-Centered Course for Instrumentation Professionals. Accessed: March 5, 2015. http://www.emersonprocessxperts.com/2014/06/combined-reliability-centered-course-for- instrumentation-technicians/.

[19] History of PLC and DCS. Accessed: March 5, 2015. http://www.control.lth.se/media/Education/DoctorateProgram/2012/HistoryOfControl/Vanessa_Alfred_report.pdf.

[20] W. B. Dai, V. Vyatkin, and J. H. Christensen, "Essential Elements for Programming of Distributed Automation and Control Systems," *IEEE 18th conference on Emerging Technologies & Factory Automation (ETFA)*, 2013.

[21] IEC 61131-3, Programmable controllers - Part 3:Programming languages. International Standard, Second Edition, 2003.

[22] J. P. Morgan, Global Industrial Automation Market Review. Global Equity Research, June 2010.

[23] M. Yang, et al., "A Human-Centered Approach based on Multilevel Flow Models for Diagnosing Fault in Nuclear Power Plant," *Proceedings of ICAPP 2007*, Nice, France, May13-18, 2007.

[24] Worldwide DCS Market Shares (%) Source : ARC Advisory Group, 2013.

[25] CIES Research Group of UWO, Nuclear Power Control Test Facility (NPCTF) User Manual, first draft, 2015.

[26] W. Boyes and L. O'Brien, "The Top 50 Automation Companies." *Control*. pp. 22-24, December 2009.

[27] J. P. Morgan, Global Industrial Automation, Global Equity Research, June 2010.

[28] DeltaV Books Online Version 8.4.

[29] X. L. Cui and Z. C. Zhou, “The Algorithm of Translation between the Ladder Language and the Sentence Table Language of PLC,” *Microelectronics and Computer*, vol. 17, NO. I, pp. 26-30, 2000.

[30] Z. B. Zhu and D. B. Zhao, “Implementation of Ladder Diagram to STL in SoftPLC,” *Mechanical and electronic*, NO. 12, pp. 61-64, 2008.

[31] R. Wareham, “Ladder Diagram and Sequential Function Chart Languages in Programmable Controllers,” *Programmable Control and Automation Technology Conference and Exhibition*, 1988. Conference Proceedings, Fourth Annual Canadian, 1988 , Page(s): 12A - 14/1-4.

[32] IDC Technologies, *Practical Industrial Programming Using 61131-3 for PLCs*, 2007.

[33] IEC 62138, 2004. Nuclear Power Plant – Instrumentation and Control Important for Safety – Software Aspects for Computer-based Systems Performing Category B or C Functions.

[34] K. H. John and M. Tiegelkamp, *IEC 61131-3:Programming Industrial Automation Systems*, 2nd edition, New York, 2010.

[35] M. King, *Process Control-A Practical Approach*, John Wiley & Sons Ltd, UK, 2011.

[36] T. Liu and F. R. Gao, *Industrial Process Identification and Control Design*, Springer, 2012.

[37] L. Ljung, *System Identification - Theory for the User*, 2rd, Sweden,1999.

[38] R. C. Dorf and R. H. Bishop, *Modern Control System*, 12th ed, New Jersey, Pearson, 2011.

[39] D. E. Seborg, T. F. Edgar, and D. A. Mellichanp. *Process Dynamics and Control*. 2nd ed., New York, Wiley, 2004.

[40] A. O'Dwyer, “A Summary of PI and PID Controller Tuning Rules for Processes with Time Delay,” *IFAC Workshop on Digital Control*, pp. 175-180, Terrassa, Spain, April 4-7, 2000.

[41] A. O’Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 3rdedition, Imperial College Press, London, 2009.

[42] D. E. Rivera, M. Morari and S. Skogestad, “Internal Model Control. 4.PID Controller Design,” *Ind. Eng. Chem. Res.* , 25(1), pp. 252-265, 1986.

[43] I. L. Chien and P. S. Fruehauf, “Consider IMC Tuning to Improve Controller Performance,” *Chem. Eng. Prog.* , 86, pp. 33-41, 1990.

[44] I. G. Horn, J. R. Arulandu, J. Gombas, J. G. VanAntwerp and R. D. Braatz, “Improved Filter Design in Internal Model Control,” *Ind. Eng. Chem. Res.* , 35(10), pp. 3437-3441, 1996.

[45] S. Skogestad, “Probably the Best Simple PID Tuning Rules in the World,” *Journal of Process control* , September 12, 2001.

[46] C. A. Smith and A. B. Corripio, *Principles and Practice of Automatic Process Control*, John Wiley & Son, Inc., New York, 1985.

[47] C. L. Smith, *Practical Process Control-Tuning and Troubleshooting*, John Wiley & Son, Inc., US, 2009.

[48] W. S. Levine, *The Control Handbook*, CRC Press, 1999.

[49] ANSI/ISA-5.1-2009, Instrumentation Symbols and Identification.

Appendices

A. List of I/O (input/output) signals

Table A.1 lists all of the I/O signals to and from the field devices. There are four kinds of I/O signals: analog input (AI), analog output (AO), digital input (DI) and digital output (DO). For this NPCTF system, there are signals of 25 AI, 20 AO, 3 DI, 12 DO. They are connected to DeltaV IO cards through terminals in a DCS junction box and DeltaV cabinet. Table A.1 lists the terminal numbers in the DCS junction box and DeltaV cabinet, and cable label of all the signals. For example, AI signal C1_I is connected to terminals 1 in DeltaV cabinet and terminal 1 in DCS JB. The cable that connects the signal is in cable bundle labelled “DLTV1”.

Table A.1 List of I/O signals

TB# DLV	**SGLs**	**IO Cards**	**CBL #**	**TB# DCS JB**	**SGLs**	**IO Type**	**CBL #**	**DESC**
1	C1_I	**AI1**	**DLTV1**	1	C1_I1	**AI(18)**	**DLTV1**	
3	C2_I			4	C1_I2			
5	RMP_I			7	C2_I1			
7	F1_I			10	C2_I2			
9	F2_I			13	RMP_I1			
11	F3_I			16	RMP_I2			
13	F4_I			19	F1_I1			

15	L1_I			22	F1_I2			
17	T4_I	**AI2**		25	F2_I1			
19	L3_I			28	F2_I2			
21	L4_I			31	F3_I1			
23	P1_I			34	F3_I2			
25	P2_I		**DLTV2**	37	F4_I1			
27	P3_I			40	F4_I2			
29	P4_I			43	L1_I1			
31	T1_I			46	L1_I2			
33	T2_1_I	**AI3**		49	T4_I1			
35	T2_2_I			52	T4_I2			
37	T2_3_I			55	L3_I1			
39	T3_I			58	L3_I2			
41	T5_I			61	L4_I1			
43	T6_I			64	L4_I2			
45	T7_I			67	P1_I1			
47	V1_I			70	P1_I2			
49	CV1_I	**AI4**	**DLTV3**	73	P2_I1		**DLTV2**	

65	C1_O	**AO1**		76	P2_I2			
67	C5_O			79	P3_I1			
69	CV1_O			82	P3_I2			
71	CV2_O			85	P4_I1			
73	CV4_O			88	P4_I2			
75	CV5_O			91	T1_I1			
77	CV18_O			94	T1_I2			
79	CV9_O			97	T2_1_I1			
81	CV10_O	**AO2**		100	T2_1_I2			
83	CV13_O			103	T2_2_I1			
85	CV16_O			106	T2_2_I2			
87	CV17_O	**AO2**	**DLTV4**	109	T2_3_I1	**AI(6)**	**DLTV2**	
89	CV19_O			112	T2_3_I2			
91	CV21_O			115	T3_I1			
93	CV34_O			118	T3_I2			
95	CV15_O							
97	CV25_O	**AO3**						
99	CV8_O			121	T5_I1			

101	CV12_O			124	T5_I2			
103	C2_O			127	T6_I1			
113	RCV1	**DI**		130	T6_I2			
115	SDS1			133	T7_I1			
117	SDS2			136	T7_I2			
129	C1_C_D	**DO1**	**DLTV5**	139	V1_I1			
131	C5_C_D			142	V1_I2			
133	CV_20_D			145	C1_O	**AO(3)**	**DLTV3**	pump1
135	CV_11_D			148	C5_O			pump2
137	C2_C_D			151	CV1_O			CV1
139	CC_1_D			154	CV1_I1	**AI(1)**	**DLTV3**	
141	CV_14_D			157	CV1_I2			
143	CV_3_D			160	RCV1	**DI(1)**	**DLTV5**	CV1 Relay
145	ROD DOWN	**DO2**		162	RCV1			
147	DOWN			164	CV2_O	**AO(12)**	**DLV3**	

149	INTERM			167	CV4_O			
151	HOME			170	CV5_O			
				173	CV18_O			
				176	CV9_O			
				179	CV10_O			
				182	CV13_O			
				185	CV16_O			
				188	CV17_O		**DLTV4**	
				191	CV19_O			
				194	CV21_O			
				197	CV34_O			
				200	CV15_O	**AO(5)**	**DLTV4**	
				203	CV25_O			
				206	CV8_O			
				209	CV12_O			
				248	C2_O			heater
				212	SDS1_1	**DI(2)**	**DLTV4**	SDS

				214	SDS1_2			
				216	SDS2_1			
				218	SDS2_2			
				220	C1_C_D 1	**DO(12)**	**DLTV5**	Pump1
				222	C5_C_D 1			Pump2
				224	CV_20_ D1			
				226	CV_11_ D1			
				228	C2_C_D 1			Heater
				230	CC_1_D 1			Chiller
				232				
				234				
				236	CV_14_ D1			
				238	CV_3_D 1			

				240	ROD DOWN			Linear Motor Relay
				242	DOWN			
				244	INTERM			
				246	HOME			

B. Process control loops I/O

There are eleven process control loops and eleven variables to be adjusted to desired values.

There are two actuators for the four loops, one for increasing the controlled variable and another for decreasing the controlled variable. Table B.1 lists all of the process control loops.

Table B. 1 Process control loops I/O

Loop Name	**Manipulated Variables(u)**	**Controlled Variables(y)**
Primary water flow	CV-1	F1(water flow)
Water heating	C2 (heater power)	T2(average temperature)
HX outlet temperature	CV-34(chilled flow)	T3(temperature)
HX Tank pressure	CV-21	P2 (pressure)
HX Tank level	CV-25	L4(water level)
Pressurizer pressure	CV-9(Main)	P4 (pressure)
	CV-10(Fixed)	
Pressurizer water level(water inventory)(close loop with pressurizer)	CV-12(Fixed)	L3(water level)
	CV-16(Main)	

Upper Tank level	CV-15(Fixed)	L1 (water level)
	CV-17(Main)	
Water pressure(close loop without pressurizer)(no control for open loop)	CV-16(Main)	P1(pressure)
	CV-13(Fixed)	
Air pressure	CV-4	P3(pressure)
Air flow	CV-8	F2(flow)

C. Safety Systems Logic Diagrams

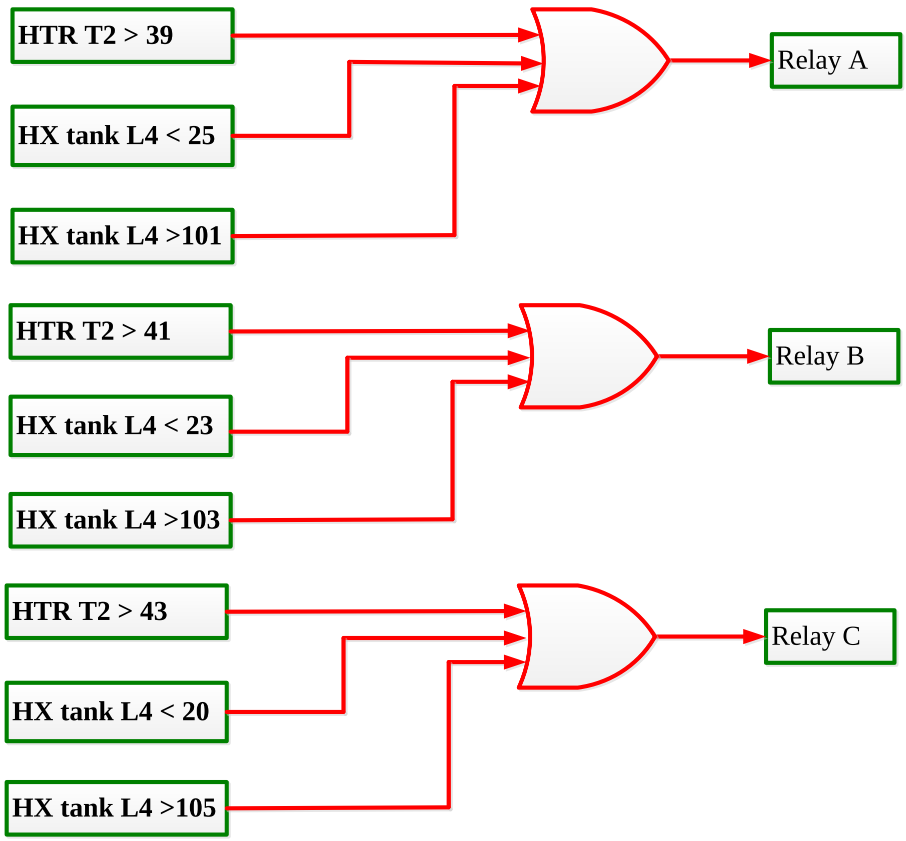

Figure C.1 SDS1 relays conditions

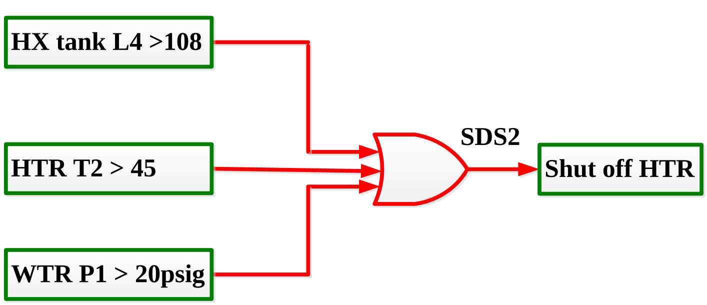

Figure C.2 SDS2 conditions

D. Programs of NPCTF in DeltaV DCS system

The whole control strategy is created using an engineering tool called DeltaV Explorer. The whole system named “PCTF_SIMON”, as shown in Figure D.1, is created under Control Strategies. There are twenty modules developed for the NPCTF system, as illustrated in Figure D.1. For the process control loops, the last alphabet of the module name after the underscore denotes the controlled parameter. For example, the module of AIR_P contains the control strategy to control the pressure of air. Modules of ANOLOG_INPUT, ANOLOG_OUTPUT, DISCRETE_INPUT, and DISCRETE_OUTPUT include all the IO signals. The module of OPERATION_MODE covers all the algorithms of three operation modes, i.e. Normal, Solid and Open mode. There are two ratio algorithms under the RATIO module. The SAFETY module includes the logic of SDS1, SDS2, ECC, and the safety protection of the Heater. The SEQUENCE module is programmed using SFC industry language and includes the procedure logic to start the system. The STOP_SYSTEM module is also created using SFC language and includes the logic to put the system in safety status. All the pumps, heaters, and motors are stopped and valves are pushed to normal status.

Each module includes different kinds of algorithm, such as function blocks or SFC (Sequential Function Chart). Figures D.2 to D.135 show the detail algorithm of each module. PID tuning parameters for each PID are also included in the figures of each module.

D.1 Control Modules

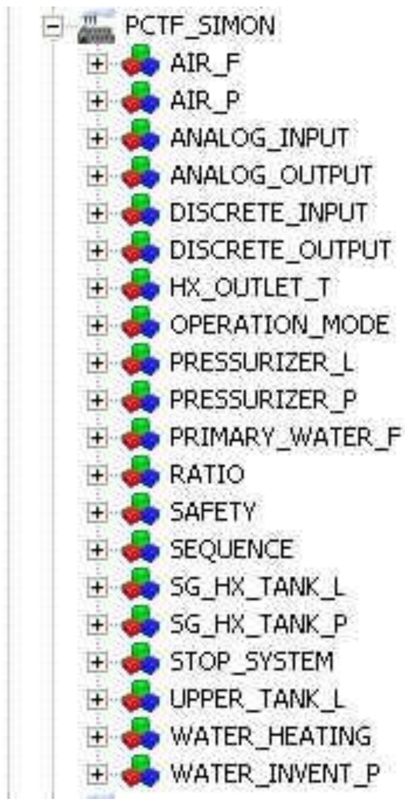

Figure D.1 Control modules

D.2 Air Flow

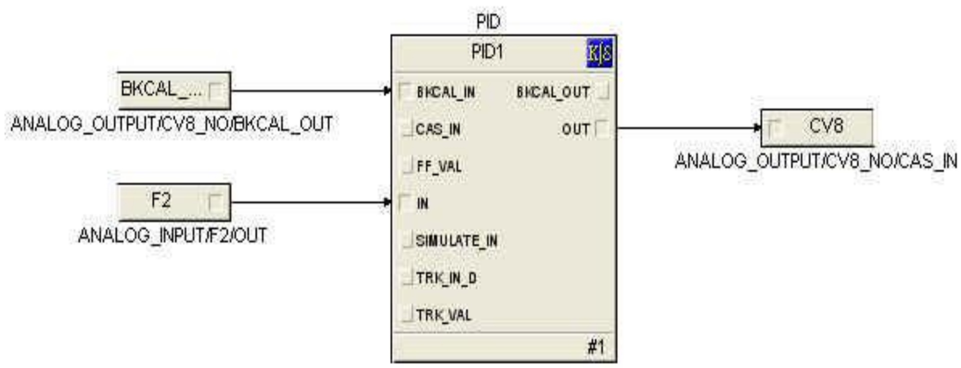

Figure D.2 Air flow control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	1
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 100.0...
RATE	0
RESET	50
SP_FTIME	0
SP_HI_LIM	100
SP_LO_LIM	0

Figure D.3 PID tuning parameters

D.3 Air Pressure

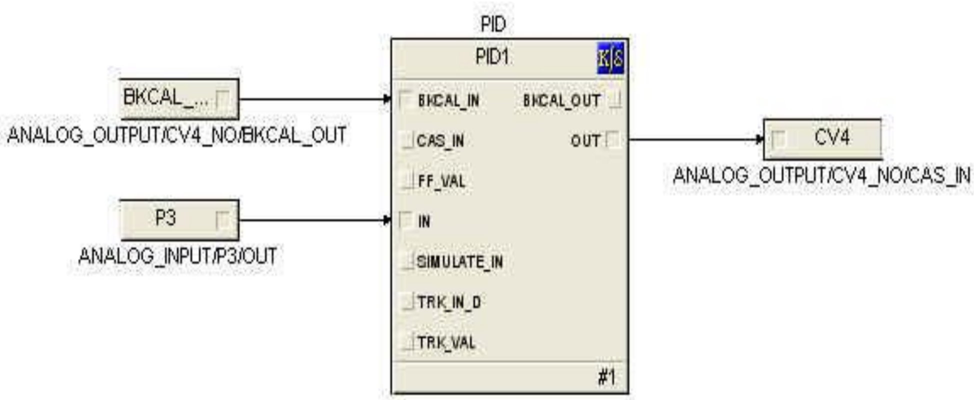

Figure D.4 Air pressure control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	0.2
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 50.0 ...
RATE	0
RESET	0
SP_FTIME	0
SP_HI_LIM	50
SP_LO_LIM	0

Figure D.5 PID tuning parameters

D.4 Analog Inputs

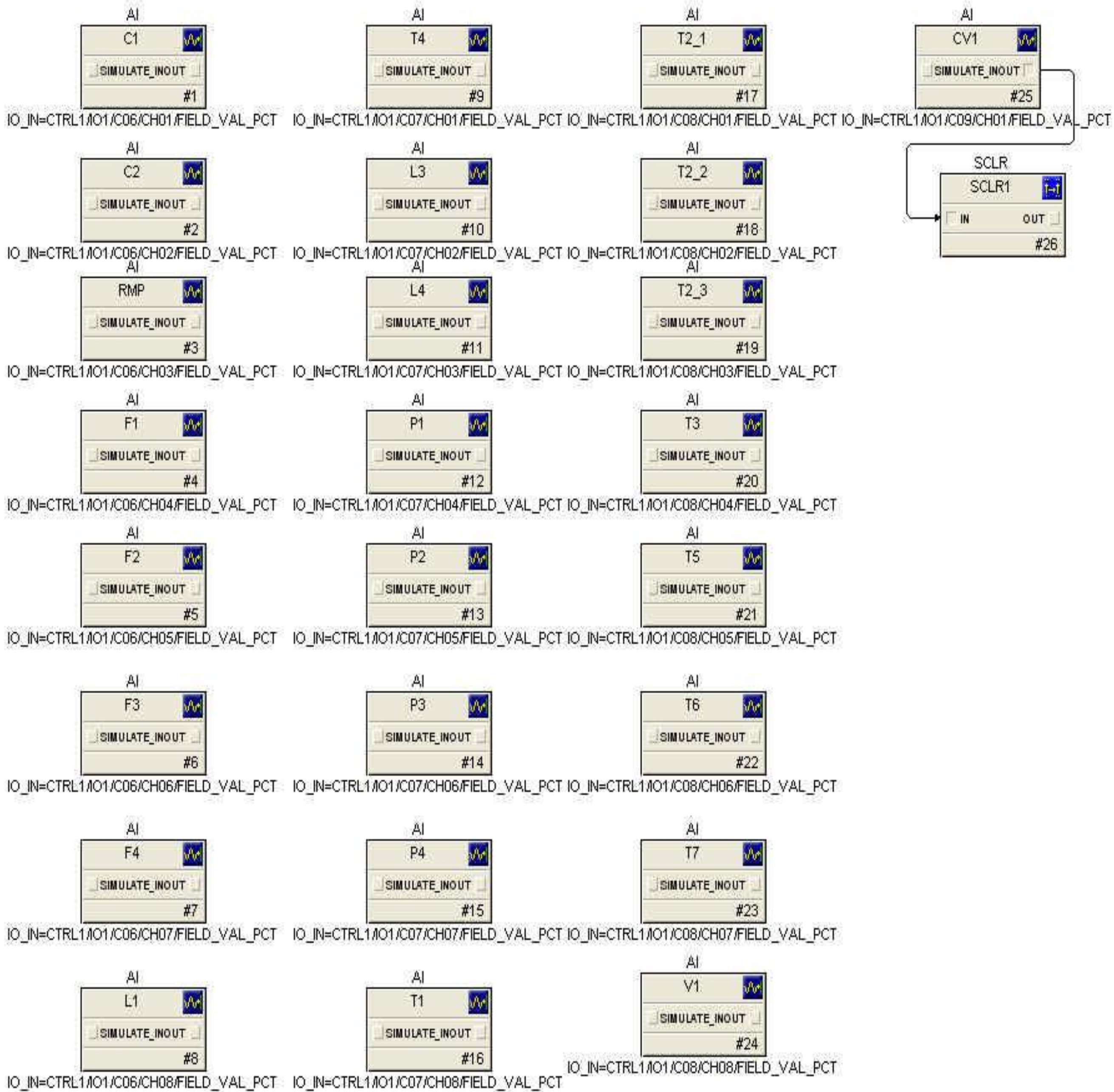

Figure D.6 Analog inputs

D.5 Analog Outputs

Figure D.7 Analog outputs

D.6 Digital Inputs

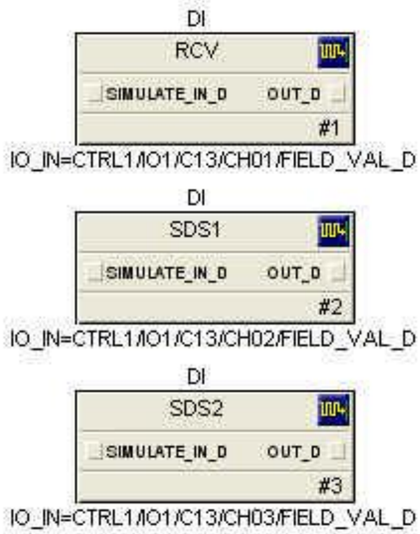

Figure D.8 Digital inputs

D.7 Digital Outputs

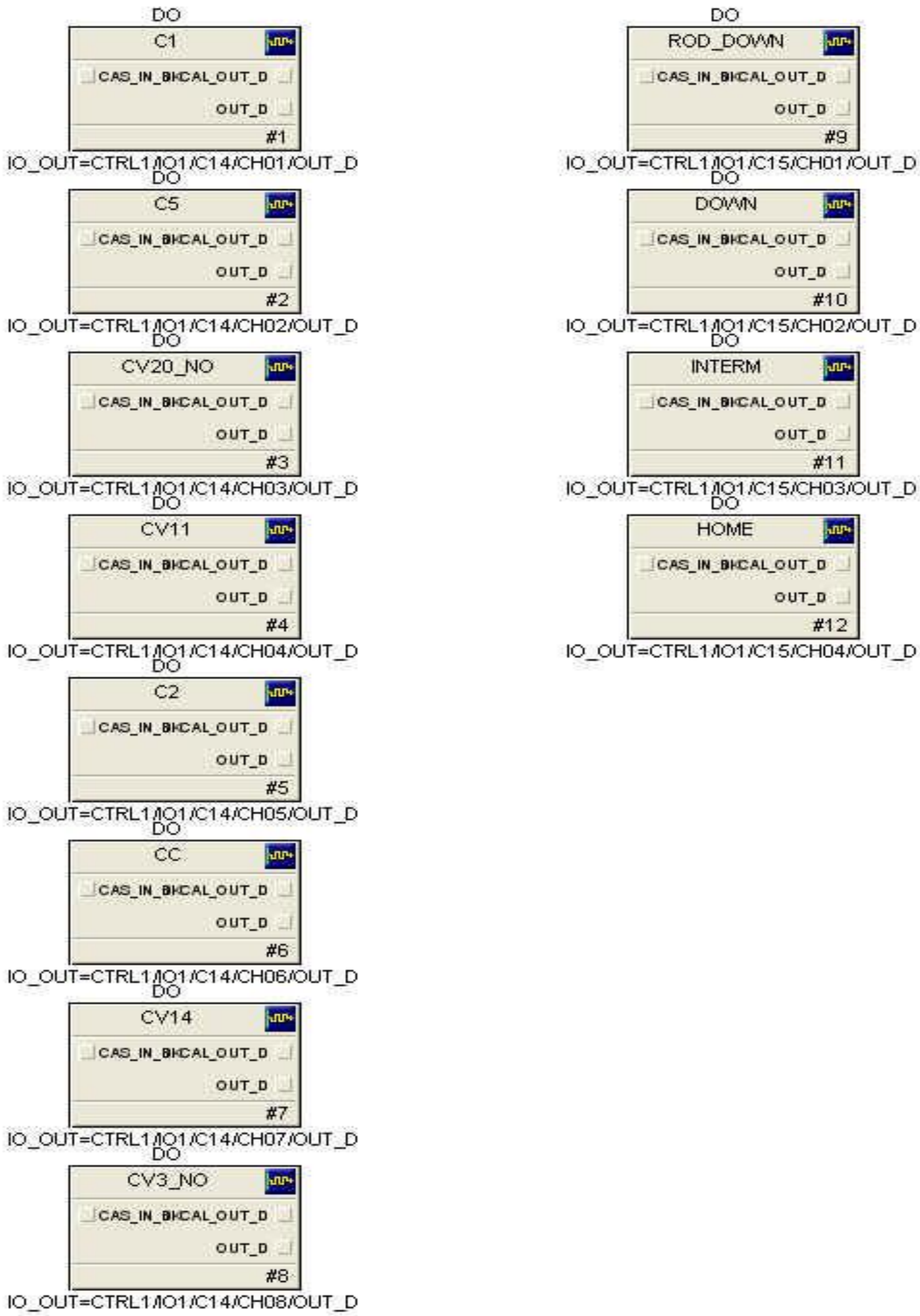

Figure D.9 Digital outputs

D.8 HX_OUTLET_T

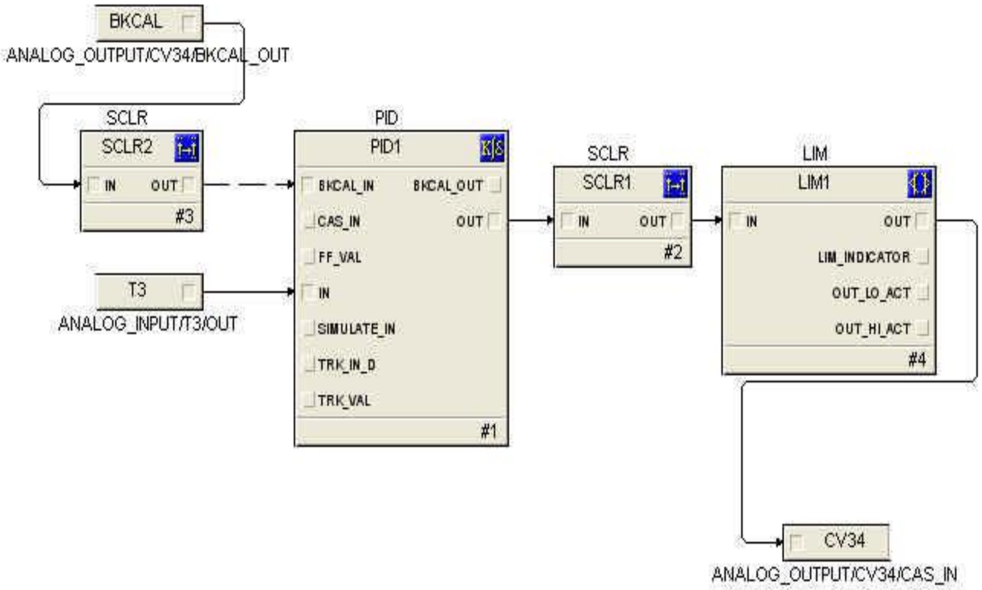

Figure D.10 HX tank outlet temperature control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	10
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 50.0 no units
RATE	0
RESET	20
SP_FTIME	0
SP_HI_LIM	50
SP_LO_LIM	0

Figure D.11 PID tuning parameters

Parameter	Default
IN	0
IN_SCALE	0.0 to 100.0...
OUT	0
OUT_SCALE	100.0 to 0.0...

Figure D.12 SCLRS function block

Parameter	Default
IN	0
LIM_INDICATOR	0
OUT	0
OUT_HI_ACT	0
OUT_HI_LIM	60
OUT_LO_ACT	0
OUT_LO_LIM	10

Figure D.13 LIM1 function block

D.9 Operation Mode

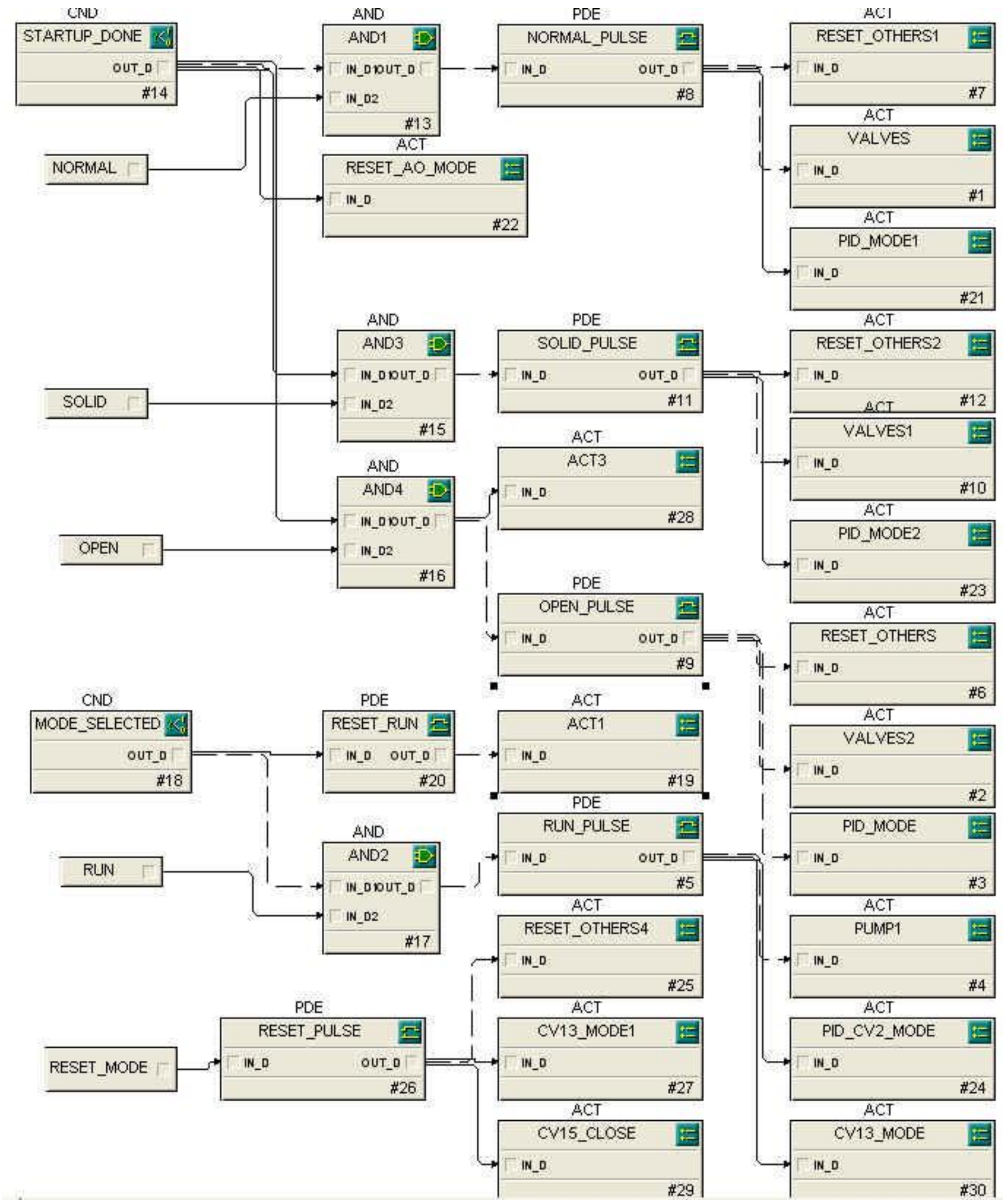

Figure D.14 Operation mode control

Parameter	Default
ABNORM_ACTIVE	False
BAD_ACTIVE	False
BLOCK_ERR	
EXEC_TIME	0
MCOMMAND	In Service
MERROR	
MSTATE	In Service
MSTATUS	
NORMAL	START
OPEN	START
RESET_MODE	START
RUN	START
SOLID	START
VERSION	1

Figure D.15 Operation mode control parameters

D.9.1 Normal mode

Expression:

```
'//ANALOG_INPUT/L4/OUT.CV' > 16 AND
'//SEQUENCE/SP.CV' = 0
```

Figure D.16 STARTUP_DONE function block

Expression:

```
'//ANALOG_OUTPUT/CV1_NO/MODE.TARGET' := CAS;
'//ANALOG_OUTPUT/CV2_NO/MODE.TARGET' := CAS;
'//ANALOG_OUTPUT/C2/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV21/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV34/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV25/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV4_NO/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV8_NO/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV9/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV10_NO/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV12_NO/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV16/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV15_NO/MODE.TARGET':= CAS;
'//ANALOG_OUTPUT/CV17/MODE.TARGET':= CAS;
```

Figure D.17 RESET_AO_MODE function block

Expression:

```
'^/SOLID.CV' := 0;
'^/OPEN.CV' := 0;
'^/RESET_MODE.CV' := 0
```

Figure D.18 RESET_OTHERS1 function block

Expression:

```
'//DISCRETE_OUTPUT/CV3_NO/OUT_D.CV' := 1;
'//ANALOG_OUTPUT/CV13_NO/OUT.CV' := 100;
'//DISCRETE_OUTPUT/CV11/OUT_D.CV' := 1;
'//DISCRETE_OUTPUT/CV14/OUT_D.CV' := 1
```

Figure D.19 VALVES function block

Expression:

```
'//PRESSURIZER_L/PID_DE_CV12/MODE.TARGET':= MAN;
'//PRESSURIZER_L/PID_IN_CV16/MODE.TARGET' := MAN;
'//WATER_INVENT_P/PID_DE_CV13/MODE.TARGET' := MAN;
'//WATER_INVENT_P/PID_IN_CV16/MODE.TARGET' := MAN;
'//PRESSURIZER_P/PID_IN_CV9/MODE.TARGET' := MAN;
'//PRESSURIZER_P/PID_DE_CV10/MODE.TARGET' := MAN;
```

Figure D.20 PID_MODE1 function block

D.9.2 Solid mode

Expression:

```
'^/NORMAL.CV' := 0;
'^/OPEN.CV' := 0;
'^/RESET_MODE.CV' := 0
```

Figure D.21 RESET_OTHERS2 function block

```
Expression:
'//DISCRETE_OUTPUT/CV3_NO/OUT_D.CV' := 1;
'//ANALOG_OUTPUT/CV13_NO/OUT.CV' := 100;
'//DISCRETE_OUTPUT/CV11/OUT_D.CV' := 0;
'//DISCRETE_OUTPUT/CV14/OUT_D.CV' := 1
```

Figure D.22 VALVES1 function block

```
Expression:
'//PRESSURIZER_L/PID_DE_CV12/MODE.TARGET' := MAN;
'//PRESSURIZER_L/PID_IN_CV16/MODE.TARGET' := MAN;
'//PRESSURIZER_P/PID_DE_CV10/MODE.TARGET' := MAN;
'//PRESSURIZER_P/PID_IN_CV9/MODE.TARGET' := MAN
```

Figure D.23 PID_MODE2 function block

D.9.3 Open mode

```
Expression:
IF '//SAFETY/ECCS/OUT_D.CV' = 0 THEN
'//UPPER_TANK_L/PID_IN_CV17/MODE.TARGET' := MAN;
'//UPPER_TANK_L/PID_IN_CV17/OUT.CV' := 0;
ENDIF;
```

Figure D.24 ACT3 function block

```
Expression:
'^/NORMAL.CV' := 0;
'^/SOLID.CV' := 0;
'^/RESET_MODE.CV' := 0
```

Figure D.25 RESET_OTHERS function block

```
Expression:
'//DISCRETE_OUTPUT/CV3_NO/OUT_D.CV' := 0;
'//ANALOG_OUTPUT/CV13_NO/OUT.CV' := 0;
'//DISCRETE_OUTPUT/CV11/OUT_D.CV' := 0;
'//DISCRETE_OUTPUT/CV14/OUT_D.CV' := 0;
```

Figure D.26 VALVES2 function block

Expression:

```
'//WATER_INVENT_P/PID_DE_CV13/MODE.TARGET' := MAN;
'//WATER_INVENT_P/PID_IN_CV16/MODE.TARGET' := MAN;
'//PRESSURIZER_L/PID_DE_CV12/MODE.TARGET' := MAN;
'//PRESSURIZER_L/PID_IN_CV16/MODE.TARGET' := MAN;
'//PRESSURIZER_P/PID_IN_CV9/MODE.TARGET' := MAN;
'//PRESSURIZER_P/PID_DE_CV10/MODE.TARGET' := MAN;
'//UPPER_TANK_L/PID_DE_CV15/MODE.TARGET' := AUTO;
'//UPPER_TANK_L/PID_IN_CV17/SP.CV' := 30;
'//ANALOG_OUTPUT/CV15_NO/MODE.TARGET' := CAS;
'//ANALOG_OUTPUT/CV17/MODE.TARGET' := CAS
```

Figure D.27 PID_MODE function block

D.9.4 Mode selected

Expression:

```
'^/NORMAL_PULSE/IN_D.CV' = 1  OR
'^/SOLID_PULSE/IN_D.CV' = 1  OR
'^/OPEN_PULSE/IN_D.CV' = 1
```

Figure D.28 MODE_SELECTED function block

Expression:

```
'^/RUN.CV' := 0
```

Figure D.29 ACT1 function block

D.9.5 Run

Expression:

```
'//DISCRETE_OUTPUT/C1/OUT_D.CV' := 1;
'//ANALOG_OUTPUT/C1/OUT.CV' := 100;
```

Figure D.30 PUMP1 function block

Expression:

```
'//PRIMARY_WATER_F/PID_CV2/MODE.TARGET' := AUTO
```

Figure D.31 PID_CV2_MODE function block

```
Expression:
'//ANALOG_OUTPUT/CV13_NO/MODE.TARGET'  := CAS
```

Figure D.32 CV13_MODE function block

D.9.6 Reset mode

```
Expression:
'^/NORMAL.CV'  := 0;
'^/SOLID.CV'  := 0;
'^/OPEN.CV'  := 0;
'^/RUN.CV'  := 0
```

Figure D.33 RESET_OTHERS4 function block

```
Expression:
'//ANALOG_OUTPUT/CV13_NO/MODE.TARGET':= MAN;
'//ANALOG_OUTPUT/CV13_NO/OUT.CV'  := 100
```

Figure D.34 CV13_MODE1 function block

```
Expression:
'//UPPER_TANK_L/PID_DE_CV15/MODE.TARGET'  := MAN;
'//UPPER_TANK_L/PID_DE_CV15/OUT.CV'  := 100
```

Figure D.35 CV15_CLOSE function block

D.10 Pressurizer Water Level

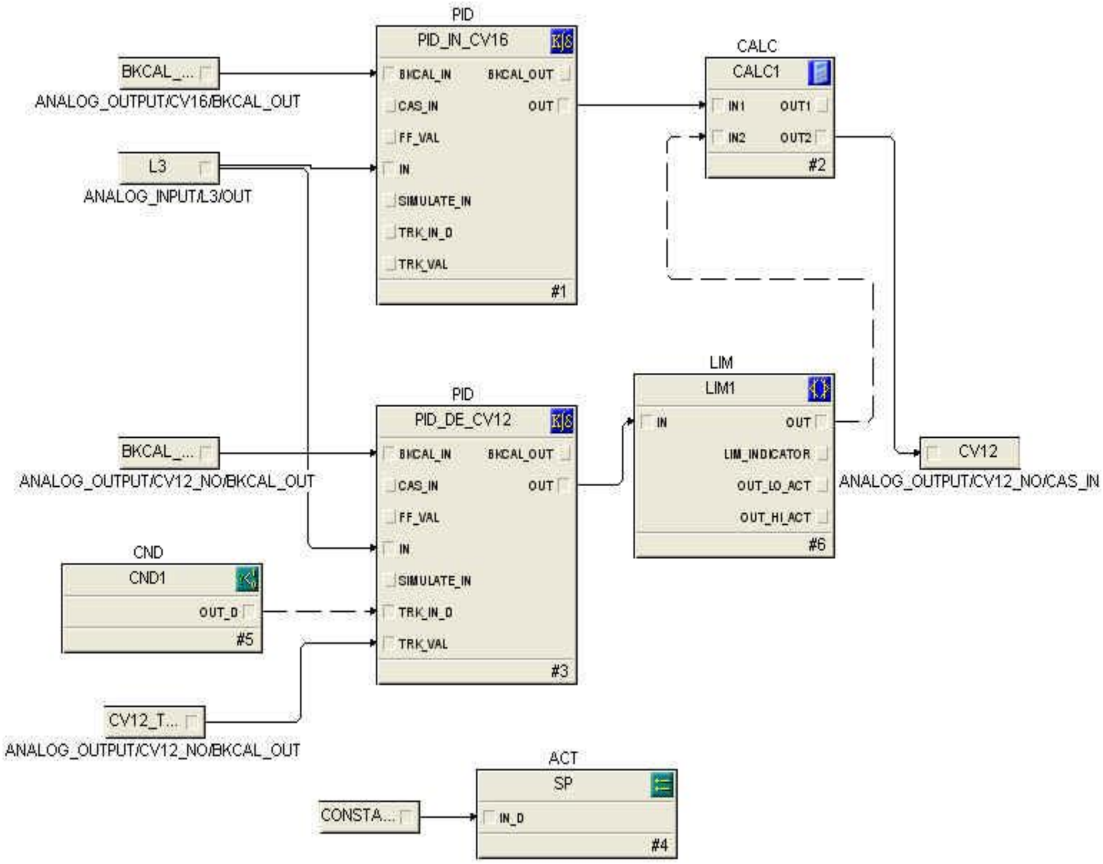

Figure D.36 Pressurizer level control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	4
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 100.0...
RATE	15
RESET	100
SP_FTIME	0
SP_HI_LIM	100
SP_LO_LIM	0

Figure D.37 PID_IN_CV16 tuning parameters

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	3
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 100.0...
RATE	10
RESET	100
SP_FTIME	0
SP_HI_LIM	100
SP_LO_LIM	0

Figure D.38 PID_DE_CV12 tuning parameters

Expression:

```
1 ABS('^/PID_IN_CV16/SP.CV'-'^/PID_IN_CV16/PV.CV')  < 1
```

Figure D.39 CND1 function block

Expression:

```
1 '^/PID_DE_CV12/SP.CV' := '^/PID_IN_CV16/BKCAL_OUT.CV'
```

Figure D.40 SP function block

Parameter	Default
IN	0
LIM_INDICATOR	0
OUT	0
OUT_HI_ACT	0
OUT_HI_LIM	100
OUT_LO_ACT	0
OUT_LO_LIM	50

Figure D.41 LIM1 function block

Expression:

```
IF ABS('^/PID_IN_CV16/SP.CV'-'^/PID_IN_CV16/PV.CV') >= 1 THEN
'OUT1' := 'IN1';
'OUT2' := 'IN2';
ELSE
'OUT1' := 'IN1';
ENDIF;

IF '^/PID_DE_CV12/MODE.TARGET' = MAN THEN
'OUT2' := 'IN2';
ENDIF;
```

Figure D.42 CALC1 function block

D.11 Pressurizer Pressure

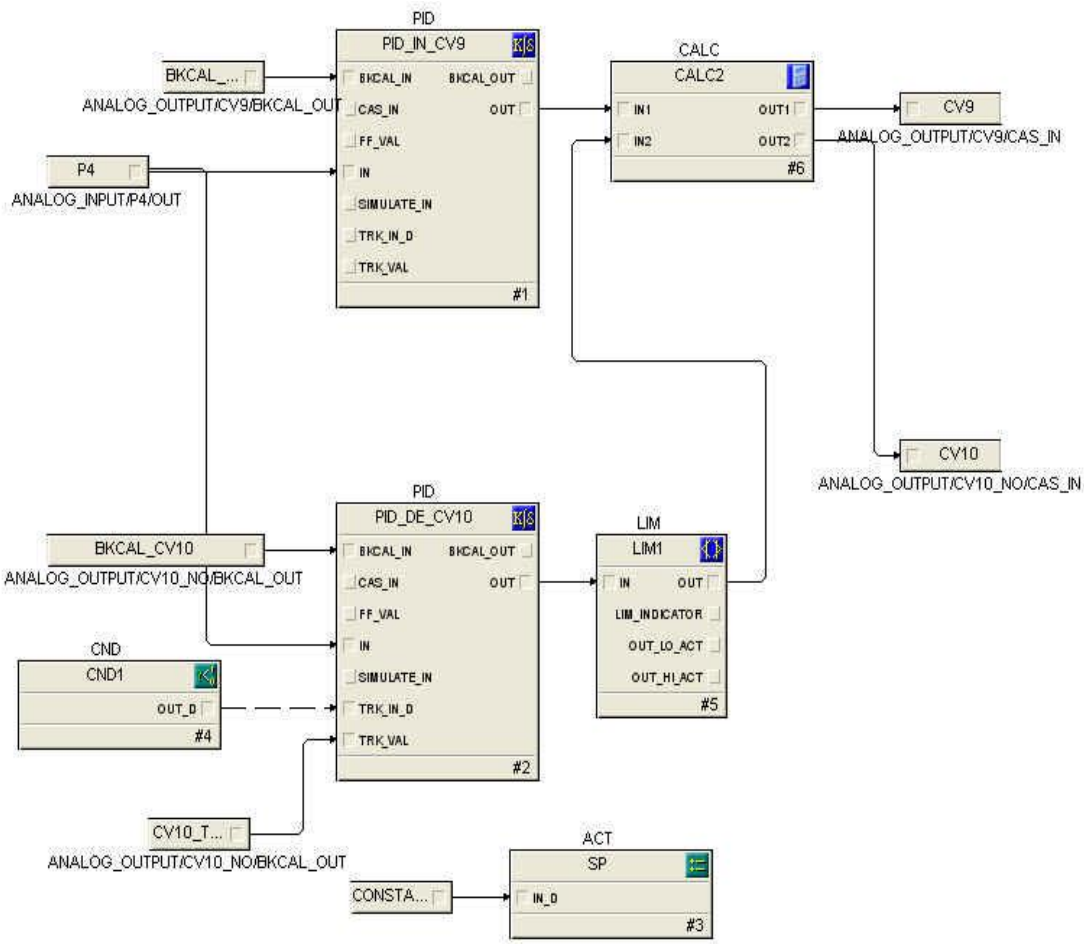

Figure D.43 Pressurizer pressure control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	1.3
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 25.0 ...
RATE	0
RESET	21.3
SP_FTIME	0
SP_HI_LIM	25
SP_LO_LIM	0

Figure D.44 PID_IN_CV9 tuning parameters

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	1
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 25.0 ...
RATE	0
RESET	20
SP_FTIME	0
SP_HI_LIM	25
SP_LO_LIM	0

Figure D.45 PID_DE_CV10 tuning parameters

Expression:

```
'^/PID_DE_CV10/SP.CV' := '^/PID_IN_CV9/BKCAL_OUT.CV'
```

Figure D.46 SP function block

Expression:

```
ABS('^/PID_IN_CV9/SP.CV'-'^/PID_IN_CV9/PV.CV')  < 0.5
```

Figure D.47 CND1 function block

Parameter	Default
IN	0
LIM_INDICATOR	0
OUT	0
OUT_HI_ACT	0
OUT_HI_LIM	100
OUT_LO_ACT	0
OUT_LO_LIM	70

Figure D.48 LIM1 function block

Expression:

```
IF ABS('^/PID_IN_CV9/SP.CV'-'^/PID_IN_CV9/PV.CV') >= 0.5 THEN
'OUT1' := 'IN1';
'OUT2' := 'IN2';
ELSE
'OUT1' := 'IN1';
ENDIF;

IF '^/PID_DE_CV10/MODE.TARGET' = MAN THEN
'OUT2' := 'IN2';
ENDIF;
```

Figure D.49 CALC2 function block

D.12 Primary Water Flow

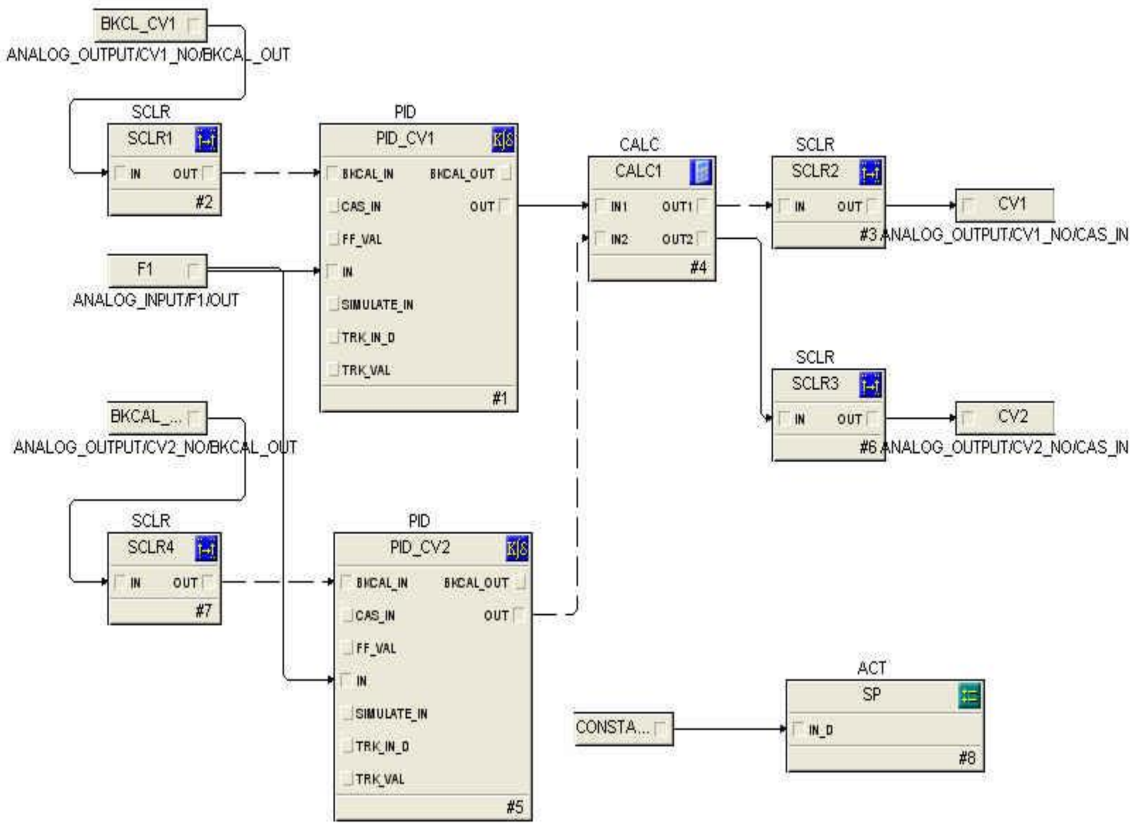

Figure D 50 Primary water flow control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	0.6
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 10.0 ...
RATE	0
RESET	16.8
SP_FTIME	0
SP_HI_LIM	10
SP_LO_LIM	0

Figure D.51 PID_CV1 tuning parameters

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	0.5
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 10.0 ...
RATE	0
RESET	20
SP_FTIME	0
SP_HI_LIM	10
SP_LO_LIM	0

Figure D.52 PID_CV2 tuning parameters

Expression:

```
'^/PID_CV2/SP.CV' := '^/PID_CV1/BKCAL_OUT.CV'
```

Figure D.53 SP function block

Parameter	Default
IN	0
IN_SCALE	0.0 to 100.0...
OUT	0
OUT_SCALE	100.0 to 0.0...

Figure D.54 SCLRs function block

Expression:

```
IF '//ANALOG_OUTPUT/CV1_NO/OUT.CV' <= 20 AND '//ANALOG_INPUT/F1/OUT.CV' <= 4 THEN
'OUT2.CV' := 'IN2.CV';
ENDIF;

IF'^/PID_CV2/MODE.TARGET' = MAN THEN
'OUT2.CV' := 'IN2.CV';
ENDIF;

'OUT1.CV' := 'IN1.CV'
```

Figure D.55 CALC1 function block

D.13 Ratio

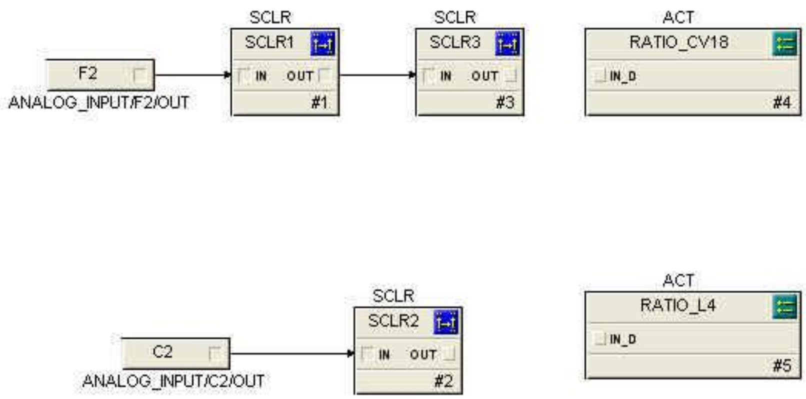

Figure D.56 Ratio control

Expression:

```
1 '//ANALOG_OUTPUT/CV18_NO/OUT.CV':= '^/SCLR3/OUT.CV'
```

Figure D.57 RATIO_CV18 function block

Parameter	Default
IN	0
IN_SCALE	0.0 to 5.0
OUT	0
OUT_SCALE	40.0 to 80.0 %

Figure D.58 SCLR1 function block

Parameter	Default
IN	0
IN_SCALE	0.0 to 100.0
OUT	0
OUT_SCALE	100.0 to 0.0 %

Figure D.59 SCLR3 function block

Expression:

```
1 '//SG_HX_TANK_L/PID1/SP.CV':= '^/SCLR2/OUT.CV'
```

Figure D.60 RATIO_L4 function block

Parameter	Default
IN	0
IN_SCALE	0.0 to 50.0
OUT	0
OUT_SCALE	35.0 to 60.0 %

Figure D.61 SCLR2 function block

D.14 Safety

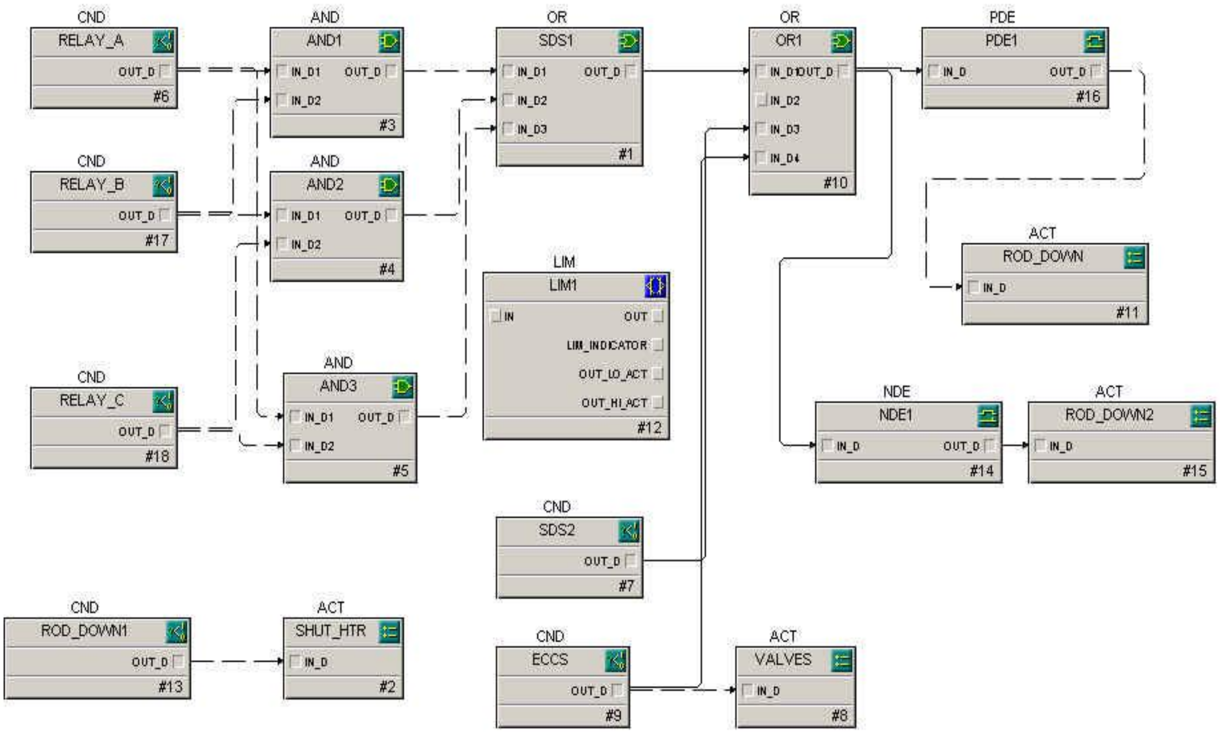

Figure D.62 Safety control

Expression:

```
'//WATER_HEATING/T2_AVG/OUT.CV' > 32 OR
 ( '//ANALOG_INPUT/L4/OUT.CV' < 25 AND  '//OPERATION_MODE/RUN.CV' = 1) OR
'//ANALOG_INPUT/L4/OUT.CV'  > 101
```

Figure D.63 RELAY A function block

Expression:

```
'//WATER_HEATING/T2_AVG/OUT.CV' > 34 OR
 ( '//ANALOG_INPUT/L4/OUT.CV' < 23 AND  '//OPERATION_MODE/RUN.CV' = 1) OR
'//ANALOG_INPUT/L4/OUT.CV'  > 103
```

Figure D.64 RELAY B function block

Expression:

```
'//WATER_HEATING/T2_AVG/OUT.CV' > 36 OR
 ( '//ANALOG_INPUT/L4/OUT.CV' < 20 AND  '//OPERATION_MODE/RUN.CV' = 1) OR
'//ANALOG_INPUT/L4/OUT.CV'  > 105
```

Figure D.65 RELAY C function block

Expression:

```
'//DISCRETE_OUTPUT/C2/OUT_D.CV' := 0;
```

Figure D.66 SHUT_HTR function block

Expression:

```
'//ANALOG_INPUT/L4/OUT.CV' > 108 OR
'//ANALOG_INPUT/T2_1/OUT.CV'>'^/LIM1/OUT_HI_LIM.CV' OR
'//ANALOG_INPUT/T2_2/OUT.CV'>'^/LIM1/OUT_HI_LIM.CV' OR
'//ANALOG_INPUT/T2_3/OUT.CV'>'^/LIM1/OUT_HI_LIM.CV' OR
'//ANALOG_INPUT/P1/OUT.CV' > 17
```

Figure D.67 SDS2 function block

Expression:

```
('^/SDS1/OUT_D.CV' = 1 OR
'^/SDS2/OUT_D.CV' = 1) AND
'//DISCRETE_OUTPUT/C1/OUT_D.CV' = 0
```

Figure D.68 ECCS function block

Expression:

```
'//UPPER_TANK_L/PID_DE_CV15/MODE.TARGET' := MAN;
'//UPPER_TANK_L/PID_DE_CV15/OUT.CV' := 100;
'//ANALOG_OUTPUT/CV19_NO/OUT.CV' := 0;
'//DISCRETE_OUTPUT/CV20_NO/OUT_D.CV' := 0;
'//PRIMARY_WATER_F/PID_CV1/MODE.TARGET' := MAN;
'//PRIMARY_WATER_F/PID_CV1/OUT.CV' := 0;
'//PRIMARY_WATER_F/PID_CV2/MODE.TARGET' := MAN;
'//PRIMARY_WATER_F/PID_CV2/OUT.CV' := 0;
```

Figure D.69 VALVES function block

Parameter	Default	Linked
OUT_HI_LIM	45	
OUT_LO_LIM	0	

Figure D.70 LIM1 function block

Expression:

```
'//DISCRETE_OUTPUT/ROD_DOWN/OUT_D.CV' := 1
```

Figure D.71 ROD_DOWN function block

Expression:

```
'//DISCRETE_INPUT/SDS2/OUT_D.CV'=0
```

Figure D.72 ROD_DOWN1 function block

Expression:

```
'//DISCRETE_OUTPUT/ROD_DOWN/OUT_D.CV' := 0
```

Figure D.73 ROD_DOWN2 function block

D.15 Sequence

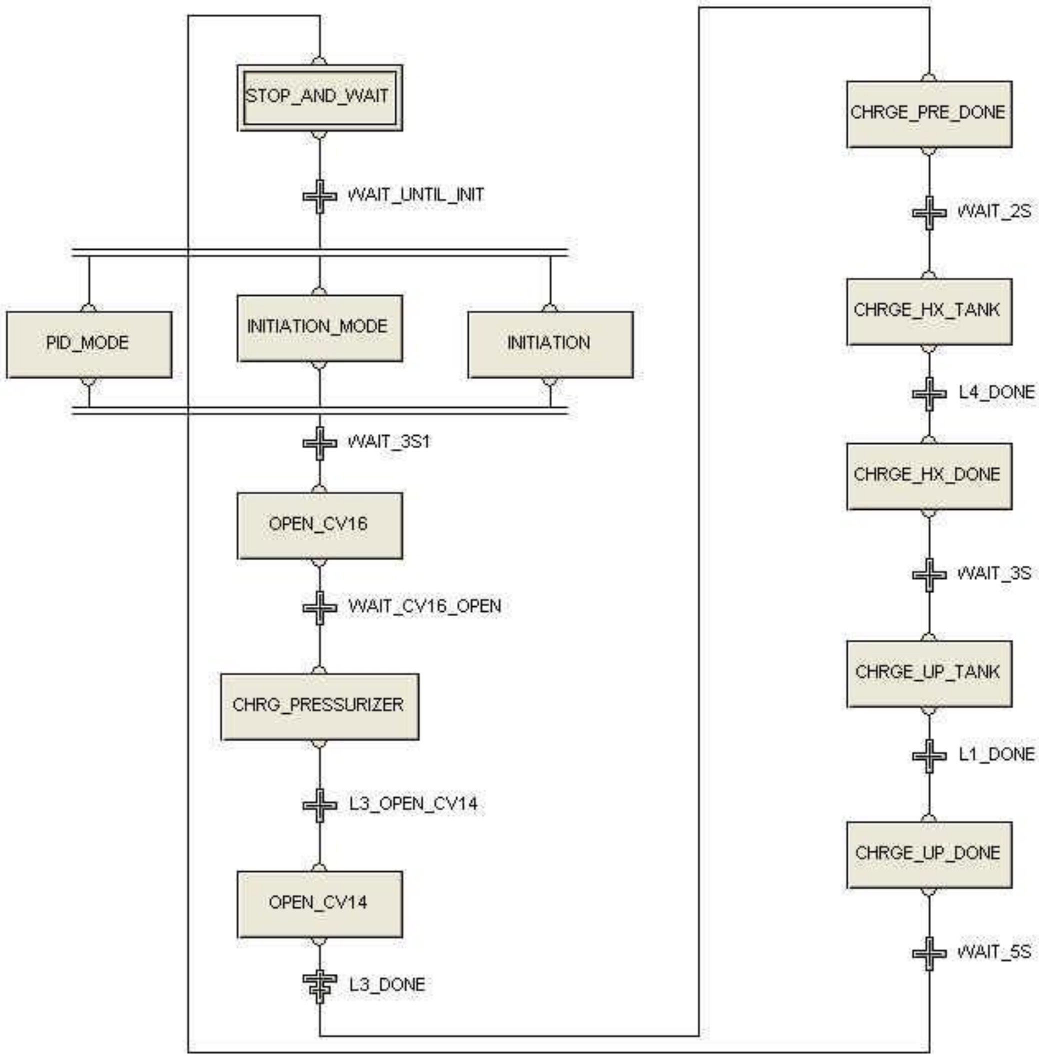

Figure D.72 Sequence control

Action	Text
SET_SP_TO_IDLE	'SP' := 'SFCCTRL:IDLE';

Figure D.73 STOP_AND_WAIT function block

Transition Condition

'SP'= 'SFCCTRL:START'

Figure D.74 WAIT_UNTIL_INIT function block

Action	Text
A1	'//AIR_F/PID1/MODE.TARGET':= MAN
A2	'//AIR_P/PID1/MODE.TARGET':= MAN
A3	'//HX_OUTLET_T/PID1/MODE.TARGET' := MAN
A4	'//PRESSURIZER_L/PID_DE_CV12/MODE.TARGET':= MAN
A5	'//PRESSURIZER_L/PID_IN_CV16/MODE.TARGET':= MAN
A6	'//PRESSURIZER_P/PID_DE_CV10/MODE.TARGET':= MAN
A7	'//PRESSURIZER_P/PID_IN_CV9/MODE.TARGET':= MAN
A8	'//PRIMARY_WATER_F/PID_CV1/MODE.TARGET':= MAN
A9	'//PRIMARY_WATER_F/PID_CV2/MODE.TARGET':= MAN
A10	'//SG_HX_TANK_L/PID1/MODE.TARGET':= MAN
A11	'//SG_HX_TANK_P/PID1/MODE.TARGET':= MAN
A12	'//UPPER_TANK_L/PID_DE_CV15/MODE.TARGET':= MAN
A13	'//UPPER_TANK_L/PID_IN_CV17/MODE.TARGET':= MAN
A14	'//WATER_HEATING/PID1/MODE.TARGET':= MAN
A15	'//WATER_INVENT_P/PID_DE_CV13/MODE.TARGET':= MAN
A16	'//WATER_INVENT_P/PID_IN_CV16/MODE.TARGET':= MAN

Figure D.75 PID_MODE function block

Action	Text
MODE_C1	'//ANALOG_OUTPUT/C1/MODE.TARGET':= MAN;
MODE_C5	'//ANALOG_OUTPUT/C5/MODE.TARGET' := MAN;
MODE_CV1	'//ANALOG_OUTPUT/CV1_NO/MODE.TARGET' := MAN;
MODE_CV4	'//ANALOG_OUTPUT/CV4_NO/MODE.TARGET' := MAN;
MODE_CV5	'//ANALOG_OUTPUT/CV5/MODE.TARGET' := MAN;
MODE_CV18	'//ANALOG_OUTPUT/CV18_NO/MODE.TARGET' := MAN;
MODE_CV9	'//ANALOG_OUTPUT/CV9/MODE.TARGET' := MAN;
MODE_CV13	'//ANALOG_OUTPUT/CV13_NO/MODE.TARGET' := MAN;
MODE_CV16	'//ANALOG_OUTPUT/CV16/MODE.TARGET' :=MAN;
MODE_CV21	'//ANALOG_OUTPUT/CV21/MODE.TARGET' := MAN;
MODE_CV34	'//ANALOG_OUTPUT/CV34/MODE.TARGET' := MAN;
MODE_CV15	'//ANALOG_OUTPUT/CV15_NO/MODE.TARGET' := MAN;
MODE_CV2	'//ANALOG_OUTPUT/CV2_NO/MODE.TARGET' := MAN;
MODE_CV10	'//ANALOG_OUTPUT/CV10_NO/MODE.TARGET' :=MAN;
MODE_CV12	'//ANALOG_OUTPUT/CV12_NO/MODE.TARGET' := MAN;
MODE_C2	'//ANALOG_OUTPUT/C2/MODE.TARGET' := MAN;
MODE_CV17	'//ANALOG_OUTPUT/CV17/MODE.TARGET' := MAN;
MODE_CV19	'//ANALOG_OUTPUT/CV19_NO/MODE.TARGET' := MAN;
MODE_CV25	'//ANALOG_OUTPUT/CV25/MODE.TARGET' := MAN;
MODE_CV8	'//ANALOG_OUTPUT/CV8_NO/MODE.TARGET' := MAN;
MODE_C1_D	'//DISCRETE_OUTPUT/C1/MODE.TARGET' := MAN;
MODE_C5_D	'//DISCRETE_OUTPUT/C5/MODE.TARGET' := MAN;
MODE_C2_D	'//DISCRETE_OUTPUT/C2/MODE.TARGET' := MAN;
MODE_CC_D	'//DISCRETE_OUTPUT/CC/MODE.TARGET' := MAN;
MODE_CV11	'//DISCRETE_OUTPUT/CV11/MODE.TARGET' :=MAN;
MODE_CV14	'//DISCRETE_OUTPUT/CV14/MODE.TARGET' := MAN;
MODE_CV3	'//DISCRETE_OUTPUT/CV3_NO/MODE.TARGET' := MAN;
MODE_CV20	'//DISCRETE_OUTPUT/CV20_NO/MODE.TARGET' := MAN;
MODE_DOWN	'//DISCRETE_OUTPUT/DOWN/MODE.TARGET' := MAN;
MODE_INTERM	'//DISCRETE_OUTPUT/INTERM/MODE.TARGET' := MAN;
MODE_ROD	'//DISCRETE_OUTPUT/ROD_DOWN/MODE.TARGET' := MAN;
MODE_HOME	'//DISCRETE_OUTPUT/HOME/MODE.TARGET' := MAN;

Figure D.76 INITIATION_MODE function block

Action	Text
OPEN_CV1	'//ANALOG_OUTPUT/CV1_NO/OUT.CV' := 0
OPEN_CV4	'//ANALOG_OUTPUT/CV4_NO/OUT.CV' := 0;
CLOSE_CV5	'//ANALOG_OUTPUT/CV5/OUT.CV' := 0;
OPEN_CV18	'//ANALOG_OUTPUT/CV18_NO/OUT.CV' := 0;
CLOSE_CV9	'//ANALOG_OUTPUT/CV9/OUT.CV' := 0;
OPEN_CV13	'//ANALOG_OUTPUT/CV13_NO/OUT.CV' := 0;
CLOSE_CV16	'//ANALOG_OUTPUT/CV16/OUT.CV' := 0;
CLOSE_CV21	'//ANALOG_OUTPUT/CV21/OUT.CV' := 0;
CLOSE_CV34	'//ANALOG_OUTPUT/CV34/OUT.CV' := 50;
OPEN_CV15	'//ANALOG_OUTPUT/CV15_NO/OUT.CV' := 0;
OPEN_CV2	'//ANALOG_OUTPUT/CV2_NO/OUT.CV' := 0;
OPEN_CV10	'//ANALOG_OUTPUT/CV10_NO/OUT.CV' := 0;
OPEN_CV12	'//ANALOG_OUTPUT/CV12_NO/OUT.CV' := 0;
STOP_C2	'//ANALOG_OUTPUT/C2/OUT.CV' := 0;
CLOSE_CV17	'//ANALOG_OUTPUT/CV17/OUT.CV' := 0;
OPEN_CV19	'//ANALOG_OUTPUT/CV19_NO/OUT.CV' := 0;
CLOSE_CV25	'//ANALOG_OUTPUT/CV25/OUT.CV' := 0
OPEN_CV8	'//ANALOG_OUTPUT/CV8_NO/OUT.CV' := 0;
STOP_C1_D	'//DISCRETE_OUTPUT/C1/OUT_D.CV':= 0;
STOP_C5_D	'//DISCRETE_OUTPUT/C5/OUT_D.CV':= 0;
STOP_C2_D	'//DISCRETE_OUTPUT/C2/OUT_D.CV':= 0;
STOP_CC_D	'//DISCRETE_OUTPUT/CC/OUT_D.CV':= 0;
CLOSE_CV11	'//DISCRETE_OUTPUT/CV11/OUT_D.CV':= 0;
CLOSE_CV14	'//DISCRETE_OUTPUT/CV14/OUT_D.CV':= 0;
OPEN_CV3	'//DISCRETE_OUTPUT/CV3_NO/OUT_D.CV':= 0;
OPEN_CV20	'//DISCRETE_OUTPUT/CV20_NO/OUT_D.CV':= 0;
SET_DOWN	'//DISCRETE_OUTPUT/DOWN/OUT_D.CV':= 0;
SET_INTERM	'//DISCRETE_OUTPUT/INTERM/OUT_D.CV':= 0;
SET_ROD	'//DISCRETE_OUTPUT/ROD_DOWN/OUT_D.CV':= 0;
SET_HOME	'//DISCRETE_OUTPUT/HOME/OUT_D.CV' := 0

Figure D.77 INITIATION function block

Transition Condition

'INITIATION_MODE/TIME.CV' > 3

Figure D.78 WAIT_3S1 function block

Action	Text
OPEN_CV16	'//ANALOG_OUTPUT/CV16/MODE.TARGET' := MAN;...
A1	'//ANALOG_OUTPUT/CV18_NO/OUT.CV' := 0;

Figure D.79.1 OPEN_CV16 function block

Action expression:

```
'//ANALOG_OUTPUT/CV16/MODE.TARGET' := MAN;
'//ANALOG_OUTPUT/CV16/OUT.CV' := 100
```

Figure D.79.2 OPEN_CV16 function block

Transition Condition

```
'OPEN_CV16/TIME.CV' >80
```

Figure D.80 WAIT_CV16_OPEN function block

Action	Text
CLOSE_CV17	'//ANALOG_OUTPUT/CV17/OUT.CV':= 0
CLOSE_CV25	'//ANALOG_OUTPUT/CV25/OUT.CV' := 0
CLOSE_CV14	'//DISCRETE_OUTPUT/CV14/OUT_D.CV' := 0
CLOSE_CV3	'//DISCRETE_OUTPUT/CV3_NO/OUT_D.CV' := 1
CLOSE_CV20	'//DISCRETE_OUTPUT/CV20_NO/OUT_D.CV' := 1
CLOSE_CV12	'//ANALOG_OUTPUT/CV12_NO/OUT.CV' := 100
CLOSE_CV13	'//ANALOG_OUTPUT/CV13_NO/OUT.CV' := 100
CLOSE_CV9	'//ANALOG_OUTPUT/CV9/OUT.CV' := 0
CLOSE_CV15	'//ANALOG_OUTPUT/CV15_NO/OUT.CV' := 100
CLOSE_CV18	'//ANALOG_OUTPUT/CV18_NO/OUT.CV' := 100
CLOSE_CV19	'//ANALOG_OUTPUT/CV19_NO/OUT.CV' := 100
OPEN_CV16	'//ANALOG_OUTPUT/CV16/OUT.CV' := 100
OPEN_CV1	'//ANALOG_OUTPUT/CV1_NO/OUT.CV' := 0
OPEN_CV2	'//ANALOG_OUTPUT/CV2_NO/OUT.CV' := 0
OPEN_CV10	'//ANALOG_OUTPUT/CV10_NO/OUT.CV' := 0
OPEN_CV11	'//DISCRETE_OUTPUT/CV11/OUT_D.CV' := 1
RUN_PUMP2	'//DISCRETE_OUTPUT/C5/OUT_D.CV' := 1
PUMP2_100	'//ANALOG_OUTPUT/C5/OUT.CV' := 100

Figure D.81 CHRG_PRESSURIZER function block

Transition Condition

```
'//ANALOG_INPUT/L3/OUT.CV' >8
```

Figure D.82 L3_OPEN_CV14 function block

Action	Text
A1	'//DISCRETE_OUTPUT/CV14/OUT_D.CV' := 1

Figure D.83 OPEN_CV14 function block

Transition Condition

'//ANALOG_INPUT/L3/OUT.CV' > 15

Figure D.84 L3_DONE function block

Action	Text
CLOSE_CV10	'//ANALOG_OUTPUT/CV10_NO/OUT.CV' := 100
CLOSE_CV16	'//ANALOG_OUTPUT/CV16/OUT.CV' := 0

Figure D.85 CHRGE_PRE_DONE function block

Transition Condition

'CHRGE_PRE_DONE/TIME.CV' >5

Figure D.86 WAIT_2S function block

Action	Text
OPEN_CV25	'//ANALOG_OUTPUT/CV25/OUT.CV' := 100
OPEN_CV5	'//ANALOG_OUTPUT/CV5/OUT.CV' := 100

Figure D.87 CHRGE_HX_TANK function block

Transition Condition

'//ANALOG_INPUT/L4/OUT.CV' > 20

Figure D.88 L4_DONE function block

Action	Text
OPEN_CV18	'//ANALOG_OUTPUT/CV18_NO/OUT.CV' := 60
CV25_CAS	'//ANALOG_OUTPUT/CV25/OUT.CV' := 10;

Figure D.89 CHRGE_HX_DONE function block

Transition Condition

'CHRGE_HX_DONE/TIME.CV' > 3

Figure D.90 WAIT_3S function block

Action	Text
A1	'//ANALOG_OUTPUT/CV17/OUT.CV' := 100
A2	'//ANALOG_OUTPUT/CV25/MODE.TARGET' := CAS
A3	'//SG_HX_TANK_L/PID1/MODE.TARGET' := AUTO
A4	'//SG_HX_TANK_L/PID1/SP.CV' := 40

Figure D.91 CHRGE_UP_TANK function block

Transition Condition

'//ANALOG_INPUT/L1/OUT.CV' > 20

Figure D.92 L1_DONE function block

Action	Text
CLOSE_CV17	'//ANALOG_OUTPUT/CV17/OUT.CV' := 0
CLOSE_CV2	'//ANALOG_OUTPUT/CV2_NO/OUT.CV' := 100
A2	'//SG_HX_TANK_L/PID1/SP.CV' := 35

Figure D.93 CHRGE_UP_DONE function block

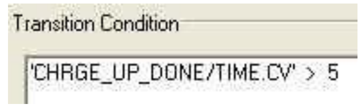

Figure D.94 WAIT_5S function block

D.16 HX Tank Level

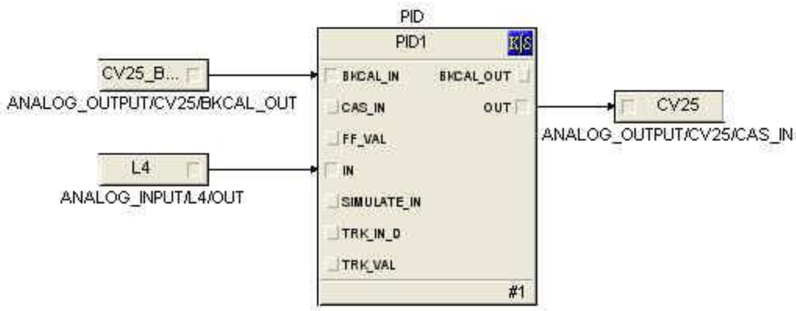

Figure D.95 HX tank level control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	3.47
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 100.0 ...
RATE	13.2
RESET	252
SP_FTIME	0
SP_HI_LIM	100
SP_LO_LIM	0

Figure D.96 PID tuning parameters

D.17 HX Tank Pressure

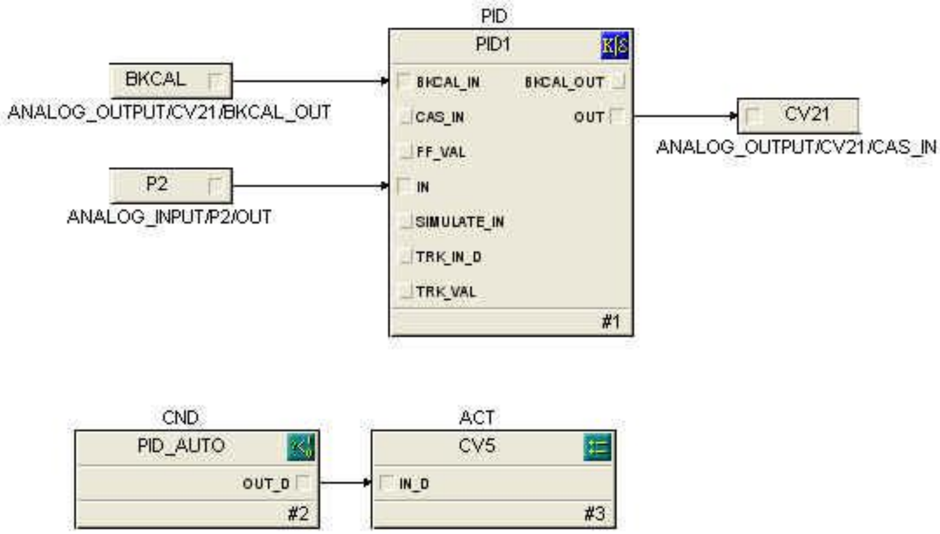

Figure D.97 HX tank pressure control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	0.8
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 25.0...
RATE	20
RESET	45
SP_FTIME	0
SP_HI_LIM	25
SP_LO_LIM	0

Figure D.98 PID tuning parameters

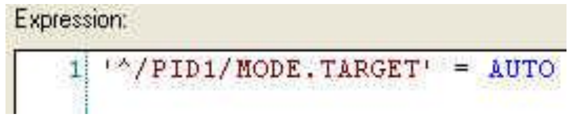

Figure D.99 PID_AUTO function block

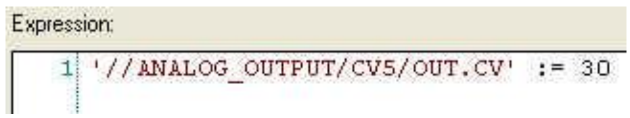

Figure D.100 CV5 function block

D.18 Stop System

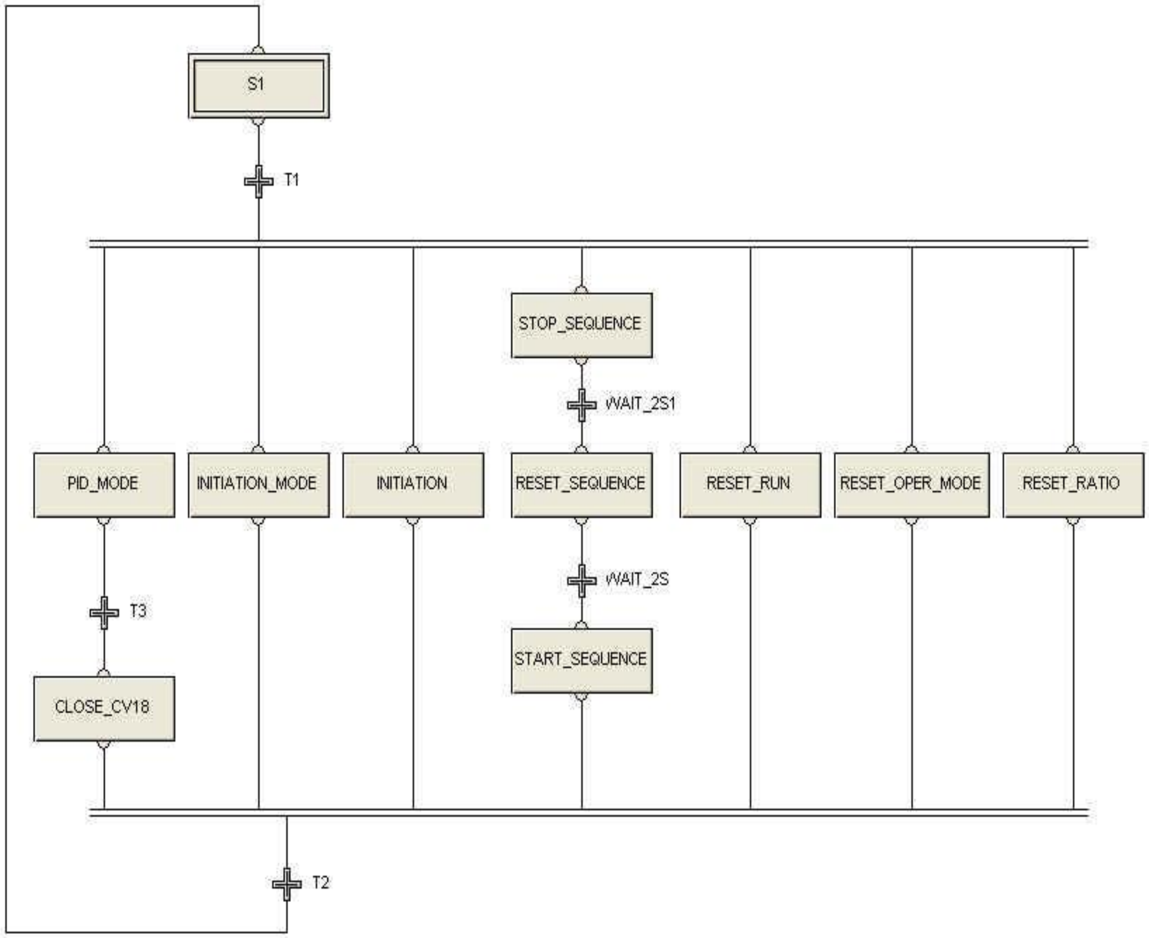

Figure D.101 Stop system control

Parameter	Default
ABNORM_AC...	False
AUTO_ADVA...	True
BAD_ACTIVE	False
BLOCK_ERR	
COMMAND	Start Sequence
CONFIRM_FAIL	False
ERROR	False
EXEC_TIME	0
MCOMMAND	In Service
MERROR	
MSTATE	In Service
MSTATUS	
RERROR	False
SP	START
STATE	Sequence Active
STATUS	
TIME	0
VERSION	1

Figure D.102 Parameters

Action	Text
SET_TO_IDLE	'SP' := 'SFCCTRL:IDLE'

Figure D.103 S1 function block

Transition Condition
'SP' = 'SFCCTRL:START'

Figure D.104 T1 function block

Action	Text
A1	'//AIR_F/PID1/MODE.TARGET':= MAN
A2	'//AIR_P/PID1/MODE.TARGET':= MAN
A3	'//HX_OUTLET_T/PID1/MODE.TARGET' := MAN
A4	'//PRESSURIZER_L/PID_DE_CV12/MODE.TARGET':= MAN
A5	'//PRESSURIZER_L/PID_IN_CV16/MODE.TARGET':= MAN
A6	'//PRESSURIZER_P/PID_DE_CV10/MODE.TARGET':= MAN
A7	'//PRESSURIZER_P/PID_IN_CV9/MODE.TARGET':= MAN
A8	'//PRIMARY_WATER_F/PID_CV1/MODE.TARGET':= MAN
A9	'//PRIMARY_WATER_F/PID_CV2/MODE.TARGET':= MAN
A10	'//SG_HX_TANK_L/PID1/MODE.TARGET':= MAN
A11	'//SG_HX_TANK_P/PID1/MODE.TARGET':= MAN
A12	'//UPPER_TANK_L/PID_DE_CV15/MODE.TARGET':= MAN
A13	'//UPPER_TANK_L/PID_IN_CV17/MODE.TARGET':= MAN
A14	'//WATER_HEATING/PID1/MODE.TARGET':= MAN
A15	'//WATER_INVENT_P/PID_DE_CV13/MODE.TARGET':= MAN
A16	'//WATER_INVENT_P/PID_IN_CV16/MODE.TARGET':= MAN

Figure D.105 PID_MODE function block

Transition Condition

'PID_MODE/TIME.CV' > 2

Figure D.106 T3 function block

Action	Text
A1	'//ANALOG_OUTPUT/CV18_NO/OUT.CV' := 0;

Figure D.107 CLOSE_CV18 function block

Action	Text
MODE_C1	'//ANALOG_OUTPUT/C1/MODE.TARGET':= MAN;
MODE_C5	'//ANALOG_OUTPUT/C5/MODE.TARGET' := MAN;
MODE_CV1	'//ANALOG_OUTPUT/CV1_NO/MODE.TARGET' := MAN;
MODE_CV4	'//ANALOG_OUTPUT/CV4_NO/MODE.TARGET' := MAN;
MODE_CV5	'//ANALOG_OUTPUT/CV5/MODE.TARGET' := MAN;
MODE_CV18	'//ANALOG_OUTPUT/CV18_NO/MODE.TARGET' := MAN;
MODE_CV9	'//ANALOG_OUTPUT/CV9/MODE.TARGET' := MAN;
MODE_CV13	'//ANALOG_OUTPUT/CV13_NO/MODE.TARGET' := MAN;
MODE_CV16	'//ANALOG_OUTPUT/CV16/MODE.TARGET' :=MAN;
MODE_CV21	'//ANALOG_OUTPUT/CV21/MODE.TARGET' := MAN;
MODE_CV34	'//ANALOG_OUTPUT/CV34/MODE.TARGET' := MAN;
MODE_CV15	'//ANALOG_OUTPUT/CV15_NO/MODE.TARGET' := MAN;
MODE_CV2	'//ANALOG_OUTPUT/CV2_NO/MODE.TARGET' := MAN;
MODE_CV10	'//ANALOG_OUTPUT/CV10_NO/MODE.TARGET' := MAN;
MODE_CV12	'//ANALOG_OUTPUT/CV12_NO/MODE.TARGET' := MAN;
MODE_C2	'//ANALOG_OUTPUT/C2/MODE.TARGET' := MAN;
MODE_CV17	'//ANALOG_OUTPUT/CV17/MODE.TARGET' := MAN;
MODE_CV19	'//ANALOG_OUTPUT/CV19_NO/MODE.TARGET' := MAN;
MODE_CV25	'//ANALOG_OUTPUT/CV25/MODE.TARGET' := MAN;
MODE_CV8	'//ANALOG_OUTPUT/CV8_NO/MODE.TARGET' := MAN;
MODE_C1_D	'//DISCRETE_OUTPUT/C1/MODE.TARGET' := MAN;
MODE_C5_D	'//DISCRETE_OUTPUT/C5/MODE.TARGET' := MAN;
MODE_C2_D	'//DISCRETE_OUTPUT/C2/MODE.TARGET' := MAN;
MODE_CC_D	'//DISCRETE_OUTPUT/CC/MODE.TARGET' := MAN;
MODE_CV11	'//DISCRETE_OUTPUT/CV11/MODE.TARGET' :=MAN;
MODE_CV14	'//DISCRETE_OUTPUT/CV14/MODE.TARGET' := MAN;
MODE_CV3	'//DISCRETE_OUTPUT/CV3_NO/MODE.TARGET' := MAN;
MODE_CV20	'//DISCRETE_OUTPUT/CV20_NO/MODE.TARGET' := MAN;
MODE_DOWN	'//DISCRETE_OUTPUT/DOWN/MODE.TARGET' := MAN;
MODE_INTERM	'//DISCRETE_OUTPUT/INTERM/MODE.TARGET' := MAN;
MODE_ROD	'//DISCRETE_OUTPUT/ROD_DOWN/MODE.TARGET' := MAN;
MODE_HOME	'//DISCRETE_OUTPUT/HOME/MODE.TARGET' := MAN;

Figure D.108 INITIATION_MODE function block

Action	Text
OPEN_CV1	'//ANALOG_OUTPUT/CV1_NO/OUT.CV' := 0
OPEN_CV4	'//ANALOG_OUTPUT/CV4_NO/OUT.CV' := 0;
OPEN_CV5	'//ANALOG_OUTPUT/CV5/OUT.CV' := 100;
OPEN_CV18	'//ANALOG_OUTPUT/CV18_NO/OUT.CV' := 0;
CLOSE_CV9	'//ANALOG_OUTPUT/CV9/OUT.CV' := 0;
OPEN_CV13	'//ANALOG_OUTPUT/CV13_NO/OUT.CV' := 0;
CLOSE_CV16	'//ANALOG_OUTPUT/CV16/OUT.CV' := 0;
CLOSE_CV21	'//ANALOG_OUTPUT/CV21/OUT.CV' := 0;
CLOSE_CV34	'//ANALOG_OUTPUT/CV34/OUT.CV' := 50;
OPEN_CV15	'//ANALOG_OUTPUT/CV15_NO/OUT.CV' := 0;
OPEN_CV2	'//ANALOG_OUTPUT/CV2_NO/OUT.CV' := 0;
OPEN_CV10	'//ANALOG_OUTPUT/CV10_NO/OUT.CV' := 0;
OPEN_CV12	'//ANALOG_OUTPUT/CV12_NO/OUT.CV' := 0;
STOP_C2	'//ANALOG_OUTPUT/C2/OUT.CV' := 0;
CLOSE_CV17	'//ANALOG_OUTPUT/CV17/OUT.CV' := 0;
OPEN_CV19	'//ANALOG_OUTPUT/CV19_NO/OUT.CV' := 0;
CLOSE_CV25	'//ANALOG_OUTPUT/CV25/OUT.CV' := 0
OPEN_CV8	'//ANALOG_OUTPUT/CV8_NO/OUT.CV' := 0;
STOP_C1_D	'//DISCRETE_OUTPUT/C1/OUT_D.CV':= 0;
STOP_C5_D	'//DISCRETE_OUTPUT/C5/OUT_D.CV':= 0;
STOP_C2_D	'//DISCRETE_OUTPUT/C2/OUT_D.CV':= 0;
STOP_CC_D	'//DISCRETE_OUTPUT/CC/OUT_D.CV':= 0;
CLOSE_CV11	'//DISCRETE_OUTPUT/CV11/OUT_D.CV':= 1;
CLOSE_CV14	'//DISCRETE_OUTPUT/CV14/OUT_D.CV':= 0;
OPEN_CV3	'//DISCRETE_OUTPUT/CV3_NO/OUT_D.CV':= 0;
OPEN_CV20	'//DISCRETE_OUTPUT/CV20_NO/OUT_D.CV':= 0;
SET_DOWN	'//DISCRETE_OUTPUT/DOWN/OUT_D.CV':= 0;
SET_INTERM	'//DISCRETE_OUTPUT/INTERM/OUT_D.CV':= 0;
SET_ROD	'//DISCRETE_OUTPUT/ROD_DOWN/OUT_D.CV':= 0;
SET_HOME	'//DISCRETE_OUTPUT/HOME/OUT_D.CV' := 0

Figure D.109 INITIATION function block

Action	Text
STOP_SEQUENCE	'//SEQUENCE/COMMAND.CV' := 1

Figure D.110 STOP_SEQUENCE function block

Transition Condition
'STOP_SEQUENCE/TIME.CV' > 2

Figure D.111 WAIT_2S1 function block

Action	Text
RESET_SEQUENCE	'//SEQUENCE/COMMAND.CV' := 5

Figure D.112 RESET_SEQUENCE function block

Transition Condition

```
'RESET_SEQUENCE/TIME.CV' > 2
```

Figure D.113 WAIT_2S function block

Action	Text
START_SEQUENCE	'//SEQUENCE/COMMAND.CV' := 0

Figure D.114 START_SEQUENCE function block

Action	Text
A1	'//OPERATION_MODE/RUN.CV' := 0
A2	'//OPERATION_MODE/STARTUP_DONE/OUT_D.CV' := 0

Figure D.115 RESET_RUN function block

Expression:

```
'//OPERATION_MODE/NORMAL.CV' := 0;
'//OPERATION_MODE/SOLID.CV' := 0;
'//OPERATION_MODE/OPEN.CV' := 0;
'//OPERATION_MODE/RESET_MODE.CV' := 0
```

Figure D.116 RESET_OPER_MODE function block

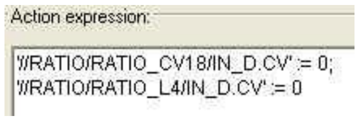

Figure D.117 RESET_RATIO function block

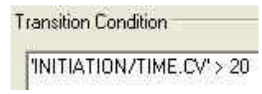

Figure D.118 T2 function block

D.19 Upper Tank Level

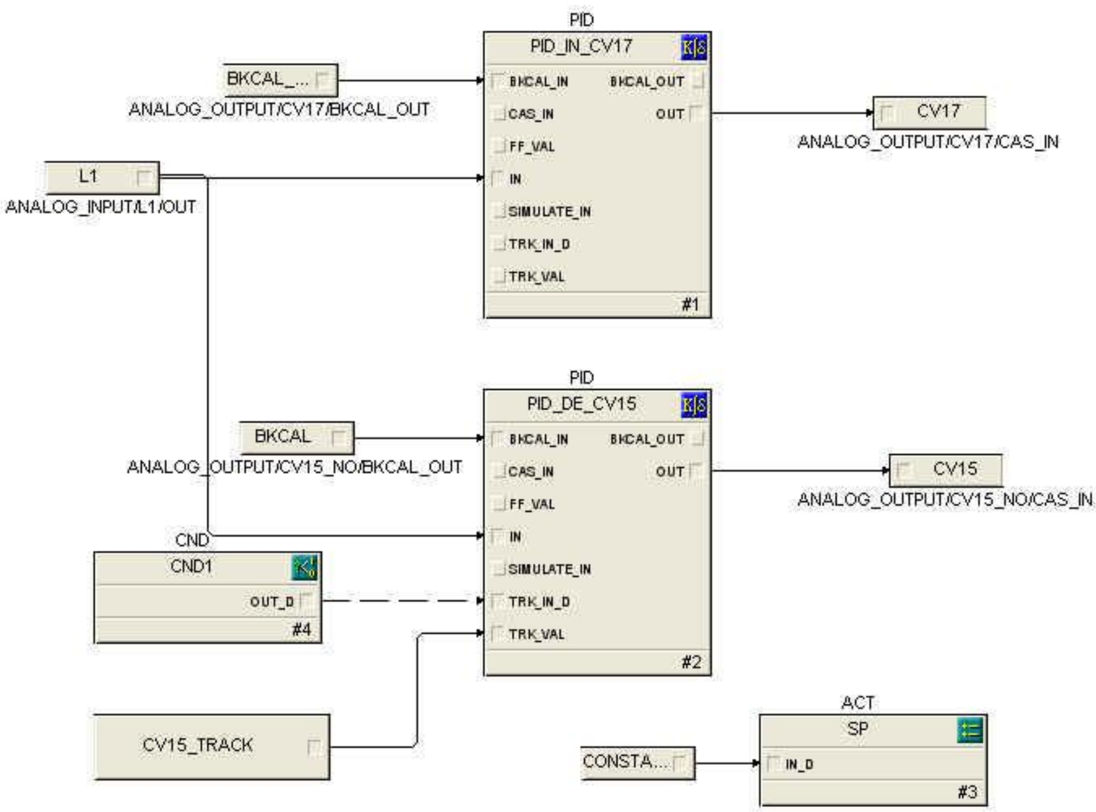

Figure D.119 Upper tank level control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	3
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 100.0...
RATE	10
RESET	250
SP_FTIME	0
SP_HI_LIM	100
SP_LO_LIM	0

Figure D.120 PID_CV17 tuning parameters

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	0.4
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 100.0...
RATE	0
RESET	100
SP_FTIME	0
SP_HI_LIM	100
SP_LO_LIM	0

Figure D.121 PID_CV15 tuning parameters

Expression:

```
1 '^/PID_DE_CV15/MODE.TARGET' = AUTO
```

Figure D.122 CND1 function block

Expression:

```
1  '^/PID_DE_CV15/SP.CV' := '^/PID_IN_CV17/SP.CV'
```

Figure D.123 SP function block

D.20 Water Heating

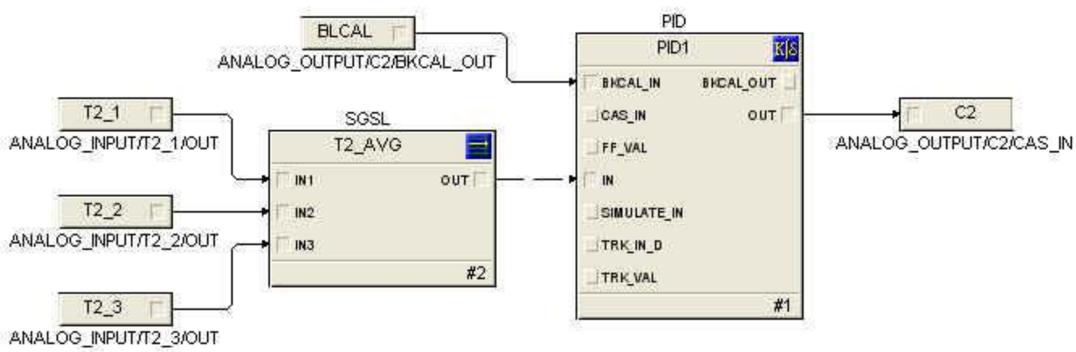

Figure D.124 Water heating control

Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	3
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 50.0 ...
RATE	0
RESET	30
SP_FTIME	0
SP_HI_LIM	50
SP_LO_LIM	0

Figure D.125 PID tuning parameters

D.21 Water Inventory Pressure

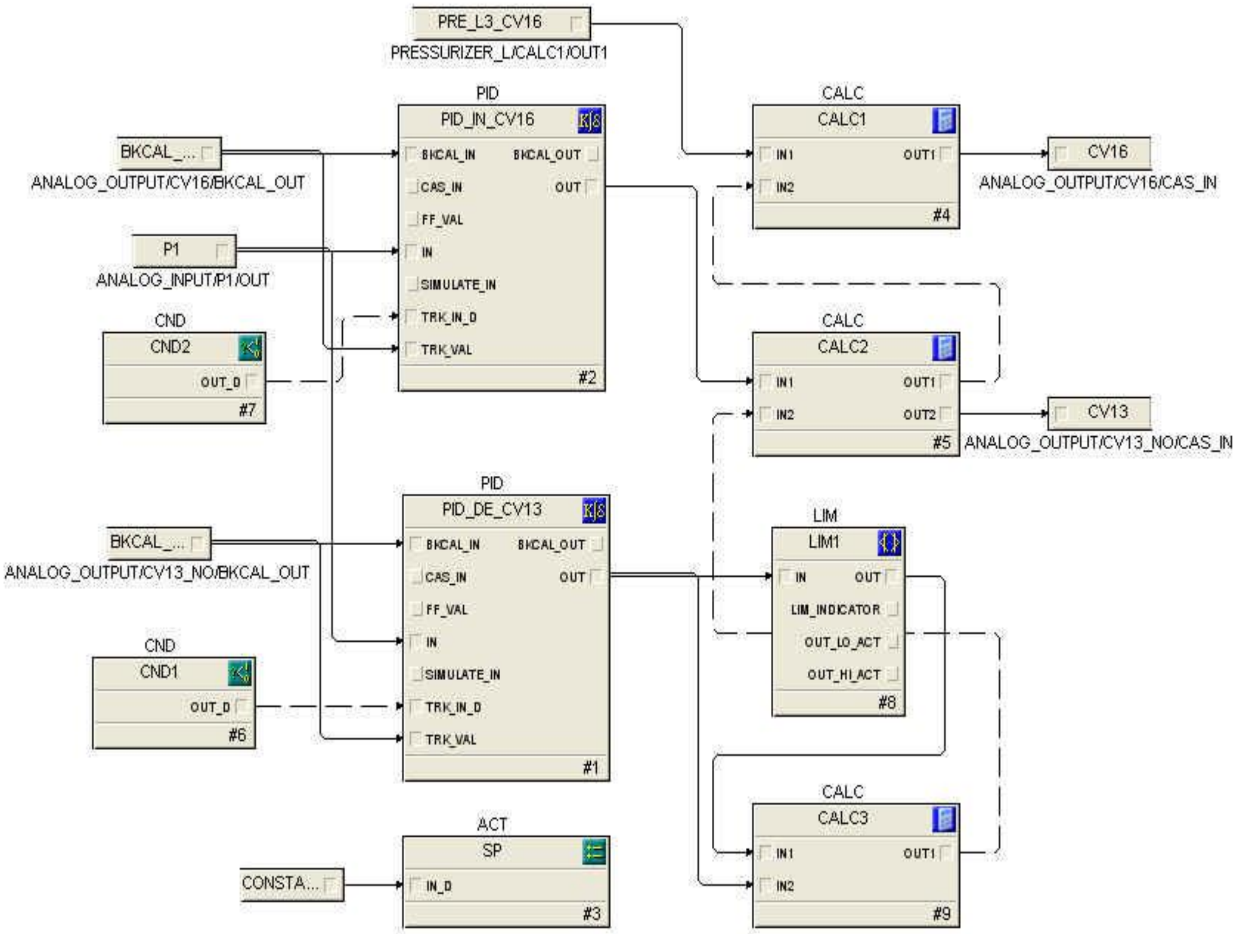

Figure D.126 Water inventory pressure control

⊟ Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	1.36
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 25.0 ...
RATE	16.4
RESET	65
SP_FTIME	0
SP_HI_LIM	25
SP_LO_LIM	0

Figure D.127 PID_CV16 tuning parameters

⊟ Tuning	
ARW_HI_...	100
ARW_LO_...	0
BAL_TIME	10
CONTROL...	Non-zero
GAIN	3.65
IDEADBAND	0
L_TYPE	Indirect
OUT_HI_LIM	100
OUT_LO_...	0
PV_FTIME	0
PV_SCALE	0.0 to 25.0 ...
RATE	6.6
RESET	26.5
SP_FTIME	0
SP_HI_LIM	25
SP_LO_LIM	0

Figure D.128 PID_CV13 tuning parameters

Expression:

```
ABS('^/PID_DE_CV13/SP.CV'-'^/PID_DE_CV13/PV.CV') < 0.6 AND ('^/PID_IN_CV16/PV.CV'  -  '^/PID_IN_CV16/SP.CV' > 0)
```

Figure D.129 CND2 function block

Expression:

```
ABS('^/PID_DE_CV13/SP.CV'-'^/PID_DE_CV13/PV.CV') < 0.6 AND ('^/PID_IN_CV16/PV.CV' - '^/PID_IN_CV16/SP.CV' <= 0)
```

Figure D.130 CND1 function block

Expression:

```
'^/PID_DE_CV13/SP.CV' := '^/PID_IN_CV16/BKCAL_OUT.CV'
```

Figure D.131 SP function block

Parameter	Default
IN	0
LIM_INDICATOR	0
OUT	0
OUT_HI_ACT	0
OUT_HI_LIM	100
OUT_LO_ACT	0
OUT_LO_LIM	70

Figure D.132 LIM1 function block

Expression:

```
IF '//OPERATION_MODE/SOLID.CV' = 1 THEN
'OUT1.CV' := 'IN1.CV';
ELSE
'OUT1.CV' := 'IN2.CV';
ENDIF
```

Figure D.133 CALC3 function block

Expression:

```
IF ABS('^/PID_DE_CV13/SP.CV'-'^/PID_DE_CV13/PV.CV') >= 0.6 THEN
'OUT1' := 'IN1';
'OUT2' := 'IN2';
ENDIF;

IF ABS('^/PID_DE_CV13/SP.CV'-'^/PID_DE_CV13/PV.CV') < 0.6 AND ('^/PID_IN_CV16/PV.CV' - '^/PID_IN_CV16/SP.CV'<=0) THEN
'OUT1' := 'IN1';
ENDIF;

IF ABS('^/PID_DE_CV13/SP.CV'-'^/PID_DE_CV13/PV.CV') < 0.6 AND ('^/PID_IN_CV16/PV.CV'  -  '^/PID_IN_CV16/SP.CV' >= 0) THEN
'OUT2' := 'IN2';
ENDIF;

IF '^/PID_IN_CV16/MODE.TARGET' = MAN THEN
'OUT1' := 'IN1';
ENDIF;

IF '^/PID_DE_CV13/MODE.TARGET' = MAN THEN
'OUT2' := 'IN2';
ENDIF;
```

Figure D.134 CALC2 function block

Expression:

```
IF '//OPERATION_MODE/SOLID_PULSE/IN_D.CV' = 1 THEN
'OUT1.CV' := 'IN2.CV'
ELSE
'OUT1.CV' := 'IN1.CV'
ENDIF
```

Figure D.135 CALC1 function block

Curriculum Vitae

Name:	Ximing Liu
Place of birth:	Shantou, Guangdong, China
Year of birth:	1968
Post-secondary Education and Degrees:	**Master of Engineering Science** May 2012 – April 2015 Department of Electrical and Computer Engineering The University of Western Ontario London, Ontario, Canada
	Graduate Diploma 1999-2001 International Law Dalian Maritime University Dalian, Liaoning, China
	Bachelor of Engineering Science 1987 – 1991 Environmental Engineering North China Electric Power University Baoding, Hebei, China

Publication:

X. L. Liu and J. Jiang, "Control Strategies in Supercritical Fossil Power Plant and Related Considerations for Canadian SCWR Power Plants," *The 6th International Symposium on Supercritical Water-Cooled Reactors (ISSCWR-6)*, 2013, China.