To the Graduate Council:

I am submitting herewith a thesis written by Aaron Patrick Sawyer entitled "Data-Based Modeling Methods for Fault Detection and Isolation in Heat Exchangers." I have examined the final paper copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Arthur E. Ruggles, Major Professor

We have read this thesis
and recommend its acceptance:

Belle R. Upadhyaya

Lawrence W. Townsend

Accepted for the Council:

Vice Chancellor and
Dean of Graduate Studies

Thesis
2004
.S39

DATA-BASED MODELING METHODS FOR FAULT DETECTION AND ISOLATION IN HEAT EXCHANGERS

A Thesis Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Aaron Patrick Sawyer

May 2004

ACKNOWLEDGMENTS

I would like to express my most sincere gratitude are great appreciation to all that helped me in obtaining my Master of Science degree. First, there is my advisor Dr. A. E. Ruggles, whose encouragement, insight, and sense of humor has been invaluable to me. Dr. B. R. Upadhyaya advised and guided me throughout my time at the University. Thanks to Dr. L. W. Townsend who serves on my committee and is an excellent teacher. Many thanks to Dr. H. L. Dodds, Head of the Nuclear Engineering Department for his invaluable assistance, advice and help throughout my years at the University.

During my time here, I was funded by a variety of excellent organizations. The Institute of Nuclear Power Operators funded my research through a National Academy of Nuclear Training scholarship. I would also like to thank Dr. Evren Eryurek, Dr. Kadir Kavaklioglu and John Miller with Emerson Process Management for their assistance in the research as well as funding. Additional support was provided by the SNS Experimental Facilities Division, including John Haines, who funded the last phase of my research at the University.

The contributions of my peers and colleagues cannot be understated. The open environment in the graduate student offices fostered many rewarding intellectual exchanges. I thank all the graduate students, but Martin Williamson, Jarrod Edwards and T. Jay Harrison deserve particular attention.

Finally, I wish to thank my parents, David and Debbie, whose love and support helped me through my undergraduate and graduate studies. And not the least, I thank Brandi, whose love and patience got me through the rough spots.

ABSTRACT

A multivariate analysis method is developed for processing measurements, and for detecting and isolating faults and monitoring performance degradation in heat exchanger control loops. A heat exchanger in a typical temperature-to-flow cascade loop is considered. A mechanistic thermal-fluid model for the components in the system is developed and compared to an installed laboratory heat exchanger control loop. A supplemental model for condenser heat transfer is included. The mechanistic model generates data to develop a data driven model using the Group Method of Data Handling (GMDH) approach. The GMDH model matches the mechanistic model well.

A Fault Detection and Isolation (FDI) rule-base is formulated from results of simulations performed using these models. The rule base allows the identification of faults in a heat exchanger control loop given suitable process measurements. The mechanistic model matches the physical system performance well and is used to create a Fault Detection and Isolation (FDI) algorithm for the system.

TABLE OF CONTENTS

Chapter 1. Introduction **1**

1.1 Problem Statement and Objectives 3

1.2 Heat Exchanger Review 4

1.2.1 Phase Change Categorization of Heat Exchangers 5

1.2.2 Flow Configuration Categorization of Heat Exchangers 6

1.2.3 Heat Exchanger Degradation 8

1.3 Contributions of Thesis Research 9

1.4 Organization of the Thesis 10

Chapter 2. Heat Transfer in Heat Exchangers **11**

2.1 Heat Transfer Coefficients 11

2.2 Calculation of Heat Transfer Coefficients 16

2.3 Accuracy of Heat Exchanger Model 20

2.4 Heat Transfer in Condensers 22

2.4.1 Theory 24

2.4.2 Vertical Tubes 26

2.4.3 Horizontal Tubes 30

2.4.4 Condenser Model Validation 32

2.5 Computer Modeling of Heat Exchanger Operation 34

Chapter 3. Heat Exchanger Control Loop Simulation **37**

3.1 Heat Exchanger Control Loop Configuration 37

3.2 Control Loop Components 40

3.3 Faults Induced in the Heat Exchanger Control Loop 42

3.4 Industrial Scaling for Control Loop 43

Chapter 4. Proposed Fault Detection and Isolation (FDI) Method 46

4.1 System Behavior During Faults 46

4.2 Extreme Values During Faults 48

4.3 Group Method of Data Handling (GMDH) Fault Detection 48

4.3.1. Brief Overview of the GMDH Algorithm 50

Chapter 5. Results and Analysis 54

5.1 Fault Outcome Tables 57

5.2 Group Method of Data Handling Fault Detection Results 57

Chapter 6. Summary, Conclusions, and Recommendations for Future Work 65

6.1 Summary 65

6.2 Conclusions 66

6.3 Recommendations for Future Work 67

List of References 68

Appendices 71

Appendix 1 – Nomenclature 72

Appendix 2 - MATLAB Code for Heat Exchanger Simulation 74

Description of Files 74

Appendix 3 – GMDH Matlab Code for Fault Detection 99

Vita.... 110

LIST OF TABLES

Table 2.1. Heat Transfer Coefficients 20

Table 2.2. Heat Exchanger Simulation Accuracy 21

Table 2.3. Sequoyah Condenser Parameters (TVA 1990) 35

Table 2.4. Flow Rates for Sequoyah Condenser 35

Table 3.1. Process Variables 38

Table 3.2. PID Parameters in the Control Loop 41

Table 3.3. Changes Necessary for a Large-Scale Heat Exchanger. 44

Table 3.4. Properites of Three Heat Exchangers 44

Table 4.1. Behavior of process variables and control functions for various fault conditions (Emerson Process Management data) 47**

Table 4.2. Extreme Values Caused by Faults* 49

Table 5.1. Fault Outcome Table for Simulations 58

Table 5.2. GMDH Prediction Variables 59

Table 5.3. GMDH Fault Identifiers 63

LIST OF FIGURES

Figure 1.1. Depiction of a Temperature to Flow Cascade Control Loop..... 2

Figure 1.2. Cutaway View of Typical Shell-and-Tube Heat Exchanger..... 6

Figure 1.3. Flow Paths for Heat Exchanger Categorization 7

Figure 2.1. A Typical Heat Exchanger Physical Process Model..... 12

Figure 2.2. Heat Exchanger Simulation Accuracy for Several Data Points 22

Figure 2.3. Typical Condenser – Horizontal Tube Configuration. 23

Figure 2.4. Laminar Film on Vertical Surface 27

Figure 2.5. Horizontal Tubes Bank Heat Transfer Coefficient vs. Number of Tubes Vapor Shear in Tube Banks..... 33

Figure 3.1. Heat Exchanger Control Loop. 39

Figure 3.2. Simulink Model of the Heat Exchanger Control Loop. 41

Figure 4.1. GMDH Layer Structure with k Layers (Ferreira 1999) 51

Figure 4.2. Arrangement of Input for GMDH Algorithm 52

Figure 5.1. Measured and actual temperatures of the hot fluid at the outlet T(h,out). 55

Figure 5.2. Measured Flow for Temperature Bias Fault at t=1,000 seconds. 56

Figure 5.3. GMDH Predictions (x) plotted against the model results (o) for 256 test cases for the Cold Flow Setpoint Prediction..... 60

Figure 5.4. The percentage error from the GMDH prediction to the model results for the Cold Flow Setpoint Prediction. The error does not exceed 2.5%..... 61

CHAPTER 1.

INTRODUCTION

The general temperature to flow cascade (TFC) involves two fluid streams passing through a heat exchanger. A proportional-integral-differential (PID) controller is used to maintain the outlet temperature on one side of the heat exchanger by modulating the flow on the other side of the exchanger. The specific system considered herein has a hot water stream delivered by a centrifugal pump through a control valve to the tube side of a conventional shell and tube heat exchanger. The PID controller monitors the outlet temperature of the cold water flow on the shell side of the exchanger and modifies the cold side flow through the control valve to maintain the hot side outlet temperature near to the prescribed set-point.

The Temperature-to-Flow-Cascade (TFC) control loop is common to many industrial processes. The proper function of these systems is often important to product quality or energy efficiency in chemical process plants and refineries. The fault detection and isolation (FDI) rule base developed herein facilitates timely and accurate identification of system faults, and is very valuable to those who operate and maintain these facilities. Figure 1.1 depicts a TFC control loop.

The pump, heater, valves, heat exchanger, piping, controllers, and instrumentation are modeled using Matlab and Simulink. The performance of the model is validated

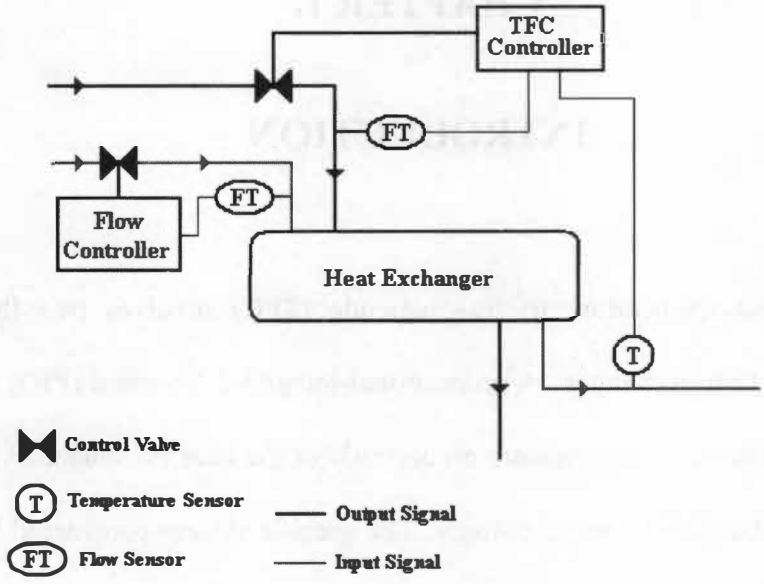

Figure 1.1. Depiction of a Temperature to Flow Cascade Control Loop

using steady-state data from an experimental system. The faults imposed on the physical system and the model include sensor biases, fluid leaks, unresponsive valves, plugged process lines, and fouling. The steady state change in process variables before and after the fault is used to generate tables that indicate process trends caused by the faults. The set of trend identifiers form a 'fault ID' that can categorize faults as they occur. Results for the physical system and model are presented. The simulation results indicate that unique identifiers exist for most faults. Two simulated faults, tube fouling and plugged tubes, share the same identifiers.

The mechanistic model is used to create data to develop a data driven model using the Group Method of Data Handling (GMDH) method. The GMDH method is often used in FDI system development when good mechanistic models are not available. The

GMDH model matches the mechanistic model well and the resulting FDI system remains useful.

1.1 Problem Statement and Objectives

Manufacturing and industrial plants of all types have heat exchangers installed. These heat exchangers are often critical to continued operation, and failure of heat exchangers could mean costly outages. It is of interest to detect and diagnose incipient faults in heat exchangers by analyzing data already available from the common instrumentation in any heat exchanger installed in a control loop. Use of the proposed FDI system requires only a data-acquisition system and a small software package. It is important to note that the system operates with data from existing sensors; these data are processed by the FDI system to diagnose faults or degradation. Monitoring the condition of common equipment with existing sensors provides an economical way to detect faults or degradation as they happen.

The objective is to take the various process signals from an instrumented control loop and use the behavior of these signals over time to determine potential faults. Examples of input signals include inlet and outlet temperatures, pressure, flow rate, and valve position. The time domain behavior of these signals is trended, and the trends allow faults to be identified.

A heat exchanger (HX) in a temperature to flow cascade control loop is modeled to facilitate simulation of a variety of faults. This model is validated using data from a laboratory control loop. The model is scalable to larger equipment size typical of industrial applications. A rule base to characterize normal operation and operation with

faults is developed. The heat exchanger and control system is also modeled using the Group Method of Data Handling (GMDH). The mechanistic, GMDH, and the physical system are used together to develop and validate the FDI system.

1.2 Heat Exchanger Review

Heat exchangers are fundamental elements of this thesis research and our technological society. They are present in almost any engineering system. The heat exchanger moves energy in the form of heat from one environment to another. A complete listing of heat exchangers uses would be nearly impossible, so the author only presents a few useful examples. Most homes have several heat exchangers in refrigerator or home heating and cooling systems. Vehicles have heat exchangers to remove waste heat. Modern computer systems utilize heat exchangers to prevent overheating of the circuitry. Power plants use heat exchangers to transfer energy from one fluid stream to another and to remove waste heat.

An excellent low-level introduction to heat exchangers is present in the *Introduction to Thermal Sciences* by Schmidt, et. al (1993). Two general subcategories for application exist; no phase change in the heat exchanger, or phase change in the heat exchanger. The other primary method of categorizing heat exchangers is by flow path. Another excellent resource on heat exchangers is the *Handbook of Heat Exchanger Design* edited by Geoffrey Hewett (1990). The comprehensive volume contains overviews of all significant issues for heat exchanger design.

1.2.1 Phase Change Categorization of Heat Exchangers

If no phase change is present, two fluids transmit heat through some sort of intermediary vessel. In most cases, as in this thesis, the fluids are both liquids, though gas-gas or liquid-gas heat exchangers are used for a variety of applications. The most common type in this category is the *shell and tube* configuration, shown in Figure 1.2.

In this configuration, one fluid flows through a set of tubes while the other flow outside the tubes through the shell. A series of baffles inside the shell direct the flows according to the required design. The shell and tube is the most popular configuration for liquid-to-liquid heat transfer. Other types of heat exchangers where no phase change occurs include *plate* or *compact* heat exchangers. These heat exchangers are popular for gas-gas or gas-liquid heat transfer, and utilize connected plates with fins to enhance the heat transfer. Heat exchangers of this type are present on most modern electrical components when over-heating is a concern.

The other subcategory includes any heat exchanger where a phase change occurs. Phase change allows very high heat transfer coefficients, maximizing energy transfer per unit area. The higher pressures and temperatures tend to make design more difficult. Again, the shell and tube design is a common configuration. If the heat exchanger produces vapor, it is known as a steam generator, reboiler, or evaporator. The vapor may be formed on the tube or shell side. If the heat exchanger produces liquid via the condensation process, it is known as a condenser.

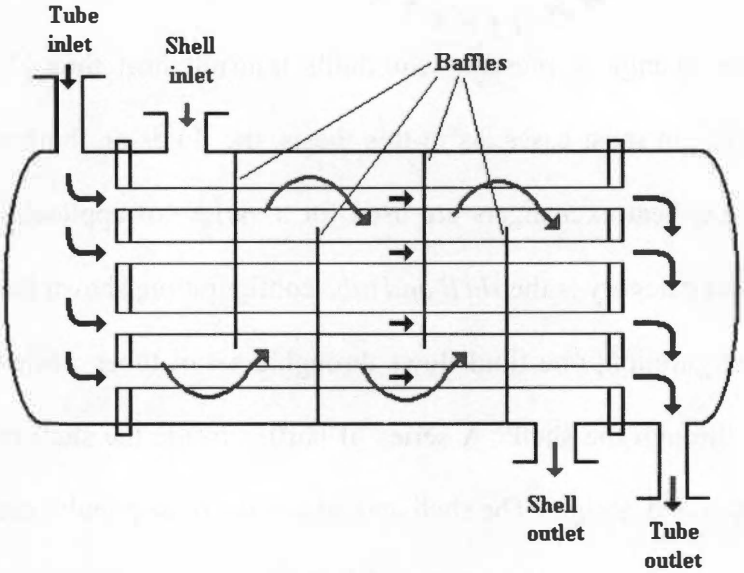

Figure 1.2. Cutaway View of Typical Shell-and-Tube Heat Exchanger.

1.2.2 Flow Configuration Categorization of Heat Exchangers

Heat exchangers are also identified by how fluid flows through the vessels. Five basic flow paths identify almost all heat exchangers. Figure 1.3 depicts all these flow patterns.

- Parallel flow: The fluid streams enter and exit on the same side and flow in the same direction.
- Counterflow: The fluid streams enter and exit on opposite sides, the flow direction is opposed.
- Crossflow: the flow is at right-angles to the other.

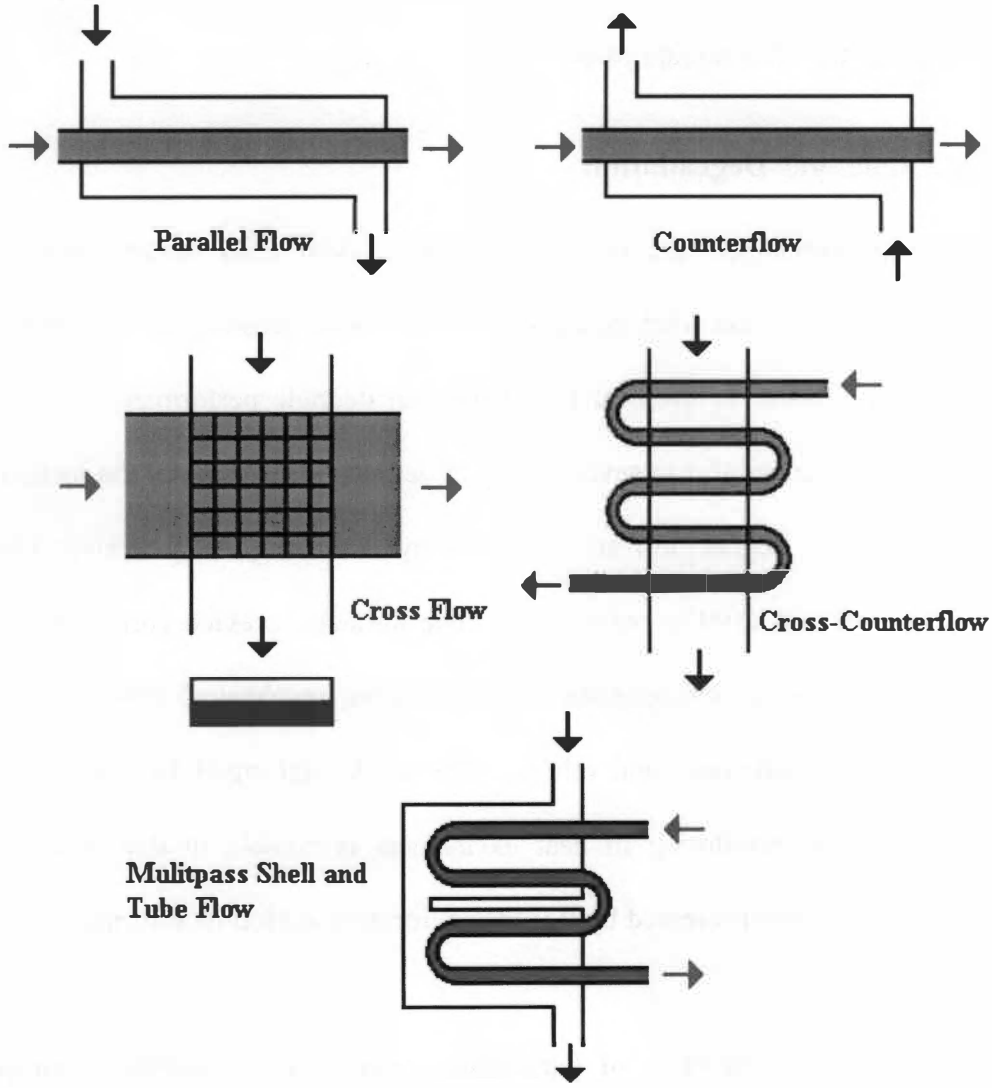

Figure 1.3. Flow Paths for Heat Exchanger Categorization

- Cross-counterflow: One fluid stream crosses the other at right angles several times.
- Multipass shell and tube flow: This method utilizes several sets of U-shaped tubes over which the other stream passes.

1.2.3 Heat Exchanger Degradation

Most heat exchangers are in constant contact with fluid streams at elevated temperatures, often in demanding conditions. Associated sensors, valves, and other component faults external to the heat exchanger can degrade performance. The heat exchanger itself is often subject to severe aging or degradation. Some of the factors that cause heat exchanger degradation are reviewed by Holman (1997), Walker (1982), Yokell (1990), and Bott (1997): corrosion of tube surfaces, crevice corrosion, pitting corrosion, stress corrosion, intergranular attack, erosion, mechanical fretting or wear, fouling, poor flow distribution, and others. The work performed by Huang (2003) regarding degradation monitoring in heat exchangers is notable in that it applies a method, similar to the one presented by this thesis, for degradation monitoring of fouling in steam generators.

Heat exchanger fouling is of particular concern. Low thermal conductivity deposits form on the heat transfer surface, impairing heat transfer. The fouling deposits also provide resistance to flow. It is not uncommon for the deposits to boost heat transfer initially due to the introduction of turbulence to the boundary layer, but over time fouling lowers the effectiveness of the heat exchanger. Fouling can result from particulate deposits from suspended solids in solution, crystallization, material freezing (phase

change), chemical reactions, or biological activity on the heat transfer surfaces. The reader is referred to Bott (1997) for a detailed review of fouling in heat exchangers.

1.3 Contributions of Thesis Research

Data-based models for fault detection and isolation have recently been applied in other systems, such as level control in a fluid column (Ferreira 1999). This is the first attempt to use data-based model for FDI of heat exchanger control loops. The nonlinear heat transfer models required to simulate a typical heat exchanger make modeling of the control loop challenging. The work presented in this thesis is a collaboration with Emerson Process Management, which provided the funding and experimental data from a heat exchanger control loop. The author combined these data with engineering heat transfer models to develop a simulation of a heat exchanger control loop complete with valves, sensors, and miscellaneous components. This simulation is then used to generate data for a heat exchanger control loop when operating properly, and with various faults. Then the trends resulting from various faults are analyzed to create a rule base describing 'symptoms' for the faults. To supplement this mechanistic model, another data-based modeling method, the Group Method of Data Handling (GMDH), was used to simulate the system behavior so that a rule base could be developed to detect and isolate faults on-line. The GMDH model is trained using data from the mechanistic model. The GMDH is a desirable modeling technique for applications where a mechanistic model is not easily developed.

1.4 Organization of the Thesis

This thesis is divided into six chapters. The first chapter is the introduction, which includes objectives, contributions, and organization. The second chapter includes background material on heat transfer coefficients and heat transfer in water-to-water heat exchangers. Some models for condensers are also developed and validated, but these models are not employed in a FDI system development. Computer modeling using these heat transfer methods is briefly described. Chapter 3 discusses the integration of a heat exchanger model into a typical heat exchanger control loop. Chapter 4 is the proposed Fault Detection and Isolation (FDI) system, where the behavior of the process variables helps diagnose and identify faults. Chapter 5 discusses and analyses the results of both the heat exchanger simulation and the FDI system. Concluding remarks and recommendations for future work are described in Chapter 6.

CHAPTER 2.

HEAT TRANSFER IN HEAT EXCHANGERS

After initial research into heat exchangers and a review of popular methods of heat exchanger modeling, two methods were selected for implementation: The Number of Transfer Units (NTU) method and the Log Mean Temperature Difference (LMTD) method. These schemes allow the heat flow q (watts) between two fluid streams in a heat exchanger, to be calculated. Factors used in the calculations include heat exchanger geometry, in conjunction with fluid mass flows and thermo-physical properties. A typical heat exchanger flow configuration is shown in Figure 2.1.

2.1 Heat Transfer Coefficients

In order to arrive at an understanding of the energy balance in a heat exchanger, we start with the general Reynold's transport equation for an energy balance in a simple pipe (the tube side),

$$\begin{aligned}&\iint_V \frac{d}{dt}\left(\rho(u+\frac{v^2}{2})\right)dV + \iint_A \left(\rho(u+\frac{v^2}{2})\right)\vec{v}\cdot\hat{n}dS = \\ &\iiint_V \rho\left(\frac{q'''}{\rho}+\vec{g}\cdot\vec{v}\right)dV + \iint_A \left(-q''+\left(\bar{\bar{\tau}}-P\bar{\bar{I}}\right)\cdot\vec{v}\right)\cdot\hat{n}dS\end{aligned} \tag{2.1}$$

The shear work is the portion of the pump energy converted to thermal energy by the heat exchanger and is negligible for water-water heat exchangers of the type considered herein. Potential energy and gravity terms cancel each other out, and no volumetric

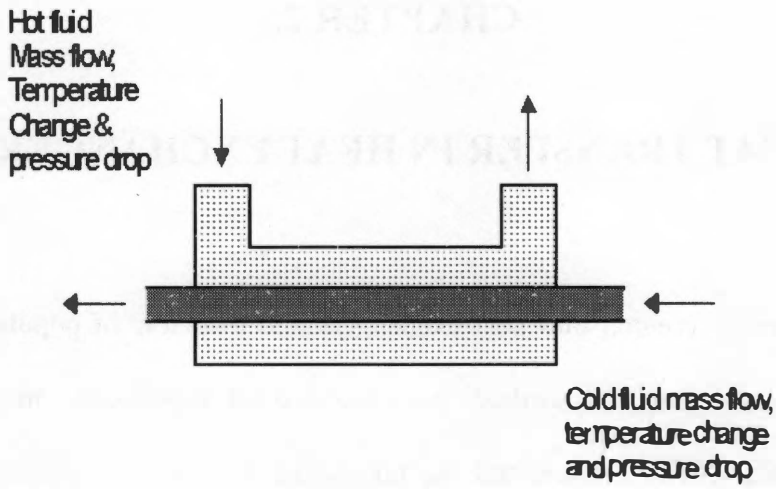

Figure 2.1. A Typical Heat Exchanger Physical Process Model.

generation is present. Note also that gravity and Pv terms balance for steady flow. The cross-sectional areas for the inlet and exit (A_{xs}) are equal.

$$-\iint_{A_{xs}in}\left(u+\frac{v^2}{2}\right)\rho v dS+\iint_{A_{xs}out}\left(u+\frac{v^2}{2}\right)\rho v dS=\iint_{A_{HT}} q'' dS+\iint_{A_{xs}in} Pv dS-\iint_{A_{xs}out} Pv dS) \tag{2.2}$$

Now the equation must be rearranged for integration of common terms. Constant density is assumed.

$$\rho\left\{-\iint_{A_{xs}}\left(u+\frac{v^2}{2}+\frac{P}{\rho}\right)v dS+\iint_{A_{xs}}\left(u+\frac{v^2}{2}+\frac{P}{\rho}\right)v dS\right\}=q'' A_{HT} \tag{2.3}$$

The enthalpy, h, is substituted into the equation by using the following relations,

$$h=u+\frac{P}{\rho}$$
$$h=C_P T \tag{2.4}$$

Neglect velocity terms by assuming,

$$\left(v^2/2\right)_{out} - \left(v^2/2\right)_{in} << \left(h_{out} - h_{in}\right) \tag{2.5}$$

The simplified equation is,

$$q'' A_{HT} = \rho v A_{xs} \left\{ \left(h + v^2/2\right)_{out} - \left(h + v^2/2\right)_{in} \right\} \tag{2.6}$$

Substituting the relation for mass flow,

$$\dot{m} = \rho v A_{xs} \tag{2.7}$$

We get,

$$\left(\dot{m} C_P\right)_{tube} \left(T_{out} - T_{in}\right)_{tube} \cong \left(q'' A_{HT}\right)_{tube} \tag{2.8}$$

The shell side development is identical and with same assumptions, to arrive at

$$\left(\dot{m} C_P\right)_{shell} \left(t_{out} - t_{in}\right)_{shell} \cong \left(q'' A_{HT}\right)_{shell} \tag{2.9}$$

Conservation of energy dictates the $q'' A_{HT}$ must be equal, so

$$\left(\dot{m} C_P\right)_{shell} \left(t_{out} - t_{in}\right)_{shell} = \left(\dot{m} C_P\right)_{tube} \left(T_{out} - T_{in}\right)_{tube} \tag{2.11}$$

This is the first order energy balance for a simple counter-flow heat exchanger. We can develop from this the following,

$$T_{out} = T_{in} - \frac{q}{(\dot{m} c_p)_{tube}} \tag{2.12}$$

$$t_{out} = t_{in} + \frac{q}{(\dot{m} c_p)_{shell}} \tag{2.13}$$

In words, the heat flow (q) is equal to the mass flow rate ($\dot{m}$) multiplied by the heat capacity (c_p) and the change in temperature in each leg. An uppercase T denotes the tube side, and a lowercase t denotes the shell side. This assumes heat losses from the heat

exchanger exterior are negligible and the heat capacities are constant across the heat exchanger.

The LMTD method evaluates a heat flow q using the following strategy:

$$q = U \cdot A \cdot F \cdot \Delta T_m$$
$$\Delta T_m = \frac{(T_{out} - t_{in}) - (T_{in} - t_{out})}{\ln(\frac{T_{out} - t_{out}}{T_{in} - t_{in}})} \tag{2.14}$$

U Overall heat transfer coefficient

A Heat transfer area

F Geometry Correction Factor, F = 1 for single-shell, single-tube heat exchanger

ΔT_m Log Mean Temperature Difference.

The quantity ΔTm results from a weighted average temperature difference that comes as a result of an energy balance for either co-current or counter current flow. The LMTD method was coded into MATLAB-SIMULINK. The NTU method is semi-empirical in nature, and uses the following relationships to determine q:

$$q = \varepsilon \cdot (\dot{m}c_p)_{\min} \cdot \Delta T_{in}$$
$$\Delta T_{in} = T_{in} - t_{in}$$
$$\varepsilon = f(NTU, (\dot{m}c_p)_{\min} / (\dot{m}c_p)_{\max}, geometry) \tag{2.15}$$
$$NTU = \frac{U * A}{(\dot{m}c_p)_{\min}}$$

Most of the symbols are as in the LMTD method, with the exception of ε, the effectiveness of a heat exchanger. Effectiveness is a function of U, A, the heat flow

capacities, and the geometry. The formula for effectiveness in a tube and shell heat exchanger is given by Todreas and Kazimi (1993):

$$\varepsilon = 2\left[1 + C_r + (1 + C_r^2)^{1/2} \frac{1 + e^{-NTU(1+C_r^2)^{1/2}}}{1 - e^{-NTU(1+C_r^2)^{1/2}}}\right]^{-1}$$

$$C_r = \frac{(\dot{m}c_p)_{\min}}{(\dot{m}c_p)_{\max}} \tag{2.16}$$

To determine NTU, the overall heat transfer coefficient, U, is required to complete the heat exchanger model, and includes terms that represent the heat transfer between the shell side fluid and the exterior of the tubes, h_o, and the heat transfer between the tube side fluid and the interior of the tubes, h_i. Fouling factors are also resident in the overall heat transfer coefficient model. The interior and exterior surfaces may both exhibit fouling represented by factors F_i and F_o, respectively. These factors may be complex functions of operating history, coolant quality and heat exchanger materials.

$$U = \frac{1}{\left[\frac{1}{h_i} + \frac{A_i \ln(r_o / r_i)}{2\pi kL} + \frac{A_i}{A_o h_o} + F_i + \frac{A_i}{A_o} F_o\right]} \tag{2.17}$$

Subscript *i* is for internal to the tube, and o is for external to the tube. The other variables in the expression for the overall heat transfer coefficient are:

A heat transfer area

r radius

k conductivity of the tube wall material.

L length of tube.

The bulk of the computation work is done in the calculation of the heat transfer coefficients, which is covered in the next section. It is important to note the inverse relationship of the heat transfer coefficients. If h_o and h_i differ significantly, the smaller value will dominate the overall heat transfer.

2.2 Calculation of Heat Transfer Coefficients

The Nusselt number, Nu, is defined as follows:

$$Nu_x = \frac{h_x x}{k} \tag{2.18}$$

where x is the characteristic dimension. Relations that determine the Nusselt number are known for various geometries and flow conditions. For the tube side of our heat exchanger, the geometry is flow in tubes and the characteristic dimension is the tube diameter. The equation can be solved for h:

$$h_x = \frac{Nu_x k}{x} \tag{2.19}$$

To validate this work, Emerson Process Management provided the geometry of a small heat exchanger, and additional parameters were obtained from the vendor documentation.

The Reynolds numbers for the flow were calculated for various flow conditions to determine the flow regime. The provided data indicates the experimental heat exchanger operates in laminar, transition and turbulent flow regimes.

Holman (1997) provides a diverse list for many correlations for the Nusselt number. The most accurate equation is given by Sieder and Tate (1936) as:

$$Nu_d = 1.86(\mathrm{Re}_d \mathrm{Pr})^{1/3}(\frac{d}{L})^{1/3}(\frac{\mu}{\mu_w})^{0.14} \tag{2.20}$$

μ dynamic viscosity

Re Reynolds Number

Pr Prandtl number

d Tube diameter

L Tube length.

The subscript *w* refers to the properties evaluated at the wall temperature. All other properties are evaluated at the bulk temperature. The above equation is only applicable for *Re Pr (d/L)* > 10. This restriction is not satisfied (Sieder and Tate 1936) in low flow conditions (less than 0.02 kg/s) on the tube side. Another laminar model developed by Hausen (1943) is used in the code for low flow situations.

$$Nu_d = 3.66 + \frac{0.0668(d/L)\mathrm{Re}_d \mathrm{Pr}}{1 + 0.04[(d/L)\mathrm{Re}_d \mathrm{Pr}]^{2/3}} \tag{2.21}$$

For turbulent flow in tubes, a good general model for the heat transfer coefficient on the interior of the tubes is given by Petukhov (1970).

$$Nu_d = \frac{(f/8) * \mathrm{Re} * \mathrm{Pr}}{1.07 + 12.7(f/8)^{1/2}(\mathrm{Pr}^{2/3} - 1)} \left(\frac{\mu_b}{\mu_w}\right)^{0.25} \tag{2.22}$$

The subscript *b* refers to the property evaluated at the bulk temperature, and the subscript *w* refers to the property evaluated at the wall temperature. When the flow is turbulent, this relationship should be accurate to within 6 % for the conditions in the heat exchanger (Holman 1997). Calculation of the Nusselt number with these equations allows the tube side heat transfer coefficient *h* to be determined. See Table 2.1 for values of the tube side heat transfer coefficient.

The shell side heat transfer coefficient is more difficult to evaluate, as it involves tube cross flow. The flow is heavily dependent on the tube bank geometry, which is described primarily by pitch-to-diameter ratio, P/D. Diagrams of the tube layout indicate a staggered triangular tube array, allowing calculation of P/D. In brief, the P/D ratio is given by the following formula, taken from Todreas and Kazimi (1993),

$$\frac{P}{D} = \left(\frac{d_{shell}}{d_{tube}}\right)\sqrt{\frac{2\pi}{4N\sqrt{3}}} \tag{2.23}$$

N is the number of tubes. The heat transfer coefficient is calculated as above, but the characteristic dimension is the tube diameter *d*.

$$h_o = CF * \frac{Nu \cdot k}{d} \tag{2.24}$$

The correction factor CF provided by Zukauskus (1972) accounts for the tube banks in cross flow. Zukauskus also presents a correlation of the Nusselt number that accounts for a wide variety of Reynolds numbers and property variations.

$$\mathrm{Re} \leq 100 \qquad Nu = C \cdot \mathrm{Re}^{n} \cdot \mathrm{Pr}^{0.36} \cdot \left(\frac{\mathrm{Pr}}{\mathrm{Pr}_w}\right)^{0.25} \quad C = 0.683 \qquad n = 0.466$$

$$100 < \mathrm{Re} < 1000 \quad Nu = 0.35 + 0.56 \cdot \mathrm{Re}^{0.52} \cdot \mathrm{Pr}^{0.3} \tag{2.25}$$

$$\mathrm{Re} > 1000 \qquad Nu = C \cdot \mathrm{Re}^{n} \cdot \mathrm{Pr}^{0.36} \cdot \left(\frac{\mathrm{Pr}}{\mathrm{Pr}_w}\right)^{0.25} \quad C = 0.27 \qquad n = 0.6$$

This set of relationships provides an accurate relation for determining the shell side heat transfer coefficient. Table 2.1 shows shell side heat transfer coefficients along with the tube side heat transfer coefficients typical for the HX used in this study.

The shell side coefficients in this study tend to be greater than the tube side, but this is not always true. The heat transfer coefficients for either side are dependent upon geometry, flow velocity and fluid thermo-physical properties. As seen in the table, overlap exists when high tube side flow and low shell-side flow occur. Both heat transfer coefficient models are important for model accuracy.

Table 2.1. Heat Transfer Coefficients

Tube Side		Shell Side		
Flow Rate (kg/s)	h_i (W/m^2 °C)	h_o (W/m^2 °C)	Flow Rate (kg/s)	U (W/m^2 °C)
0.032	2582.8	4035.8	0.063	1480
0.038	2964.2	4546.2	0.082	1690
0.044	3333.1	5002.4	0.101	1880
0.050	3691.7	5419.7	0.120	2050
0.057	4040.7	5807.1	0.139	2220
0.063	4382.3	6170.5	0.158	2390
Shell flow 0.158 kg/s		Tube flow 0.05 kg/s		

2.3 Accuracy of Heat Exchanger Model

Data for an operating heat exchanger was provided by Emerson Process Management to validate the heat exchanger model prior to its use in a control loop simulation. Table 2.2 shows twenty representative points from a data set of 71 total points. These results are also plotted in Figure 2.2. Note the error bars, provided for the heat exchanger data, are simply to indicate how close the simulation values are to the real results. The error bars are 5% of the temperature value, with error given as:

$$\frac{T_{actual} - T_{sim}}{T_{actual}} \tag{2.26}$$

The maximum error between simulated and actual temperature for the entire data set is 8%. Most values are within 3% of the prediction. The simulated values appear to consistently under predict the data, this trend is not evident in all 71 available data points.

Table 2.2. Heat Exchanger Simulation Accuracy

T(h,out) (° F)	Simulated T(h,out) (°F)	T(h,out) % Error	T(c,out) (° F)	Simulated T(c,out) (°F)	T(c,out) % Error
104.84	99.7	4.88%	85.39	88.2	3.29%
102.84	97.0	5.73%	81.78	84.3	3.05%
101.21	94.9	6.21%	79.15	81.4	2.79%
99.91	93.3	6.57%	76.76	79.1	3.00%
98.86	92.0	6.89%	75.14	77.3	2.85%
97.91	90.7	7.36%	73.74	75.6	2.51%
97.26	89.7	7.78%	72.62	74.3	2.28%
96.52	88.7	8.06%	71.45	73.2	2.42%
111.46	108.3	2.79%	94.05	97.4	3.53%
108.31	104.7	3.35%	89.09	91.8	3.05%
106.09	102.0	3.82%	85.23	87.8	3.02%
104.33	99.8	4.36%	82.36	84.6	2.74%
102.35	97.7	4.56%	80.05	81.9	2.34%
101.59	96.4	5.08%	78.09	80.0	2.49%
100.67	95.1	5.55%	76.57	78.3	2.24%
99.65	94.0	5.71%	75.17	76.9	2.24%
98.84	93.1	5.82%	74.06	75.8	2.28%
115.08	112.2	2.55%	97.49	100.2	2.77%
111.8	109.1	2.44%	92.29	95.2	3.15%
109.38	106.5	2.65%	88.41	91.1	3.08%

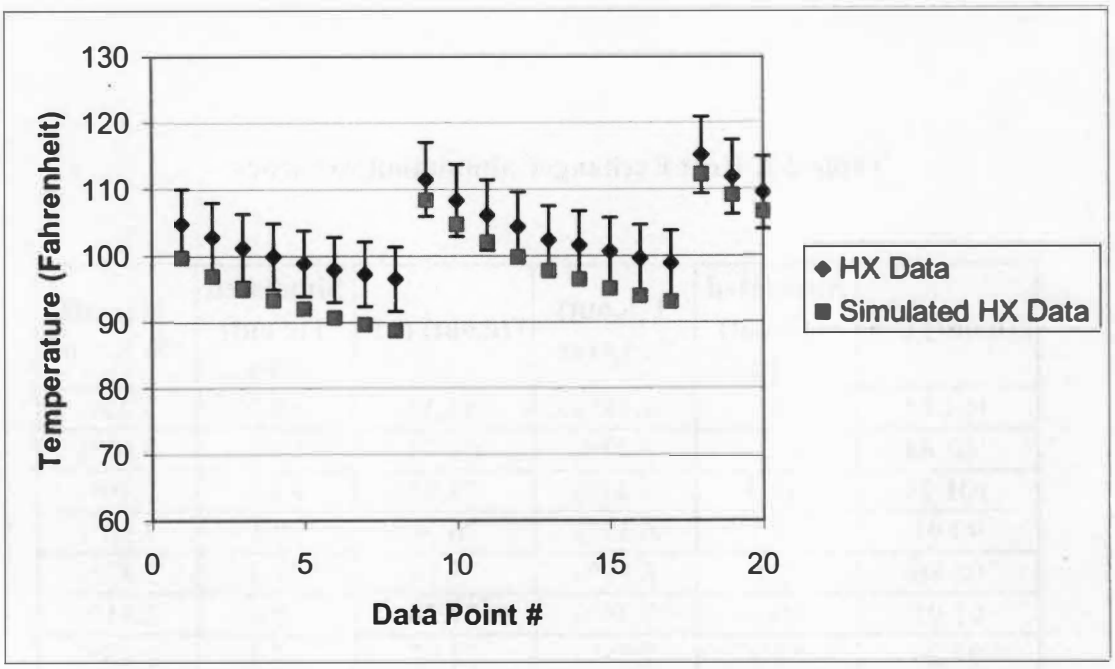

Figure 2.2. Heat Exchanger Simulation Accuracy for Several Data Points

2.4 Heat Transfer in Condensers

The phase change of vapor to liquid is a very efficient method of heat transfer. Condensers are a common type of heat exchanger that exploit condensation as a heat transfer method. Figure 2.3 depicts typical condenser operation. This thesis reviews available correlations to calculate the heat transfer coefficient for large-scale condensers. Condensation on the external surface of vertical and horizontal tubes is considered. Inclined tubes may be treated as a special case of horizontal tubes. The condensation models determine condensate flow rates using geometry information from a large-scale power plant condenser. The models predict the condensate flow within 10% of the given

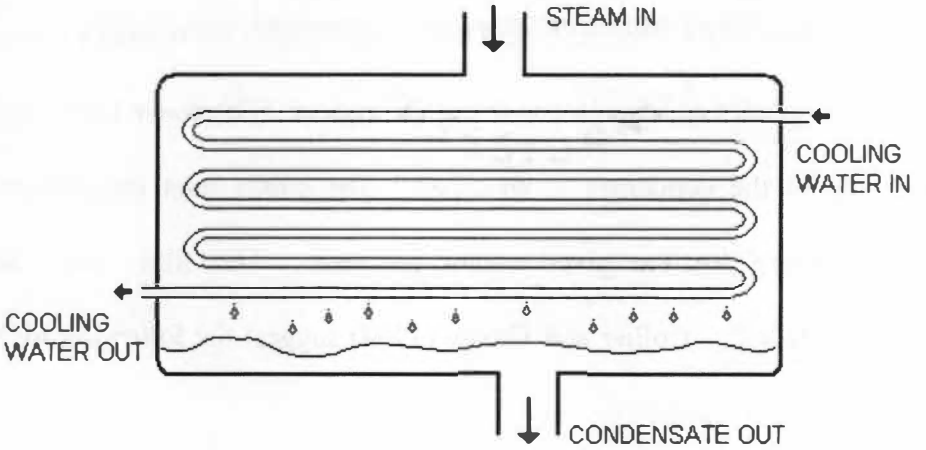

Figure 2.3. Typical Condenser – Horizontal Tube Configuration.

safety report values. One correlation gives a result within 1% of the given safety report condensate flow value.

The condenser heat transfer correlations are used in a computer code written with the MATLAB-Simulink software package. Condenser orientation (vertical or horizontal) and geometry information must be provided. Condensate flow rate is then calculated as a function of inlet conditions for both the steam and water flows. Thermo-physical properties are included in the code.

The condenser model may also be used in a heat exchanger control loop simulation in a manner similar to that employed for the water-water heat exchanger. However, the full implementation of this model into a rule based FDI is beyond the scope of this thesis.

2.4.1 Theory

Good estimates of the film, wall, and bulk temperatures are needed to evaluate the thermophysical properties of a given condensation system. The water inlet temperature on the tube side of the condenser is specified. The steam inlet temperature is the saturation temperature for the given steam pressure. The film, wall, and bulk temperatures are needed. Collier and Thome (1994) suggest the following for the film temperature,

$$T_{film} = T_w + F(T_{sat} - T_w) \tag{2.27}$$

The constant F is given as 0.25 or 0.31. Equation (2.28) gives an adaptation of the previous equation, now used for calculation of wall temperature, with F chosen as 0.25,

$$T_w = T_{in} + 0.25(T_{sat} - T_{in}) \tag{2.28}$$

The bulk temperature is taken to be the water inlet temperature. For additional accuracy the process may be repeated and a new bulk temperature calculated. The results indicate that only one iteration is needed.

All film properties are evaluated at the film temperature, with a few given exceptions. The vapor density is taken at saturation temperature. The evaporation enthalpy is evaluated at saturation temperature, but an adjustment may be made to

account for condensate subcooling. Collier and Thome (1994) suggest the following modification,

$$h'_{fg} = h_{fg} + 0.68C_P\left(T_{sat} - T_w\right) \tag{2.29}$$

The dynamic viscosity is calculated using a weighted average method, shown in Equation (2.30).

$$\mu_l = \frac{3\mu_w + \mu_{sat}}{4} \tag{2.30}$$

All thermophysical data are from Schmidt, et al (1984).

Many available correlations for condensation heat transfer are expressed in terms of the film Reynolds number. The flow regime will also have to be determined with the use of the Reynolds number. The value of Re used is typically that at the bottom of tube surface (L in a vertical arrangement and D in a horizontal arrangement). The definition of the Reynolds number for condensate film flow is (Butterworth 1990),

$$\mathrm{Re}_L = \frac{4\dot{M}_L}{\mu_l} \tag{2.31}$$

The symbol $\dot{M}$ is the mass flow rate per unit width. To calculate the true mass flow rate use the following relation:

$$\dot{m} = \dot{M}s \tag{2.32}$$

Where s is the width of the surface or the tube circumference in the case of condensation on tubes. The mass flow rate can be calculated using a first order energy balance as follows:

$$\dot{M}_L = \frac{\bar{h}(T_{sat} - T_w)L}{h_{fg}} \tag{2.33}$$

The h_{fg} term is modified using Equation (2.29) to account for film sub-cooling.

Three main flow regimes are considered for condensation heat transfer: laminar, wavy laminar, and turbulent (Butterworth 1990). The transition to wavy laminar is reached quickly, at a film Reynolds number of 30. The correlations for laminar flow may include the wavy results as a tuning constant or already have the wave effects integrated into the result. There is some disagreement as to the transition number for turbulent flow; Butterworth suggests a Reynolds number of 1600. For condensate flow on vertical tube surfaces, vapor shear effects with upward flow reduce the transition number.

2.4.2 Vertical Tubes

The original condensation heat transfer model developed by Nusselt (1916), adjusted for buoyancy is given by Butterworth (1990) is shown here. Figure 2.4 shows a laminar film on a vertical surface. First consider a force balance on a vertical film of thickness δ at position x,

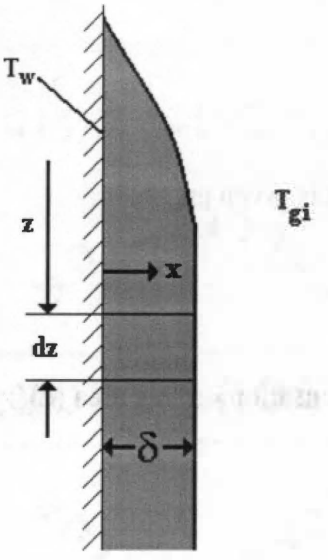

Figure 2.4. Laminar Film on Vertical Surface

$$(\delta - x)dz(\rho_f - \rho_g)g = \mu_l\left(\frac{d\mu_x}{dx}\right)dz \tag{2.34}$$

With the boundary condition $\mu_x = 0$ at $x = 0$, the equation may be integrated to determine μ_x,

$$\mu_x = \frac{(\rho_l - \rho_g)g}{\mu_l}\left[x\delta - \frac{x^2}{2}\right] \tag{2.35}$$

The mass flow per unit width, $\dot{M}$, can be found by integration of the velocity profile across the film, with constant liquid denisty,

$$\dot{M} = \rho_l\int_0^\delta \mu_x dx = \left[\frac{\rho_l(\rho_l - \rho_g)g\delta^3}{3\mu_l}\right] \tag{2.36}$$

As the thickness of the film builds from the top point, the rate of increase can be given as,

$$\frac{d\dot{M}}{d\delta} = \left[\frac{\rho_l(\rho_l - \rho_g)g\delta^2}{3\mu_l}\right] \tag{2.37}$$

The heat transfer via conduction is given as,

$$d\phi = \frac{k_l}{\delta}(T_{gi} - T_w)dz \tag{2.38}$$

Therefore the mass rate of condensation on this area (dΓ) is,

$$d\dot{M} = \frac{k_l}{h_{fg}\delta}(T_{gi} - T_w)dz \tag{2.39}$$

Substituting this expression Equation (2.39) into Equation (2.37), then separating variables and integrating with the boundary condition $\delta = 0$ at $z = 0$.

$$\mu_l k_l z(T_{gi} - T_w) = \rho_l(\rho_l - \rho_g)g\left(\frac{\delta^4}{4}\right)h_{fg} \tag{2.40}$$

The film thickness δ can be determined now as,

$$\delta = \left[\frac{4\mu_l k_l z(T_{gi} - T_w)}{gh_{fg}\rho_l(\rho_l - \rho_g)}\right]^{1/4} \tag{2.41}$$

The local heat transfer coefficient, *h(z)*, is shown as follows,

$$h(z) = \frac{k_f}{\delta} = \left[\frac{\rho_l(\rho_l - \rho_g)gh_{fg}k_f^3}{4\mu_l z(T_{gi} - T_w)}\right]^{1/4} \tag{2.42}$$

The Nusselt number is given by,

$$Nu(z) = \left[\frac{h(z)z}{k_f}\right] = \left[\frac{\rho_l(\rho_l - \rho_g)gh_{fg}z^3}{4\mu_l z(T_{gi} - T_w)}\right]^{1/4} \tag{2.43}$$

The local Nusselt number may be integrated over the surface to produce a mean value on the surface. The integration produces the following expression for overall heat transfer coefficient:

$$\bar{h} = 0.943\left[\frac{k_l \rho_l (\rho_l - \rho_g) g h_{fg}}{4\mu_l (T_{sat} - T_w) L}\right]^{1/4} \tag{2.44}$$

This model is valid only for laminar flows, Re < 30. The equation may be modified to be valid for Re < 1600 by replacing the constant at the front of the expression with 1.13. Equation (2.45) shows this result.

$$\bar{h} = 1.13\left[\frac{k_l^3 \rho_l (\rho_l - \rho_g) g h_{fg}}{4\mu_l (T_{sat} - T_w) L}\right]^{1/4} \tag{2.45}$$

Another condensation model appropriate for Re < 1600 is given as (Butterworth 1990):

$$\frac{\bar{h}}{k_l}\left[\frac{\mu_l^2}{\rho_l (\rho_l - \rho_g) g}\right]^{1/3} = \frac{\mathrm{Re}_L}{1.08\,\mathrm{Re}_L^{1.22} - 5.2} \tag{2.46}$$

Re_L can be determined using Equations (2.31) and (2.33) to obtain Equation (2.47).

$$\mathrm{Re} = \frac{4\bar{h}(T_{sat} - T_w) L}{\mu_f h_{fg}} \tag{2.47}$$

This result is substituted into Equation (2.46) to obtain Equation (2.48).

$$\bar{h} = \left(\frac{\dfrac{4(T_{sat} - T_w)L}{\mu_f h_{fg}} + \dfrac{5.2}{k_l}\left[\dfrac{\mu_l^2}{\rho_l(\rho_l - \rho_g)g}\right]^{1/3}}{\dfrac{1.08}{k_l}\left[\dfrac{\mu_l^2}{\rho_l(\rho_l - \rho_g)g}\right]^{1/3}\left(\dfrac{4(T_{sat} - T_w)L}{\mu_f h_{fg}}\right)^{1.22}} \right)^{\frac{1}{1.22}} \tag{2.48}$$

For treating turbulent Reynolds numbers (Re > 1600) Equation (2.49) is suggested (Butterworth 1990),

$$\bar{h} = \frac{\mathrm{Re}_L}{C_0\left[8750 + 58\,\mathrm{Pr}^{-0.8}\left(\mathrm{Re}_L^{0.75} - 253\right)\right]} \tag{2.49}$$

Now we have two correlations for laminar flow and one for turbulent flow on vertical tubes. Vertical tube banks may be treated as many individual vertical tubes.

2.4.3 Horizontal Tubes

The heat transfer coefficient for condensation on the outer surface of a single horizontal tube has a form similar to that for vertical tubes.

$$\bar{h}_{tube} = 0.728\left[\frac{k_l^3 \rho_l(\rho_l - \rho_g)g h_{fg}}{\mu_l(T_{sat} - T_w)D}\right]^{1/4} \tag{2.50}$$

The constant in front is different, accounting for tube curvature, and the characteristic length L is replaced with D, the tube diameter. The tube may be treated as inclined by replacing the gravitational constant g with g*sinθ, where θ is the angle of inclination from horizontal. This equation is valid for laminar flow. It is not likely that a single tube will experience turbulent flow with most condenser geometries, so this possibility is not considered for a single tube. Turbulence is possible in tube banks, as condensate drips from tube to tube.

To account for the effects of dripping condensate and turbulence in the film, one must consider a stack of horizontal tubes and determine an average heat transfer coefficient to cover all the tubes in a stack. Butterworth (1990) gives two suggestions for the average heat transfer coefficient given the single tube heat transfer coefficient.

$$\bar{h} = \bar{h}_{tube} N^{-1/4} \tag{2.51}$$

$$\bar{h} = \bar{h}_{tube} N^{-1/6} \tag{2.52}$$

Both multiply the single tube coefficient. Equation (2.53) is an empirical correction to the analytical result in Equation (2.49). For a sufficiently large number of tubes, N > 10, an analytical result can be obtained.

$$\bar{h} = \bar{h}_{tube} (1.24) N^{-1/4} \tag{2.53}$$

Figure 2.5 compares the three methods, plotting the heat transfer coefficient multiplier against the number of tubes.

Previously, vapor shear was not considered in the calculation of the heat transfer coefficients. Generally it is sufficient to omit the effects of vapor shear, but with a high vapor velocity, it must be considered. The following correction coefficient applies when the vapor flow is perpendicular to the condensate flow (Butterworth 1990),

$$\frac{h'}{\bar{h}} = 1.4\left[\frac{u_g^2(T_{sat} - T_w)k_l}{gDh_{fg}\mu_l}\right]^{0.05} \tag{2.54}$$

The above correlation should be used only when the heat transfer coefficient ratio is greater than 1. Also the model is not validated if the correction is greater than 1.7. Note this result is for a single tube. Application to a tube bank is similar, but some thought must go into choosing a proper vapor velocity. Butterworth (1990) recommends the velocity at the maximum cross sectional area.

2.4.4 Condenser Model Validation

Sequoyah Nuclear Power plant, operated by the Tennessee Valley Authority is located in Soddy-Daisy, Tennessee. Each of the two units produces 1170 MW of electricity, and operates at a core power of 3400 MW thermal (TVA 1990). Excess thermal energy must be removed from the plant, and Sequoyah has a suitably sized condenser to move this heat to the cooling towers and the environment.

The shell side of the condenser operates at a pressure of one pound per square inch (actual). Using the given temperature change and circulating water flow rate, a rough estimate for condensate production is 930 kg/s. This compares to the heat balance

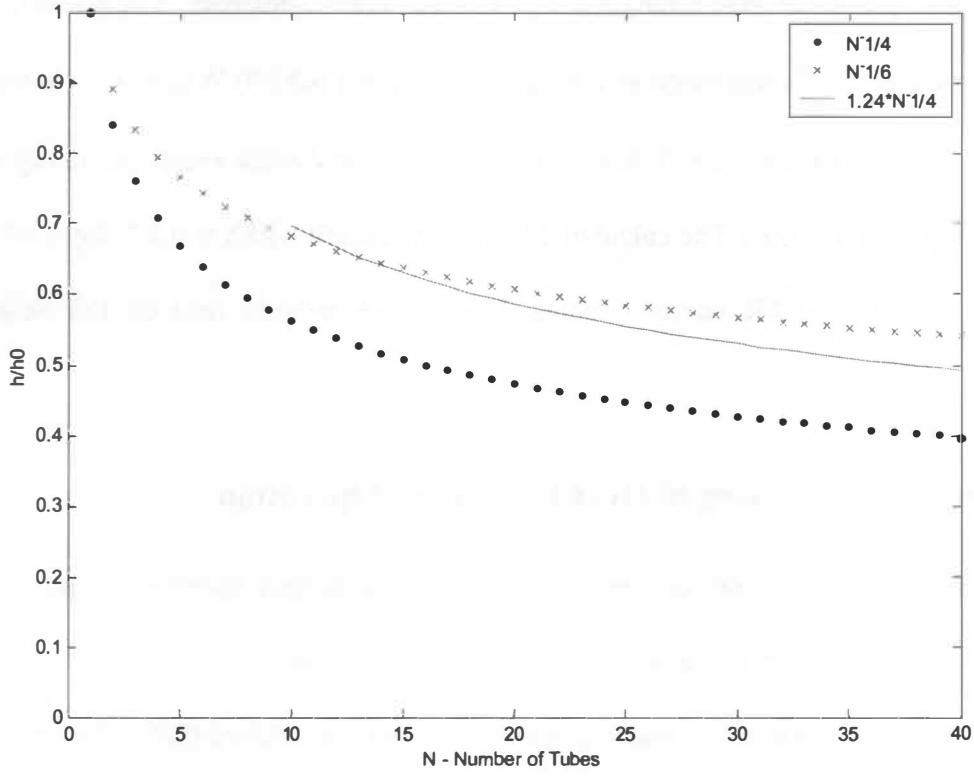

Figure 2.5. Horizontal Tubes Bank Heat Transfer Coefficient vs. Number of Tubes

Vapor Shear in Tube Banks

diagram in the Final Safety Analysis Report (FSAR) of 1084 kg/s in a typical operation mode (TVA 1990). This latter value will be used to compare to the heat transfer correlations. The horizontal tube bank values are given as a comparison. If all the tube were arrayed in a square bank, the number of tubes in a stack would be 242. This approximates the actual tube arrangement in a power plant condenser. For a single tubes, the heat transfer coefficient calculated by Equation (2.26) is 8240 W/(m^2°C). In the tube stack, this value becomes 2590 W/(m^2°C) using the analytical result for many tubes, given in Equation (2.53). The calculated flow rate per unit width is 0.276 kg/s, which is within 5% of the FSAR value. Tables 2.3 and 2.4 provide data on the Sequoyah condenser.

2.5 Computer Modeling of Heat Exchanger Operation

The MATLAB code for simulating the heat exchanger operation is provided in Appendix 2. A summary of the code operation is given below.

The driver model takes four parameters as input: the temperature and flow rate for both the hot and cold leg. The input must be in metric units, °C and kg/s. (NOTE: The Simulink model has input adapters that can convert English units to metric prior to running the code). The heat exchanger geometry is built into the code along with a property table for water over the appropriate temperature ranges. The thermal conductivity for steel (stainless 316) is also included. The heat exchanger is made of stainless steel 316L.

Table 2.3. Sequoyah Condenser Parameters (TVA 1990)

Parameter	Given Measurement	Conversion
Tube Length	49'10"	15.2 m
Number of Tubes	58,860	
Tube OD	7/8"	2.223 cm
Wall Thickness	20 BWR	0.00888 cm
Circulating Water Flow Rate	522000 GPM	32900 kg/s
Circulating Water Temperature Change	30 °F	17 °C

Table 2.4. Flow Rates for Sequoyah Condenser

Flow Type	Flow Rate
Total Flow	1084 kg/s
Flow per Tube	0.0184 kg/s
Flow per Tube per unit width	0.264 kg/s
Model Estimate	0.276 kg/s

Bulk temperatures are estimated using the NTU method, and an assumed effectiveness of 0.5. This estimation is the first step in an iteration that produces bulk temperatures within the accuracy of the thermo-physical properties table for the expected operation range of the heat exchanger. The calculations are repeated twice to obtain accurate bulk temperatures. The first step simply assumes an effectiveness of 0.5 to calculate estimated bulk temperatures. The calculation is carried forward with these estimates and produces the estimated outlet temperatures. This information is used to generate new bulk temperatures, which are accurate within the tolerance of the provided thermo-physical data. The effectiveness is updated twice within each time step. The number of iteration steps could be increased to support fluids such as hydrocarbons whose thermo-physical properties change rapidly over certain temperature ranges.

The bulk temperatures, along with the input, are passed to two sub-functions that calculate the tube and shell heat transfer coefficients. There are provisions in the code for fouling factor simulation and tuning constants to fit the model to the data.

The overall heat transfer coefficient, U, is then calculated. The NTU model is then used in full to calculate the heat transfer in watts. An energy balance is used to determine the resulting steady state temperatures for the tube and shell sides.

CHAPTER 3.

HEAT EXCHANGER CONTROL LOOP SIMULATION

The heat exchanger control loop is based on a test facility built by Emerson Process Management, Performance Technologies division. The software used for simulation is the SIMULINK package. The heat transfer calculation used is coded in MATLAB, and is integral to the simulated loop. Table 3.1 documents the list of basic process variables for the simulation.

3.1 Heat Exchanger Control Loop Configuration

The heat exchanger control loop and simulation are based on a common configuration used at many industrial locations. The Temperature-to-Flow Cascade (TFC) control loop uses a PID control to determine the cold water flow rate such that the hot outlet temperature is near the set point value. Figure 3.1 illustrates the control loop and associated flow paths.

Note that this diagram does not include the TFC PID controller or the flow PID controllers. The simulation model emphasizes the controllers. The heat exchanger, heater, pump, and water supply are all assumed to function correctly and supply the demanded flow at steady temperatures. Additionally, the manual bypass valve on the hot side is not incorporated into the simulation. The following section details components incorporated into the model.

Table 3.1. Process Variables

Variable	Description
Time	-
Temperature	Hot and cold legs, inlets and outlet, measured and actual
Flow	Hot and cold legs, measured and actual
Pressure	Pressure drop across the heat exchanger, hot and cold side
Controls	Valve position, controller demanded position and flow, PID output

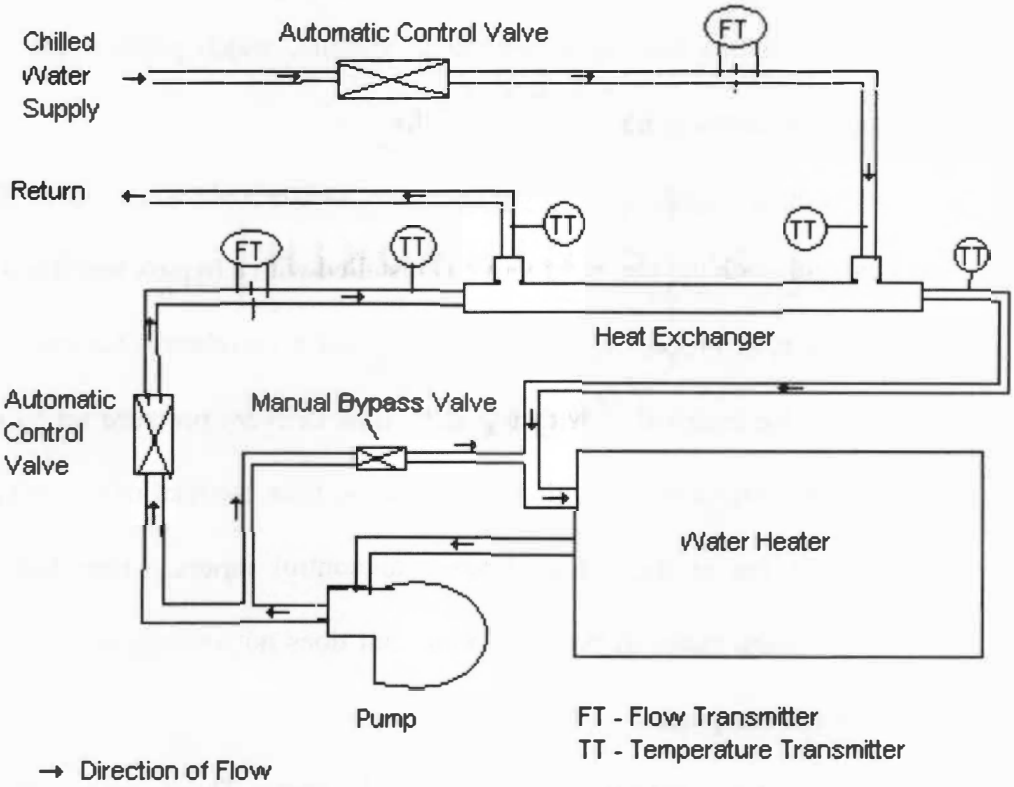

Figure 3.1. Heat Exchanger Control Loop.

3.2 Control Loop Components

The primary component of the control loop is the heat exchanger. Its response is simulated using the steady-state NTU method. The simulation provides the flow rates and inlet temperatures to the heat exchanger code module, which predicts the outlet temperatures. The other components include the following:

Automatic Control Valves for both the hot water and the cold water sides. These valves are identical, although the hot water valve is installed with a bypass line fitted with a manual valve. Emerson Process Management provided a correlation between valve position and flow rate for each valve, with the valve/flow delivery pressure set by other system attributes. A transfer function with a step response time constant of 0.1 second is used to place a slight lag in the valve response to control inputs. This facilitates movement between system states in the simulation, but does not constitute an accurate rendering of the transient response.

Valve Controllers are present for each control valve. These components are placed with the valves and include set point inputs and a proportional-integral-derivative (PID) controller. Table 3.2 lists the PID parameters used for the valves. Temperature-to-flow cascade controller modifies cold water flow to match the hot temperature outlet set point. The PID parameters used match those in the Emerson test loop.

Sensors are included in the simulation, including the hot stream outlet temperature and the flow rate of the hot and cold water streams. The simulation includes the ability to bias the reported measurements of the sensors. Figure 3.2 shows the detailed Simulink model with the output modules deleted.

Table 3.2. PID Parameters in the Control Loop

Device	Valve Position Controller (% VP/GPM)	Temperature Cascade Controller (°F/GPM)
Proportional	1.5	0.3
Integral (Reset)	10	42
Derivative (Rate)	0.3	12

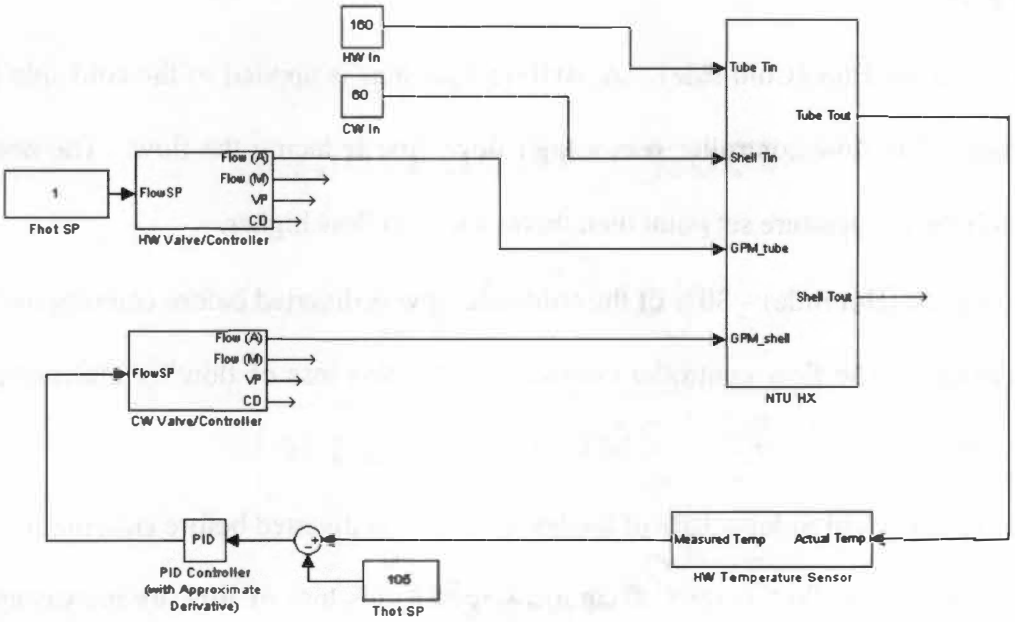

Figure 3.2. Simulink Model of the Heat Exchanger Control Loop.

3.3 Faults Induced in the Heat Exchanger Control Loop

The control loop model was developed to support the study of eight device faults:

- Temperature Sensor Bias (Hot Outlet) – A +3 degree Fahrenheit bias is applied to the hot outlet temperature sensor. This impacts the Cascade Controller directly, causing a perceived high temperature, which increases the cold-side flow.
- Flow Sensor Bias (Hot side) – For the hot side flow sensor, a +0.0095 kg/s bias is applied. The flow controller reads this as high, then decreases the hot flow to the heat exchanger, which also results in the cold flow being reduced to match the temperature set point.
- Flow Sensor Bias (Cold side) – A +0.0315 kg/s bias is applied to the cold side flow sensor. The flow controller sees a high flow, first reducing the flow. The need to match the temperature set point then drives the cold flow higher.
- Fluid Leak (Hot Side) – 30% of the cold-side flow is diverted before entering the heat exchanger. The flow controller compensates for this loss of flow by increasing the set point.
- Fluid Leak (Cold Side) – 30% of the hot-side flow is diverted before entering the heat exchanger. The flow controller compensates for this loss of flow by increasing the set-point.
- Stuck/Unresponsive Valve (Hot and Cold Side) – The valve is locked at a position near that (rounded to the nearest integer, % open) of the pre-fault position. For example, a valve at 24.5% open would lock to 25% open and would not respond to further controller demands.

- Plugged Process Lines – One or more of the heat exchanger tubes could become plugged during operation, decreasing the available heat transfer area. The simulated fault is 3 of the 37 tubes having completely blocked flow. Note this is about a 10% decrease in heat transfer area, but velocity is increased in the remaining tubes.
- Heat Exchanger Fouling – Fouling deposits could build up on the heat transfer surfaces on the tubes or on the shell side, thus decreasing the overall heat transfer coefficient. The simulation uses a fouling factor of 0.0001 m^2 °C/W for the shell-side fluid. This fouling factor represents the standard recommended design basis for fouling in feed-water under 50 °C (Bott 1997). This fouling factor will change the heat transfer coefficient U by 10%-20%.

3.4 Industrial Scaling for Control Loop

The heat exchanger control loop model is easily modified to simulate larger equipment. The appropriate turbulent flow correlations are included in the heat exchanger model. To update the model for larger heat exchanger configurations, the following changes are necessary to individual model components and code. Table 3.3 shows the necessary changes required to accomplish this. Two configurations of large heat exchanger are being modeled, with the properties listed in the following table. Table 3.4 shows the properties of the two large heat exchangers modeled. Heat exchanger #3 is the small heat exchanger from the Emerson test loop, and is provided for comparison. The heat exchanger parameters have been selected to represent typical large industrial heat exchangers. All listed heat exchangers are counter-current shell and tube designs. The industrial scale heat exchangers under fault behave similar to a small heat exchanger

Table 3.3. Changes Necessary for a Large-Scale Heat Exchanger.

Component	Changes
Thermo-physical Properties	None, the system is still water-water.
NTU Method	Effectiveness equation changed to two-tube pass model. (Also applies for any even number of tube passes.)
Tube Heat Transfer	None
Shell Heat Transfer	Removed correction factors for crossflow in tube banks with small numbers of rows.
Flow/Valve Model	Valve positions versus flow changed for high flow conditions.

Table 3.4. Properites of Three Heat Exchangers

	Heat Exchanger 1	**Heat Exchanger 2**	**Heat Exchanger 3**
Shell Outer Diameter	0.35 m	1 m	2.54 cm (1")
Tube Diameter	1 cm	1.9 cm (3/4")	0.24 cm
Number of Tubes	299	800	37
Tube Length	4 m	4 m	0.406 m
Heat Transfer Area	41.3 m^2	191.5 m^2	0.11 m^2
Typical Flow Rate – Tube	252 kg/s	631 kg/s	0.063 kg/s
Typical Flow Rate - Shell	505 kg/s	2524 kg/s	0.126 kg/s
Overall Heat Transfer Coefficeint U*	9000 $W/m^2{}^\circ C$	5000 $W/m^2{}^\circ C$	2500 $W/m^2{}^\circ C$

**U calculated for typical flows, 71° C (160°F) tube inlet, 16 ° C (60°F) shell inlet.*

under fault. The magnitude of the fault response may vary, but the trend of the response is similar. Thus the fault responses describes in Section 5 leading into the fault identification are also valid for large-scale heat exchangers.

CHAPTER 4.

PROPOSED FAULT DETECTION AND ISOLATION (FDI) METHOD

The previously proposed FDI system functions by using empirical and analytical relationships to verify sensor outputs in the control loop. These relationships, using a physics model as well as a data-driven modeling method called Group Method of Data Handling (GMDH), predict the state of the system given the time-averaged input from various sensors.

4.1 System Behavior During Faults

The current analysis is appropriate for the detection and isolation of faults during steady-state operation. The steady state limitation allows for a less complex model and provides greater flexibility for data collection with time averaging of data. Transient phenomena are currently not included in the analysis.

The FDI system receives process variables from sensors and then calculates other process variables, using the system model. Data should be time-averaged over an appropriate period based on thermal time constants and the data acquisition rate for the system. Table 4.1 shows the behavior of various process variables and was generated

Table 4.1. Behavior of process variables and control functions for various fault conditions (Emerson Process Management data)**

Fault ID	Fault	T(c,out) (°C)	SP(F,c) (kg/s)	Flow(c) (kg/s)	VP(c) (%)	CD(c) (%)	SP(F,h) (kg/s)	Flow(h) (kg/s)	VP(h) (%)	CD(h) (%)	DelP(h) (psi)	DelP(c) (psi)
1	T(h,out) Bias	↓	↑	↑	↑	↑	-	-	-	-	-	↑
2	Flow(h) Bias	↑	↓	↓	↓	↓	-	-	↓	↓	↓	↓
3	Flow(c) Bias	-	↑	↑	↓	↓	-	-	-	-	-	-
4	HW Leak	↑	↓	↓	↓	↓	-	-	↓	↓	↓	↓
5*	CW Leak 1	↑	↑	↑	↑	↑	-	-	-	-	-	-
5	CW Leak 2	-	↑	↑	↑	↑	-	-	-	-	-	-
6	CW Valve	-	↓	↑	↑	↓	-	-	-	-	-	↑
6	HW Valve	-	↑	↑	↑	↑	-	-	-	↑	-	↑

*CW Leak 1 results in cascade control saturation

**After faults occur, an increase in the listed process variable is recorded as an up arrow. A decrease is recorded as a down arrow. Changes of less than 4% are ignored. See section 4.2 for fault responses for saturated controller response.

from data provided by Emerson Process Management using the physical heat exchanger loop. The simulation results for the various faults are given in Section 5.

4.2 Extreme Values During Faults

Faults can cause the controllers to reach minimum or maximum possible values (saturate). The typical cause is an unattainable set point condition. The simulations indicate any fault can induce controller saturation if the fault magnitude is sufficiently large. The set of extreme fault condition values is given in Table 4.2.

Valve sticking will result in different trends depending on the controller correction for flow required to reach the T(h) set point. Cold water valve sticking can be easily determined if the T(h) is not at the correct set point. Flow sensor biases introduce more complex behavior depending on the magnitude of the flow bias and the direction of the bias. These are considered in more detail in Section 5.

4.3 Group Method of Data Handling (GMDH) Fault Detection

The Group Method of Data Handling (GMDH) is a technique to empirically model a process given a training set of process measurements (Ferreira 1999). For this research all process values used in training are results from the *simulated* heat exchanger control loop. The GMDH model is specific to a certain hardware configuration. If the amount of process data is limited, simulation may be combined with process data to generate GMDH models. The GMDH generates a polynomial-type fit to the data, without the need to specify the model complexity. The approach also allows the use of rational functions of the process variables as inputs.

Table 4.2. Extreme Values Caused by Faults*

Fault	SP(F,c) (kg/s)	Flow(c) (kg/s)	VP(c) (%)	CD(c) (%)	Flow(h) (kg/s)	VP(h) (%)	CD(h) (%)
T(h,out) Bias	↑	↑	↑	↑	-	-	-
HW Leak	-	-	-	-	↑	↑	↑
CW Leak	↑	↑	↑	↑	-	-	-
CW Valve Stuck (High)	-	-	Locked	↑	-	-	-
HW Valve Stuck (High)	-	-	-	-	-	Locked	↑
CW Valve Stuck (Low)	-	-	Locked	↓	-	-	-
HW Valve Stuck (Low)	-	-	-	-	-	Locked	↓

** (Up Arrow = Maximum, Down Arrow = Minimum)*

GMDH operates by building a model given a training set consisting of multiple process variables (list $X=\{x_1, x_2, \ldots x_m\}$) and their corresponding process output y. This formulation can be extended to the prediction of multiple outputs $\{y_1, y_2, \ldots, y_n\}$ as well. For a single output the general model has the form

$$y = a + \sum_{i=1}^{m} b_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} d_{ijk} x_i x_j x_k + \cdots \tag{4.1}$$

A typical node of a GMDH modeling layer is a basic quadratic predictor using variables $[x_i, x_j]$. The model parameters such as {A, B, C, D, E, F}, are estimated from a least-squares fit using N observations of the input and output variables. See section 4.3.1 for specifics on the GMDH algorithm. A model to estimate the variable y is constructed using the training set. This training set consists of many measured samples of y, each with a corresponding X. All input data must be scaled, or normalized to a range from 0 to 1 for the best fit. The GMDH algorithm uses a set of polynomial functions to build many models to estimate y using the input list X. The algorithm then tests each model against a test set, and discards inaccurate models. The best model is then stored with its calculated coefficients. Figure 4.1 shows the GMDH model structure with inputs (x) mapped to the final prediction through the layer structure.

4.3.1. Brief Overview of the GMDH Algorithm

The implementation of the GMDH algorithm used in this research is developed from Ferreira (1999) and the algorithm is from Farlow (1984). The reader is directed to these

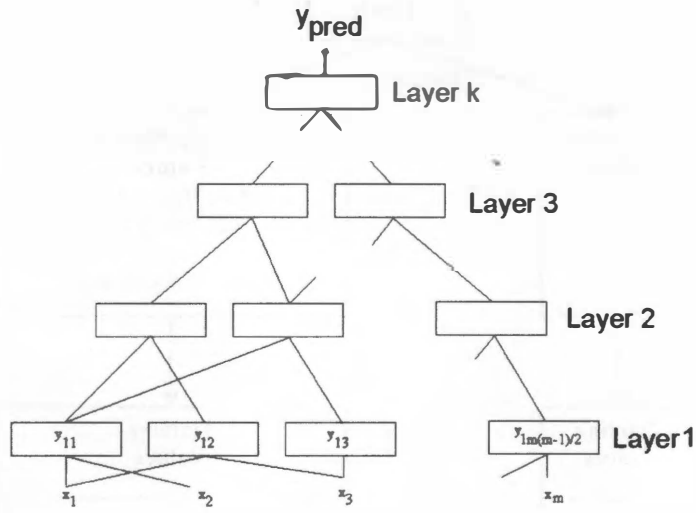

Figure 4.1. GMDH Layer Structure with k Layers (Ferreira 1999)

sources for details, the following is a brief overview of the GMDH algorithm. The initial step is to obtain and normalize the data so that the input range is 0 to 1. The data consists of the dependent y values $\{y_1, y_2, \ldots, y_n\}$ and independent x values $\{x_1, x_2, \ldots x_m\}$. Figure 4.2 shows how the data is arranged into inputs, training and test data. The following steps describe the GMDH algorithm.

1. Divide the data into the training and test sets as shown in Figure 4.2.
2. Compute the regression polynomial for each pair of training set input variables x_i and x_j. $y = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_ix_j$ (4.2)
3. For each regression computed above, build a matrix **Z** with new observations by evaluating the regression polynomial for all observations. This process creates 'improved' variables with better predictabilty.

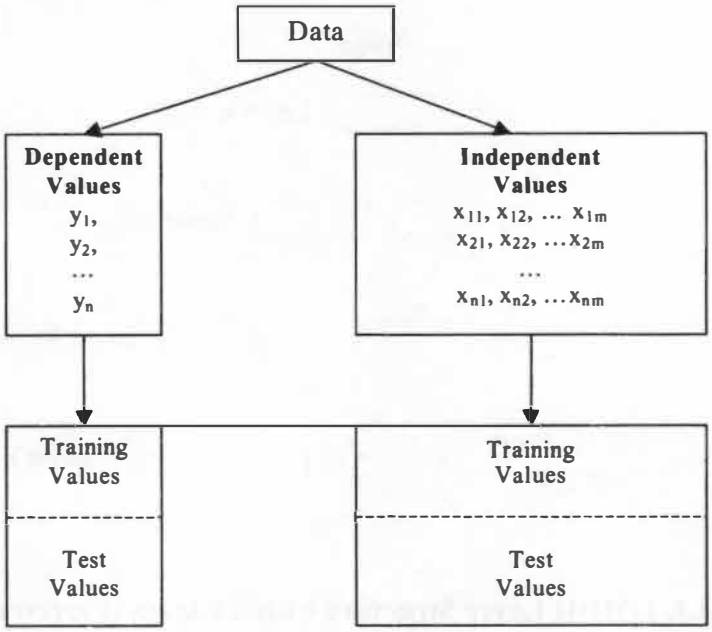

Figure 4.2. Arrangement of Input for GMDH Algorithm

4. Screen out the least effective variables by computing the root means square of each column of Z.

$$r_j^2 = \frac{\sum_{i=1}^{nt} (y_i - z_{ij})^2}{\sum_{i=1}^{nt} y_i^2} \tag{4.3}$$

5. Sort the matrix **Z** according to the r_j values, and keep all columns satisfying $r_j < R$ where R is some acceptance value given by the user.

6. Take the minimum r_j value for each generation. Test if the current r_j is large than that of the previous generation. If r_j has an increasing trend, the process should stop. Additional iterations may overfit the data. The minimum r_j corresponds to the minimum *Ivakhnenko* polynomial (Farlow 1984).

7. Use the test data to predict the y values, and determine if the GMDH model is adequate for the user's goals.

The GMDH model may now be used to generate predictions from given input data.

CHAPTER 5.

RESULTS AND ANALYSIS

The Temperature-to-Flow Cascade control (TFC) loop model simulates both normal and faulty operating conditions. The model covers a variety of initial conditions, including flow, temperatures, and hot water temperature set points. The behavior of the control loop as it attempts to control the hot water temperature is recorded for normal and faulty cases. The normal response is compared to the fault response, and an outcome table is generated. An up arrow on the outcome table indicates the magnitude of that process variable increased after the fault was initiated. A down arrow indicates the magnitude of the process variable decreased after the fault was initiated.

For an example of the output for the simulation loop, Figures 5.1 shows the system response for a temperature bias fault of 1.7 °C (3 °F) in the hot fluid at the outlet, with a tube side (hot water) flow of 0.05 kg/s (0.8 GPM). The bias error was introduced at t = 1,000 sec. Note that the plotted process variable T(h,out) is controlled by the Cascade Control Loop. Figure 5.2 demonstrates how the Cascade Controller responds to the fault. After the fault at t = 1,000 seconds the flow goes to a new steady-state value of 0.10 kg/s (1.6 GPM). The increase would appear as an up arrow in the fault outcome tables. Please note that the predicted transient is without meaning since parameters appropriate to allow transient simulation are not prescribed. The steady state solutions are the important part of the plots.

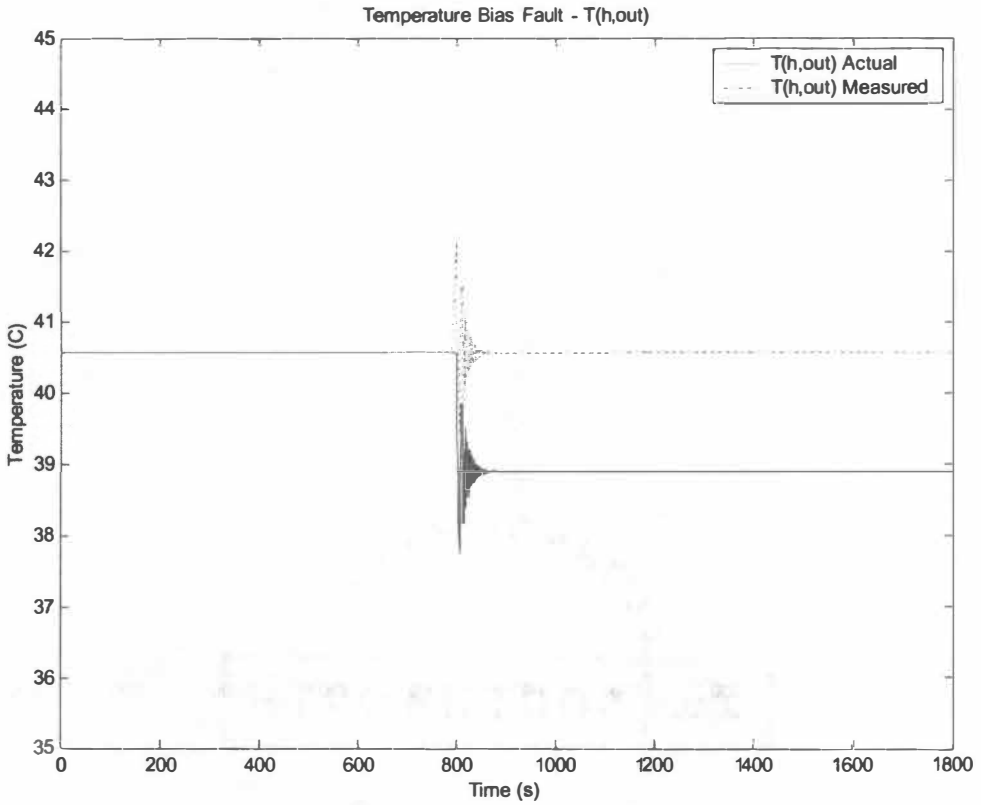

Figure 5.1. Measured and actual temperatures of the hot fluid at the outlet T(h,out).

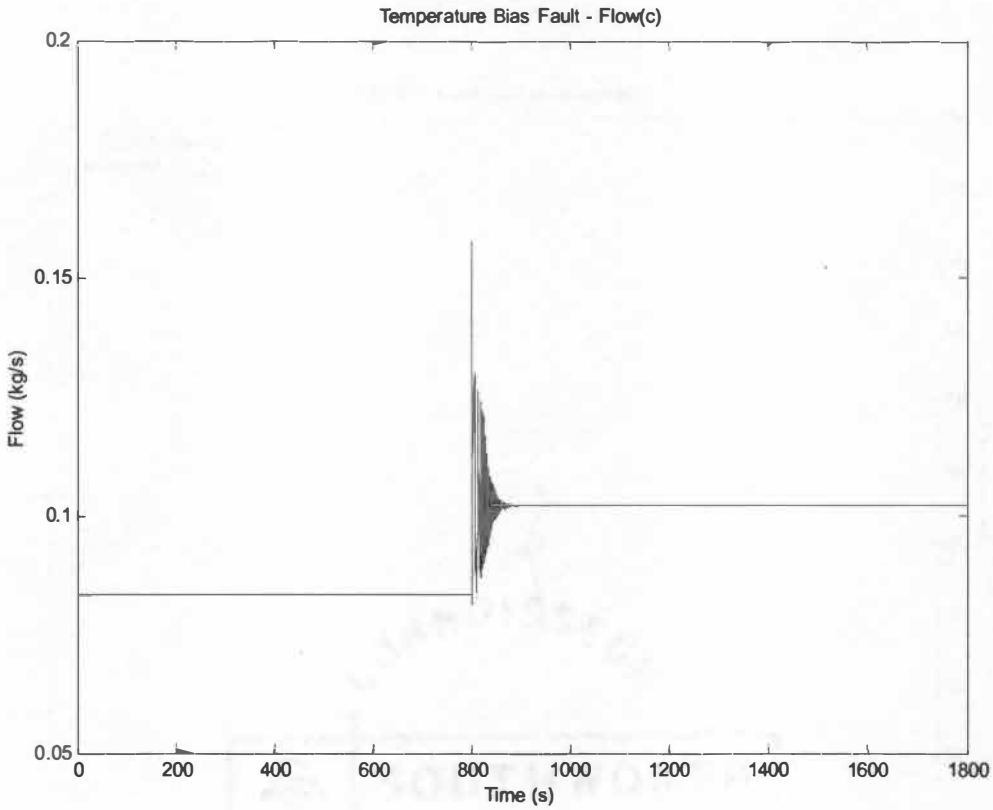

Figure 5.2. Measured Flow for Temperature Bias Fault at t=1,000 seconds.

5.1 Fault Outcome Tables

Five of the faults had the potential for different directions for the failures, such as a temperature sensor bias. The applied bias could be positive or negative. For the stuck valve fault, the valve could stick above or below the proper set point. These faults simply reverse arrows when the fault directions are reversed. Table 5.1 records the process variable response to all of the considered faults.

The results in Table 5.1 match well with the Emerson Process Management data from the test loop, shown in Table 4.1. The only inconsistencies occurred in the leak faults. The leak involved removing 30% of the hot water flow into the heat exchanger. The simulation does not model the pressure or flow changes that occur in the test loop following a leak, which may be important to properly simulate this event.

5.2 Group Method of Data Handling Fault Detection Results

Seven process variables from the heat exchanger control loop were used as inputs. Table 5.2 shows the variables and the corresponding inputs.

The GMDH input needs a minimum of three input variables. For the cases with insufficient input 'pseudo-variables' must be created to fill in. These pseudo-variables are simple variants of the process variables, such as $\sin(x_1)$ or x_1/x_2, where x_1 and x_2 are process variables. The GMDH algorithm determines this relationship and can accept the limited input.

Table 5.1. Fault Outcome Table for Simulations

Fault	T(c,out) (°C)	SP(F,c) (kg/s)	Flow(c) Measured	VP(c) (%)	CD(c) (%)	Flow(h) Measured	VP(h) (%)	CD(h) (%)	Del P(h) (psi)	Del P(c) (psi)
T(h) Bias (+)	↓	↑	↑	↑	↑					↑
T(h) Bias (-)	↑	↓	↓	↓	↓					↓
F(h) Bias (+)	↑	↓	↓	↓	↓		↓	↓	↓	
F(h) Bias (-)	↓	↑	↑	↑	↑		↑	↑	↑	↑
F(c) Bias (+)		↑	↑							
F(c) Bias (-)		↓	↓							
HW Leak							↑	↑	↓	↓
CW Leak				↑	↑					
HW Stuck Valve(+)		↑	↑	↑	↑	↑		↓	↑	↑
HW Stuck Valve (-)		↓	↓	↓	↓	↓		↑	↓	↓
CW Stuck Valve (+)	↓	↓	↑		↓					↑
CW Stuck Valve (-)	↑	↑	↓		↑					↓
HX Fouling*	↓	↑	↑	↑	↑					↑
Plugged Tubes*	↓	↑	↑	↑	↑					↑

**All the faults have unique outcome lists, except for fouling and plugged tubes. Both the fouling and plugged tubes have similar effects on the heat transfer performance in the heat exchanger.*

Table 5.2. GMDH Prediction Variables

Predicted Variable	Input Variables
T(h,out)	T(h,in), T(c,in), Flow(c), Flow(h)
T(c,out)	T(h,in), T(c,in), Flow(c), Flow(h)
SP(F,c)	T(h,in), T(c,in), SP T(h,out), Flow(h)
Flow(c)	Valve Positon(c), DelP(c)
VP(c)	Flow Set Point (c)
Flow(h)	Valve Positon (h), DelP(h)
VP(h)	Flow Set Point (h), Flow(h)

Figure 5.3 demonstrates the accuracy of the GMDH process when predicting the model results. The plot shows both the predicted and modeled cold-water flow rate. The values are normalized by setting the maximum value to 1.0. Figure 5.4. shows the corresponding error plot that shows the percentage difference between the model results and the GMDH predictions.

The heat exchanger control loop model is used to generate many cases of faulty operation. One hundred and twenty eight (128) cases of initial operating conditions were submitted to seven different faults, for a total of 896 cases. The GMDH algorithm then analyzed all the data, flagging the predicted process variables as high, expected or low. The algorithm graded each case, looking at the reported process variables and making predictions. The prediction results were graded with a +1,0 or –1. A high flag (+1) corresponds to a predicted value greater than the 106% of the reported value. An expected flag (0) is used when the prediction and reported values are within 6% of each other. The low flag is set when the prediction is below 94% of the reported value.

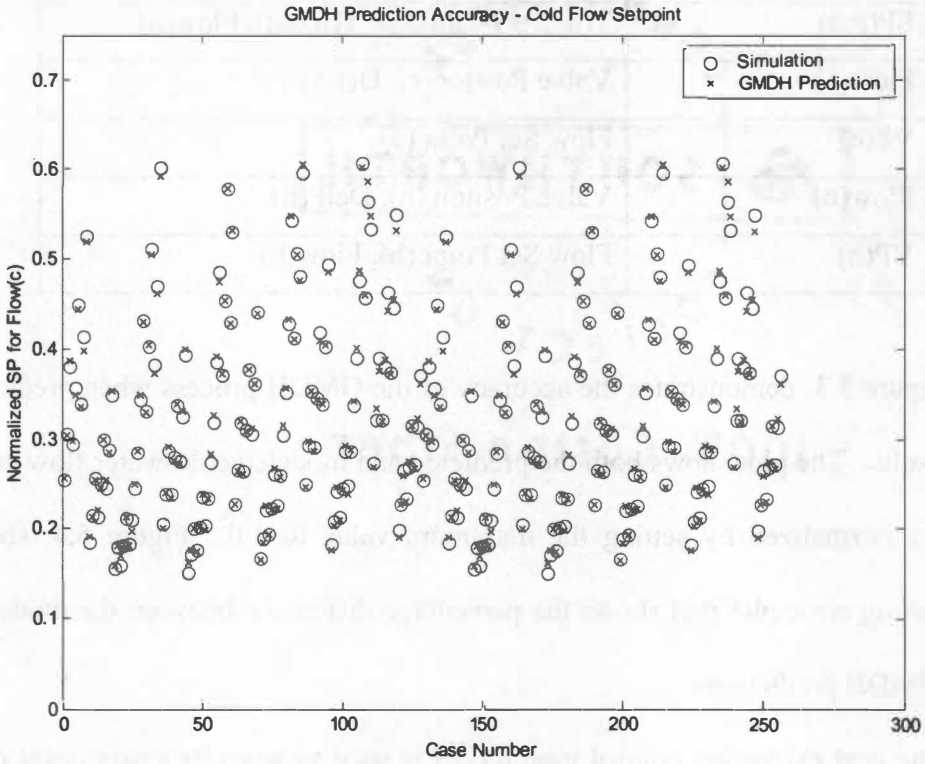

Figure 5.3. GMDH Predictions (x) plotted against the model results (o) for 256 test cases for the Cold Flow Setpoint Prediction.

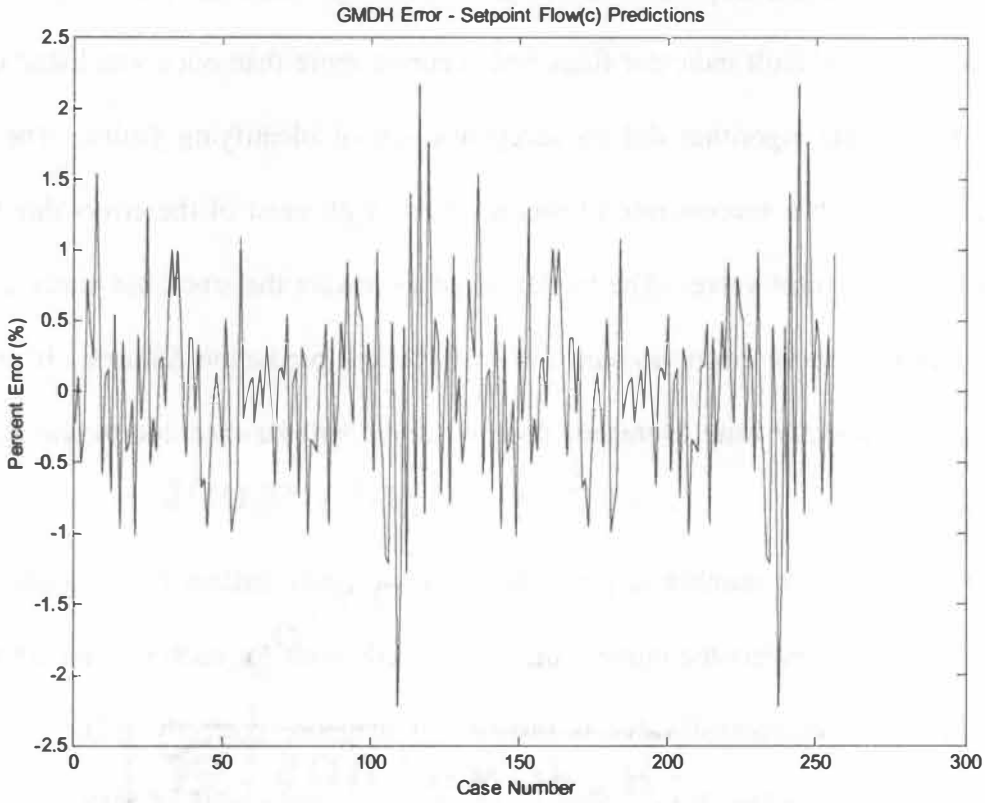

Figure 5.4. The percentage error from the GMDH prediction to the model results for the Cold Flow Setpoint Prediction. The error does not exceed 2.5%.

Table 5.3 shows the GMDH fault identifiers for the simulated faults designed to match the Emerson test loop. Several faults have multiple fault indicator patterns. For all faults, any set of fault indicator flags that occurred more than once was listed in this table. Overall, the algorithm did an acceptable job of identifying faults. The flags indicate a fault with a success rate of roughly 80%, with most of the errors due to the stuck hot water control valve. The model failed to predict the stuck hot water control valve completely, these errors account for 8 of 10 fault prediction failures. If process signals are sufficiently time averaged, false positives will be rare due to the training process of GMDH.

The confidence number is given for only the main indicator of a given fault. Confidence (%) represents the number of correct predictions for each set of fault cases. The stuck hot water control valve is difficult to diagnose correctly. The predicted variable changes caused by this simulated fault are on the order of the GMDH prediction accuracy. Therefore, the fault detection algorithm could not establish if a fault was present. Another significant limitation to using the GMDH simulation occurs when saturation or limit values for certain process variables occur during a fault. As discussed in Section 4.2, process variables such as valve position or temperature may not reach the desired set point in fault operation. The GMDH model often gives spurious results when saturated input is provided. This occurs because the transition from fault accommodation to saturation is very non-linear, and difficult to simulate using GMDH without careful training in the region of the process variable state space.

Table 5.3. GMDH Fault Identifiers

Fault	Confidence	T(h,out)	T(c,out)	SP(F,c)	Flow(c)	VP(c)	Flow(h)	VP(h)	DelP(h)	DelP(c)
T(h) Sensor Bias	91%			-1						
		-1		-1						
F(h) Sensor Bias	100%			1			-1	-1		
		1		1			-1	-1		
			1	1			-1	-1		
		1	1	1			-1	-1		
F(c) Sensor Bias	93%			-1	-1	1				
		-1		-1	-1	1				
			-1	-1	-1	1				
		-1	-1	-1	-1	1				
HW Leak	100%						-1	-1	1	
		1					-1	-1	1	
			1				-1	-1	1	
		1	1				-1	-1	1	
CW Leak	100%				1	-1				
					1	-1				Any
HW Stuck Valve*	0%									
CW Stuck Valve	100%					1				
				Any		1				

**No sufficient fault identifiers*

The GMDH method of fault detection is promising, despite its current limitations. To improve the prediction accuracy, a sufficiently large set of fault-free cases would be necessary for training. The current model uses 128 fault-free operation points, but more points over a wider range of operation are needed. Precaution must be taken not to include saturated input from fault cases, by possibly using a screening algorithm that would identify faults simply from the extreme values of the process varaibles. The GMDH method should also be tested with real data. In situations where the actual measurements are limited, combining these with simulation must be considered for data-driven model generation.

CHAPTER 6.

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS FOR FUTURE WORK

6.1 Summary

This thesis develops a multivariate analysis method for relating process variables in a heat exchanger control loop, a common fixture in almost all industrial plants. This method is used to create a Fault Detection and Isolation (FDI) system that detects and isolates faults and monitors performance degradation in heat exchanger control loops. Use of the proposed FDI system requires only a data-acquisition system and a small software package. Monitoring the condition of common equipment with pre-existing sensors provides an economical way to detect faults or degradation as they happen. Diagnosis of a fault allows remedial action. A typical heat exchanger control loop is simulated to demonstrate the FDI technique. A thermal-fluid model for the components in the system is also developed to relate flow rates and temperatures. The system includes a constant speed pump with flow control valves, pressure and temperature measurement. A proportional-integral-differential (PID) controller is used to maintain a temperature set point for the exit flow on one side of the exchanger. Measurements from an installed laboratory heat exchanger control loop similar to industrial equipment are also used for validation. The heat exchanger model proved to be accurate, and was generally within 5% of the measured outlet temperatures on the laboratory heat

exchanger. Further heat exchanger specific tuning can make the model even more accurate. From this simulation a data set was generated for data-based modeling.

6.2 Conclusions

A Fault Detection and Isolation (FDI) rule-base is formulated from results of simulations performed using mechanistic models. The rule base allows the identification of faults in a heat exchanger control loop given suitable process measurements. The faults imposed on the physical system and the model includes sensor biases, fluid leaks, unresponsive valves, plugged process lines, and fouling. The changes in steady-state process variables caused by the fault are used to generate tables that indicate process trends caused by the faults. The set of trend identifiers form a 'fault ID' that can categorize faults as they occur. Results for the physical system and model are presented. The simulation results indicate that unique identifiers exist for most faults. Two simulated faults, tube fouling and plugged tubes, share the same identifiers.

The mechanistic model matches the physical system performance well and is used to create a Fault Detection and Isolation (FDI) algorithm for the system. The mechanistic model generates data to develop a data driven model using the Group Method of Data Handling (GMDH). The GMDH is often used in FDI system development when good mechanistic models are not available. The GMDH model matches the mechanistic model well and the resulting FDI system remains useful. The simulated fault identifiers from the GMDH model match with the physical system fault identifiers for several of the faults. The sensor bias faults match very well. In situations where the actual measurements are

limited, combining data with mechanistic simulation must be considered for training data-driven models.

6.3 Recommendations for Future Work

The FDI system would be applicable to a wider range of systems if simulations and data for other heat exchangers were complete. The condenser model presented herein is a step towards extending the FDI method to two-phase situations and fluids other than water. The process variable inputs could be grouped in a different fashion, or studies could be done to determine the optimal measurement set. The GMDH model and prediction methods could be improved by generating more data over a wider range of operating conditions for many sizes of heat exchangers. Another important study would be comparison of the FDI rule base with real plant archived data from heat exchanger loops. New and on-line FDI schemes could be established by extending this work with new simulations or a different data-generation scheme.

LIST OF REFERENCES

LIST OF REFERENCES

Bott, T.R. et al., *Understanding Heat Exchanger Fouling and Its Mitigation,* Engineering Foundation Conference Proceedings, Pascoli, Italy (1997).

Butterworth D., "Condensation," *Hemisphere Handbook of Heat Exchanger Design*, G. F. Hewit, Ed. Hemisphere (1990).

Collier J. G. and J. R. Thome, Convective Boiling and Condensation. 3rd Ed. Claredon Press (1994).

Farlow, S. J., *Self-organizing Methods in Modeling: GMDH-type Algorithms*, M. Dekker, New York (1984).

Ferreira, P.B., *Incipient Fault Detection and Isolation of Sensors and Field Devices,* PhD Dissertation, The University of Tennessee (1999).

Handbook of Heat Exchanger Design, Geoffrey Hewett, Editor. Hemisphere (1990).

Hausen, H. Darstellung des Wärmeaubergangs in Rohren durch verallgemeinerte Potenzbeziehungen, *VDI Z.*, no. 4, p. 91 (1943).

Holman, J.P. *Heat Transfer*, 8th Ed., McGraw-Hill, New York (1997).

Huang, Xuedong. "Theoretical and Experimental Study of Degradation Monitoring of Steam Generators and Heat Exchangers." M.S. Thesis University of Tennessee (2003).

Nusselt, W. 'Die Oberflächenkondensation des Wasserdampfes'. Zeitschr. Ver. Deutsch. Ing., 60, 541, 569, (1916)

Petukhov, B.S. "Heat Transfer and Friction in Turbulent Pipe Flow with Variable Physical Properties," Edited by J.P. Hartnett, Advances in Heat Transfer, Academic Press (1970).

Schmidt F.W. et al., *Introduction to Thermal Sciences*, 2nd Ed., John Wiley & Sons (1993).

Sieder, E. N., and C. E. Tate. "Heat Transfer and Pressure Drop of Liquids in Tubes," *Ind. Eng. Chem.*, vol 28, p. 1429 (1936).

Tennessee Valley Authority, *Final Safety Analysis Report: Sequoyah Nuclear Plant* (1990).

Todreas, N.E. with M.S. Kazimi, *Nuclear Systems I, Thermal Hydraulic Fundamentals*, Taylor & Francis (1993).

Walker, G. *Industrial Heat Exchangers,* McGraw-Hill, New York (1982).

Yokell, S. , *A Working Guide to Shell-and-Tube Heat Exchangers,* McGraw-Hill, New York (1990).

Zukauskus, A. A., Heat Transfer from Tubes in Crossflow. *Adv. Heat Transfer*, vol. 8, pp. 93-160 (1972).

Additional Non-Cited References

Erbay, A.S. and B.R. Upadhyaya, "Fault Detection and Isolation of for Control Loop #2", University of Tennessee (1999).

Ferreira, P. B. and B.R. Upadhyaya, "On-line Fault Monitoring and Isolation of Field Devices Using Group Method of Data Handling," Proc. MARCON 99, Vol. 2, pp. 79.01-79.15 (1999).

Gupta J.P. *Fundamentals of Heat Exchangers and Pressure Vessel Technology*, Hemisphere Publishing, Washington (1986).

Mathworks, The. Manual for MATLAB Version 6 Release 12, The Mathworks, Inc. (2000).

Sawyer, et. al., "Development of a Data-Based Method for Performance Monitoring of Heat Exchangers" *Engineering Conferences International: Heat Exchanger Fouling and Cleaning* (2003).

APPENDICES

APPENDIX 1 – NOMENCLATURE

$\bar{h}$	Average heat transfer coefficient
$\dot{m}$	Mass flow rate
$\dot{M}$	Mass flow rate, unit width, $\dot{m} = \dot{M} / s$
μ	Dynamic viscosity
ε	Heat exchanger effectiveness
μ	Dynamic viscosity (kg/m s)
ΔT_m	Log mean temperature difference (°C)
A	Heat transfer area (m^2)
C_p	Specific Heat (kJ/kg K)
D	Diameter (tube)
d	Tube diameter (m)
f	Friction factor
F	Fouling Factor (m^2 °C/W)
g	Gravitational acceleration
h	Heat transfer coefficient (W/m^2 °C)
h_{fg}	Evaporation enthalpy
k	Thermal Conductivity (W/m K)
L	Tube length (m)
N	Number of tubes
Nu	Nusselt number
Pr	Prandtl number
r	Radius (m)
Re	Reynolds Number
s	Unit width, pi*D for tube
T, t	Temperature (°C)
U	Overall heat transfer coefficient (W/m^2 °C)
u	Velocity
z	Axial length

Subscripts

l	Liquid or film property
sat	Saturation temperature property
w	Wall property
film	Film property
L	Average or value at length L
loc	Local
g	Vapor-Gas phase
i	internal to the tube
in	inlet
m	mean

o	external to the tube
out	exit
w	properties evaluated at the wall temperature

Acronyms

EPM	Emerson Process Management
FDI	Fault Detection and Isolation
FSAR	Final Safety Analysis Review
GMDH	Group Method of Data Handling
HX	Heat Exchanger
TFC	Temperature to Flow Cascade
TVA	Tennessee Valley Authority

APPENDIX 2 - MATLAB CODE FOR HEAT EXCHANGER SIMULATION

The following files are included in the Heat Exchanger Control Loop Simulation Software Package:

genhx1.m

hxdata.mat (data file generated by genhx1.m)

hxloop.mdl

ntuhxm2.m

setloop.m

valve1.mat (valve position vs. flow rate data file)

vtf2.m

Description of Files

genhx1.m – Contains heat exchanger geometry, thermo-physical properties for fluids and materials involved. Stores data in file hxdata.mat.

hxloop.mdl – Simulink model for heat exchanger control loop, includes valve and controller models.

ntuhxm2.m – Heat exchanger simulation, needs hxdata.mat to operate.

setloop.m – Sets initial conditions for hxloop.mdl.

valve1.mat – Stores vector associated valve position to flow rate.

vtf2.m – Function uses valve1.mat to report flow rate given a valve position.

```
% genhx1.m

% Geometry Data Generation for NTUHXM2

% Loads all relevent geometry data into memory, including water property table.


%

% Water Property Lookup

% °F  °C Cp (kJ/kg*°Rho (kg/m^3Mu (kg/m*sk (W/m*°C Pr

water=[

50  10    4.195     999.2    1.31E-03   0.585  9.40

60  15.5  4.186     998.6    1.12E-03   0.595  7.88

70  21.1  4.179     997.4    9.80E-04   0.604  6.78

80  26.6  4.179     995.8    8.60E-04   0.614  5.85

90  32.2  4.174     994.9    7.65E-04   0.623  5.12

100 37.7  4.174     993.0    6.82E-04   0.630  4.53

110 43.3  4.174     990.6    6.16E-04   0.637  4.04

120 48.8  4.174     988.8    5.62E-04   0.644  3.64

130 54.4  4.179     985.7    5.13E-04   0.649  3.30

140 60.0  4.179     983.3    4.71E-04   0.654  3.01

150 65.5  4.183     980.3    4.30E-04   0.659  2.73

160 71.1  4.186     977.3    4.01E-04   0.665  2.53

170 76.6  4.191     973.7    3.72E-04   0.668  2.33];
```

```
%Wall Properties - Stainless 316L

% Below value is for 316.

k_wall = 13.4;   % Thermal Conductivity, W/(m*K)


% TUBE

tube_OD = 0.0024;

wall_thickness = 0.00025;

L = 0.406;              %Length of tubes (m)

N = 37;                 %Number of Tubes


% SHELL

shell_OD = 2.54E-02;            % (m)

shell_wall = 4E-03;             % (m)


% TOTAL HEAT TRANSFER AREA (given)

A = 0.11; % (m^2)


%**********************************************

% TUBE/HOT CALCULATIONS

tube_ID = tube_OD-wall_thickness;

r_outer = tube_OD/2;       %Radius, Outer

r_inner = tube_ID/2;       %Radius, Inner
```

```
Ax = pi*r_inner^2;        %Cross Sectional Area

% Determine Inner/Outer Areas

Ai = pi*tube_ID*N*L;

Ao = pi*(tube_OD)*N*L;

% SHELL/COLD CALCULATIONS

shell_ID = shell_OD - shell_wall; % (m)


% END

save hxdata
```

```
% ntuhxm2.m
% Version 2.0g
% Latest Update 7/15/02
%
% Model of a Heat Exchanger using the Number of Transfer Units (NTU) Method
%  for Emerson Process Management - Performance Technololgy Division
%
% CHANGES: Takes input as GPM, Degrees F
%        Iterative approach to determine Bulk Temps.
%        Improved Effectiveness Models
%
% Geometry specific to loop built by Emerson Process Managment
% Tuning Constants also specific to loop geometry.
%
% Developed by Aaron Sawyer, Undergraduate Research Assistant, UT Nuclear
Engineering
% Contact via e-mail: asawyer1@utk.edu
% Principle Investigators: Dr. Belle R. Upadhyaya and Dr. Arthur E. Ruggles
%
% ntuhxm2 takes as input the hot leg temperature in, cold leg temperature in, how leg
% flow rate, and
% cold leg flow rate.  Temperatures are in degrees F, flow rates are in GPM.
```

```
%
% Usage is as follows,
% ntuhxm(input_list);
%
% where input_list is a five element row vector containing
% [T_in t_in GPM_tube GPM_shell time]
% Captial T refers to hot/tube side, lowercase t refers to cold/shell side.
% Time is currently not important, use only positive values,
% EXCEPT to simulation fouling and plugged tubes, a value of -1 sets fouling,
% A value of -2 sets plugged tubes.
%
% A data file named 'hxdata' must be present and contain the geometry information for
the % heat exchanger.
% See 'genhx1.m' for more details.
%
function out = ntuhxm2(in_list);

% Breaks apart input list
T_in = (5/9)*(in_list(1)-32);
t_in = (5/9)*(in_list(2)-32);
GPM_tube = in_list(3);
GPM_shell = in_list(4);
```

```
% Time currently not important, except as debug usage
time = in_list(5);
% May be used as a flag, etc, to trigger fouling.

% Load water table and geometry
load hxdata;


if (time == -1) % Fouling
   F_inner = 0;
   F_outer = 0.0001;
else
   % FOULING FACTORS
   F_inner = 0;
   % F_inner is fouling factor for tube-side.
   F_outer = 0;
   % F_outer is fouling factor for shell-side.
   % Fouling factors are usually given in tables for various fluids and temperatures.
end


if (time == -2) % Plugged Tubes
   plugged = 3;  % Number of plugged tubes
   A = ((N-plugged)/N)*A;;   % Decreases available heat transfer area.
```

```
end

% Estimation of Bulk Temperatures
% Crude approximation, using NTU method, may be refined in future.
% Due to low rate of thermophysical property changes this should be acceptable.
rho_avg = interp1(water(:,2),water(:,4),(T_in+t_in)/2);
mdot_tube = GPM_tube*rho_avg*6.309E-05;
mdot_shell = GPM_shell*rho_avg*6.309E-05;
Cp = 4174;
Q = 0.5 * Cp * min([mdot_tube mdot_shell]) * (T_in - t_in);
T_bulk = (2 * T_in - Q/(Cp*mdot_tube))/2;
t_bulk = (2 * t_in + Q/(Cp*mdot_shell))/2;

% Bulk Temperature and Properties Iterative Loop, uses two steps to obtain
% accurate bulk temperatures.
for i = 1:2

% Tuning constants obtained using actual data to minimize the error in reported
temperatures.
% Obtained by Using Microsoft Excel Solver Function.
% C1 = 0.7034;
% C2 = 1.1656;
```

```
% Tuning constant reset to unity.

C1 = 1;

C2 = 1;


% Determine heat transfer coefficeints.

% Calls two subfunctions, subfunctions listed below main program.

hi = C1 * tubeh(mdot_tube,T_in,T_bulk,t_bulk);

ho = C2 * shellh(mdot_shell,t_bulk,T_bulk);


% Determine Thermal Resistances


R_tube = (F_inner*hi+1)/hi;

R_shell = (Ai/Ao)*(F_outer*ho+1)/ho;

R_wall = Ai * log(tube_OD/tube_ID)/(2*pi*L*N*k_wall);


% Filling out output vector for next function call.

Ch = mdot_tube * 1000 * interp1(water(:,2),water(:,3),T_bulk);

Cc = mdot_shell * 1000 * interp1(water(:,2),water(:,3),t_bulk);

% Determine U


U = 1/(R_tube + R_wall + R_shell);

UA = U*A;
```

```
% Heat Exchanger Modeling

% Given Parameters

% NTU Specific Expressions

Cmin = min([Ch Cc]);

Cmax = max([Ch Cc]);

Cr = Cmin/Cmax;

if Cr == 1

   Cr = 0.999;

end

NTU = UA/Cmin;

% Effectiveness calculation

% Strict counter-flow effectiveness

% Eff = (1-exp(-NTU*(1-Cr))) / (1-Cr*exp(-NTU*(1-Cr)));

% Alternate Effectiveness Expression - tube/shell model

Eff = 2*(1+Cr+(1+Cr^2)^(1/2)*(1+exp(-NTU*sqrt(1+Cr^2)))/(1-exp(-
NTU*sqrt(1+Cr^2))))^(-1);

%

Q = Eff * Cmin * (T_in - t_in);

T_out = T_in - Q/Ch;

t_out = t_in + Q/Cc;

% Suppressable outputs, for debugging.
```

```
%fprintf('\nT_Out: %0.2f',T_out);
%fprintf('\nt_out: %0.2f',t_out);

% OUTPUT FOR CERTAIN TEST CASES AND OPTIMIZATION
%if (time == 0) & (i == 2)
%    fprintf(' %6.1f  %6.1f  %5.1f  %5.1f',hi,ho,Ch,Cc);
%end

T_bulk = (T_in+T_out)/2;
t_bulk = (t_in+t_out)/2;
mdot_tube = GPM_tube*6.309E-05*interp1(water(:,2),water(:,4),T_bulk);
mdot_shell = GPM_shell*6.309E-05*interp1(water(:,2),water(:,4),t_bulk);

end % Bulk Iteration

out = (9/5)*[T_out t_out]+32;

%-----------END MAIN BODY

%-----------------------------------BEGIN SPECIFIC FUNCTIONS

%---------Tube Heat Transfer Coefficient-----------------------
```

```
function out = tubeh(mdot_tube,T_in,T_bulk,t_bulk);

% HX Model Prototype

% Determines inner heat transfer coefficent.

% TYPE: Tube-in-Shell Model

% Data File for h-inner determination

%

% Usage tubeh(mdot,Tin,Tout,Tcw)

%   mdot: Mass flow rate for tube fluid, kg/s

%   T_in: Temperature of tube fluid in, degrees C

% T_bulk: BULK Temperature of tube fluid, degrees C

% t_bulk: BULK temperature of shell fluid, degrees C


% Determines wall temp, approximations.

T_wall = (t_bulk+T_bulk)/2;


% Load water table and geometry

load hxdata;


%FLUID MODIFICATION

% In order to run with another set of fluid properties, places table as tube_fluid, same

layout as water table.

% Check temperature ranges, and uncomment line below.
```

```
% water = tube_fluid;


% Tube fluid (FW) Properties, evaluated at bulk temperature

%Interpolated Using Water Properties Table.

Cp_FW = interp1(water(:,2),water(:,3),T_bulk);      % Specific heat capacity, kJ/(kg*K)

rho_FW = interp1(water(:,2),water(:,4),T_bulk);     % Density, kg/m^3

mu_FW = interp1(water(:,2),water(:,5),T_bulk);      % Dynamic viscosity, kg/(m*s)

nu_FW = mu_FW/rho_FW;                               % Kinematic viscosity, m^2/s

k_FW = interp1(water(:,2),water(:,6),T_bulk);       % Thermal Conductivity, W/(m*K)

Pr_FW = interp1(water(:,2),water(:,7),T_bulk);      % Prandtl Number, unitless


% Properties evaluated at wall temperature

mu_wall = interp1(water(:,2),water(:,5),T_wall);  % Dynamic viscosity, kg/(m*s)


% Reynold's Number for inner tube flow.

G = mdot_tube/(N*Ax);

Re = G*tube_ID/mu_FW;

%fprintf('\nReynolds Number (tube): %0.1f\n',Re);

% 2300

if (Re > 0000)

  % Nusselt Number, using Petukhov Relation, p. 289 Holman

  % Assumes turbulent, check this.
```

```
if T_wall > T_bulk
   n = 0.11;
else n = 0.25;
end
% Friction factor, f, p. 246 in Schmidt
% f = 0.015;
% SMOOTH PIPE, p. 289 Holman
f = (1.82 * log10(Re - 1.64))^(-2);
%
Nu_FW = (f/8)*Re*Pr_FW / (1.07+12.7*(f/8)^(1/2)*(Pr_FW^(1/3)-1)) *
(mu_FW/mu_wall)^n;
else
% Laminar Tube Flow, Nusselt Number
test = Re * Pr_FW * 2 * tube_ID / L;
if test > 10
   Nu_FW = 1.86 * (Re * Pr_FW)^(1/3) * (tube_ID/L)^(1/3) *
(mu_FW/mu_wall)^(0.14);
else
   Nu_FW = 3.66 +
0.0668*(tube_ID/L)*Re*Pr_FW/(1+0.04*(tube_ID*Re*Pr_FW/L)^(2/3));
end
end
```

```
% DEBUG CODE - IGNORE
% f = (1.82 * log10(Re - 1.64))^(-2);
%     Nu_FW1  =  (f/8)*Re*Pr_FW  /  (1.07+12.7*(f/8)^(1/2)*(Pr_FW^(1/3)-1))  *
(mu_FW/mu_wall)^0.25
%     Nu_FW2  =  1.86  *  (Re  *  Pr_FW)^(1/3)  *  (tube_ID/L)^(1/3)  *
(mu_FW/mu_wall)^(0.14)
%                    Nu_FW3          =          3.66          +
0.0668*(tube_ID/L)*Re*Pr_FW/(1+0.04*(tube_ID*Re*Pr_FW/L)^(2/3))

% Calculates h_interior (FW)
Nu_FW;
h_i = Nu_FW * k_FW / tube_ID;
out = h_i;

%---------END tubeh ----------------------------------------

%---------Shell Heat Transfer Coefficient--------------------
function out = shellh(mdot_shell,t_bulk,T_bulk)
% HX MODEL
% Shell Heat Transfer Coefficeint Determination
%
% Usage tubeh(mdot,Tin,Tout,Tcw)
```

```
% mdot: Mass flow rate for shell fluid, kg/s
% T_in: Temperature of shell fluid in, degrees C
% Tout: Temperature of shell fluid out, degrees C

% Load geometry.
load hxdata;

%FLUID MODIFICATION
% In order to run with another set of fluid properties, places table as shell_fluid, same
layout as water table.
% Check temperature ranges, and uncomment line below.
% water = shell_fluid;

%
Twall = (T_bulk+t_bulk)/2;
%
% Determines cross sectional flow area
Axs = (pi/4) * (shell_ID)^2 - N * (pi/4) * (tube_OD)^2;
% Equivalent Diameter
De = (4*Axs)/(pi*shell_ID + N*pi*tube_OD);
% Hydralic Diameter
Dh = (4*Axs)/(N*pi*tube_OD);
```

```
%
% Reynolds Number Determination
G = mdot_shell/Axs;
% Interpolation for mu
mu = interp1(water(:,2),water(:,5),t_bulk);
mu_wall = interp1(water(:,2),water(:,5),Twall);
rho = interp1(water(:,2),water(:,4),t_bulk);
Pr = interp1(water(:,2),water(:,7),t_bulk);
Prw = interp1(water(:,2),water(:,7),Twall);
nu = mu/rho;  % Kinematic viscosity, m^2/s
% uinf - Free Stream Velocity
uinf = mdot_shell/(rho*pi*(shell_ID/2)^2);
%
PD = sqrt( 2*pi/(4*N*sqrt(3)) ) * (shell_ID/tube_OD);
%
umax = (uinf * PD/2)/(sqrt((PD/2)^2+PD^2)-tube_OD);
Re = rho*umax*tube_OD/mu;
if Re <= 100
   C = 0.9; n = 0.4;
   Nu = C * Re^n * Pr^0.36 * (Pr/Prw)^0.25;
elseif (Re > 100) & (Re < 1000)
  % Alternate Method, Crossflow across a cylinder
```

```
 if (Re < 40)
    C = 0.911; n = 0.385;
 else
    C = 0.683; n = 0.466;
 end
 % Three Possible Correlations for Nusselt Number
 %Nu = C * (uinf*tube_OD/nu)^n * Pr^(1/3);  % Correlaton 1
 Nu = (0.35 + 0.56 * Re^(0.52))*Pr^(0.3);  % Correlation 2
 %Nu = (0.43 + 0.50 * Re ^(0.5))*Pr^(0.38)*(Pr/Prw)^(0.25);  % Correlation 3
else % 1000 < Re < 2*10^5
 n = 0.6; C=0.27;
 Nu = C * Re^n * Pr^0.36 * (Pr/Prw)^0.25;
end
%fprintf('\nReynolds Number (shell): %0.1f\n',Re);
% P/D Ratio, Fixed by Geometry, or use
% PD = 1.4;
% DISABLED, Old way of doing things
% Nu = 10; % This values was picked visually of a graph, Todreas p. 428.


% Calculates h_interior (FW)
k = interp1(water(:,2),water(:,6),t_bulk);        % Thermal Conductivity, W/(m*K)
h_o = Nu * k / tube_OD;
```

```
CF = 0.97; %7 rows deep CF, need to put lookup table here for correction factors.

%out = 10*k/Dh; % Old way of modeling, w/ graph

out = CF*h_o;


%---------END shellh ----------------------------------------
```

```
% setloop.m

% SETUP FOR SIMULATION - V23.MDL

i=1;

r_case=[160    60       105      0.8];

% Input conditions

th_in = r_case(i,1);

tc_in = r_case(i,2);

thot_sp = r_case(i,3);

flow_sp = r_case(i,4);

% Faults off

% These control WHEN the faults occur.

% To have a hot water temperature bias at t=1000sec, set

% hwtf_on = 1000;

hwtf_on = 9999;

hwfl_on = 9999;

cwfl_on = 9999;

hw_leak = 9999;

cw_leak = 9999;

hw_stick = 9999;

cw_stick = 9999;

% Valve Stick positions

% May need to build a stuck valve list for multiple case runs.
```

```
hw_stickp = 50;

cw_stickp = 50;

% Plugged tubes/Fouling Flag

hx_err = 0;  % Set to -1 for fouling, -2 for plugged tubes

%

% Biases for sensor faults

hwtf = 0;  % T(h)

cwfl = 0;  % F(c)

hwfl = 0;  % F(h)
```

```
% Valve Position to Flow Finder
% Usage:
% vtf2(VP,side)
% Side = 0 if tube/HOT
%       1 if shell/COLD
% load valve1.mat containing tables relating valve position to flow rate.
% Vector VC = [ 5   0.01
%           vp%  flow
%           vp%  flow
%           100  1]; and so on, for VC and VH
function out = vtf2(VP,side)
load valve1.mat
if side
   % COLD WATER
   if VP > max(VC(:,1))
     VP = max(VC(:,1));
   elseif VP < min(VC(:,1))
     VP = min(VC(:,1));
   end
   %
   flow = interp1(VC(:,1),VC(:,2),VP);
   %
```

```
    if flow > max(VC(:,2))
      flow = max(VC(:,2));
    elseif flow < min(VC(:,2))
      flow = min(VC(:,2));
    end
    %
  else
    % HOT WATER
    if VP > max(VH(:,1))
      VP = max(VH(:,1));
    elseif VP < min(VH(:,1))
      VP = min(VH(:,1));
    end
    %
    flow = interp1(VH(:,1),VH(:,2),VP);
    %
    if flow > max(VH(:,2))
      flow = max(VH(:,2));
    elseif flow < min(VH(:,2))
      flow = min(VH(:,2));
    end
  end
```

```
out = flow;
```

APPENDIX 3 – GMDH MATLAB CODE FOR FAULT DETECTION

The following files are included in the GMDH fault dection package.

gdiag.m – Reports predicted GMDH values given the simulation results.

g_report – Generate fault ID flags for given simulation results.

The following GMDH files are needed to run the GMDH fault detection:

- ANNt_PBK.m
- GMDH_PBK.m
- Fun3.m
- MyBinaryGenerator.m
- Gmbase_1.mat – Empirical model structure, generated from above.

```
% GDIAG - GMDH Diagnosis, Outputs vector, based on input
% APS 2002 for Emerson FDI System
%
% Input to this program is a list vector with the HXLOOP simulation results,
% begining with T(h,in).  HXLOOP results are an 18-element vector.
%
% If the simualtion results include a time element, use the following syntax:
% prediction = gdiag(simout(index,2:19))
%
% Output will be a 9-element vector with the GMDH predictions.


function out = gdiag(in)
global MaxPower JobCounter;
MaxPower=11;
JobCounter=2047;
load gmbase_july


a = [];
b = a; c = a; d = a; e=a;
f = a; g=a; m = a; n = a;


%STRIP OUT ANY ZEROS, Prevents divide by zero error.
```

```
for i = 1:size(in,2)
   in(find(in(:,i) < 0.001),i)=0.001;
end

% T(h,out)
set_data = norml([in(1) in(2) in(9) in(12)],[1 1 2 2]);
a = [a ANNt_PBK(set_data,Layer_a,LayerOrder_a,Coef_a)];

% T(c,out)
set_data = norml([in(1) in(2) in(9) in(12)],[1 1 2 2]);
b = [b ANNt_PBK(set_data,Layer_b,LayerOrder_b,Coef_b)];

% SP(F,c)
set_data = norml([in(1) in(2) in(6) in(14)],[1 1 1 2]);
c = [c ANNt_PBK(set_data,Layer_c,LayerOrder_c,Coef_c)];

% Flow(c)
set_data = norml([in(10) in(18)],[3 5]);
set_data = [set_data set_data(1)./set_data(2)];
d = [d ANNt_PBK(set_data,Layer_d,LayerOrder_d,Coef_d)];

% VP(c)
```

```
set_data = norml([in(7)],[2]);

set_data = [set_data sin(set_data)];

set_data = [set_data set_data(1)./set_data(2)];

e = [e ANNt_PBK(set_data,Layer_e,LayerOrder_e,Coef_e)];


% Flow(h)

set_data = norml([in(15) in(17)],[3 4]);

set_data = [set_data set_data(1)./set_data(2)];

f = [f ANNt_PBK(set_data,Layer_f,LayerOrder_f,Coef_f)];


% VP(h)

set_data = norml([in(12) in(14) in(17)],[2 2 4]);

g = [g ANNt_PBK(set_data,Layer_g,LayerOrder_g,Coef_g)];


% DP(h)

set_data = norml([in(14) in(15)],[2 3]);

set_data = [set_data set_data(1)./set_data(2)];

m = [m ANNt_PBK(set_data,Layer_m,LayerOrder_m,Coef_m)];


% DP(c)

set_data = norml([in(9) in(10)],[2 3]);

set_data = [set_data set_data(1)./set_data(2)];
```

```
n = [n ANNt_PBK(set_data,Layer_n,LayerOrder_n,Coef_n)];


temp = normb([a' b' c' d' e' f' g' m' n'],[1 1 2 2 3 2 3 4 5]);
%
% This strips out any out-of-range results and sets them to 1000.
%
for i = 1:length(temp)
   if temp(i) > 1E3
      temp(i) = 1000;
   end
end
out = temp;
%out = [a' b' c' d' e' f' g' m' n']


function out=normb(in,ids)
% Normalizes a four input set for use in GMDH, used in gdiag
%
% Case setup for ids
% 1 - Temp
% 2 - Flow
% 3 - VP (%)
% 4 - DelP(h)
```

```
% 5 - DelP(c)
if size(in,2) ~= length(ids)
   fprintf('\nError: insufficient id for input.  See help. \n');
   out = -999;
else
   for i = 1:size(in,2)
      switch ids(i)
         case 1, in(:,i) = in(:,i).*120+50;
         case 2, in(:,i) = in(:,i).*4;
         case 3, in(:,i) = in(:,i).*100;
         case 4, in(:,i) = in(:,i).*0.35;
         case 5, in(:,i) = in(:,i).*8;
      end % end switch
   end %end for
   out = in;
end %endif

function out=norml(in,ids)
% Normalizes a four input set for use in GMDH, used in gdiag
%
% Case setup for ids
% 1 - Temp
```

```
% 2 - Flow
% 3 - VP (%)
% 4 - DelP(h)
% 5 - DelP(c)
if size(in,2) ~= length(ids)
   fprintf('\nError: insufficient id for input.  See help. \n');
   out = -999;
else
   for i = 1:size(in,2)
      switch ids(i)
         case 1, in(:,i) = (in(:,i) - 50)./120;
         case 2, in(:,i) = in(:,i)./4;
         case 3, in(:,i) = in(:,i)./100;
         case 4, in(:,i) = in(:,i)./0.35;
         case 5, in(:,i) = in(:,i)./8;
      end % end switch
   end %end for
   out = in;
end %endif
```

```
% GMDH Fault Modeling
%
% g_report takes as input a simulation result vector (or matrix)
% from the HXLOOP model. (19-elements per row)
%   The output is the fault ID flags for that input.
function out = g_report(sim_results)
%
% Load Data
%load dataset
% Dataset must include:
% sim_results2, post fault steady state conditions
%
reported = sim_results(:,[5 6 8 10 11 15 16 18 19]);
%
temp = [];
for i = 1:size(sim_results,1)
   temp = [temp; gdiag(sim_results(i,2:19))];
end


% Backup diagnostic results
temp2 = temp;
%
```

```
% Error level to trigger a fault ID flag.
zone = 0.06;
%
for i = 1:size(temp,1)
   for j = 1:size(temp,2)
      if temp(i,j) > (1+zone)*reported(i,j)
         temp2(i,j) = 1; %(reported(i,j)-temp(i,j))/reported(i,j);
      elseif temp(i,j) < (1-zone)*reported(i,j)
         temp2(i,j) = -1; %(reported(i,j)-temp(i,j))/reported(i,j);
      else
         temp2(i,j) = 0; %(reported(i,j)-temp(i,j))/reported(i,j);
      end
   end
end % end loop


% Output to text file the results.
% Reported Results
if 0
   s_list = {'T(h,o)','T(c,o)','SP Fc','F(c)','VP(c)','F(h)','VP(h)','DP(h)','DP(c)'};
   fid1 = fopen('g_report1.txt','wt')
   fprintf(fid1,'Reported Results\n');
   for i = 1:size(s_list,2)
```

```
    fprintf(fid1,'%7s ',s_list{i})
  end
  for i = 1:size(reported,1)
    fprintf(fid1,'\n');
    fprintf(fid1,'%7.3f ',reported(i,:))
  end
  fclose(fid1);
end %endif

% Output to text file the results.
% Predicted Results
if 0
  s_list = {'T(h,o)','T(c,o)','SP Fc','F(c)','VP(c)','F(h)','VP(h)','DP(h)','DP(c)'};
  fid1 = fopen('g_report2.txt','wt')
  fprintf(fid1,'Predicted Results\n');
  for i = 1:size(s_list,2)
    fprintf(fid1,'%7s ',s_list{i})
  end
  for i = 1:size(reported,1)
    fprintf(fid1,'\n');
    fprintf(fid1,'%7.3f ',reported(i,:))
  end
```

```
    fclose(fid1);
end %endif

% Output to text file the results.
% Fault ID Flags
if 0
    s_list = {'T(h,o)','T(c,o)','SP Fc','F(c)','VP(c)','F(h)','VP(h)','DP(h)','DP(c)'};
    fid1 = fopen('g_report3.txt','wt')
    fprintf(fid1,'Fault ID Flags\n');
    for i = 1:size(s_list,2)
        fprintf(fid1,'%7s ',s_list{i})
    end
    for i = 1:size(reported,1)
        fprintf(fid1,'\n');
        fprintf(fid1,'%7.3f ',reported(i,:))
    end
    fclose(fid1);
end %endif

% Output Fault ID Flags
out = temp2;
```

VITA

Aaron Sawyer was born and raised in Maryville, Tennessee, where he attended Maryville High School. The Bicentennial Scholarship facilitated his undergraduate studies at the University of Tennessee, where he received his Bachelor's degree in Nuclear Engineering in May 2002. He continued his studies at the University of Tennessee and will receive his Master's degree in Nuclear Engineering in May 2004. Following completion of his graduate degree, he will be employed at Bettis Laboratory in Pittsburgh, Pennsylvania, working on advanced concept designs for nuclear propulsion. He has several technical publications, including "Fluid Transients in Mercury and Mercury-Helium Flows," for the ASME Heat Transfer/Fluids Engineering 2004 summer conference and "Development of a Data-Based Method for Performance Monitoring of Heat Exchangers," for the 2003 ECI Fouling Conference.

5990 2949 48
06/23/04 MRB