MODELING AND CONTROL OF A NOVEL SEMI-CLOSED GAS TURBINE-ABSORPTION COMBINED CYCLE

By

CHOON JAE RYU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

To my parents, Jiyeon, and Hahnerl

ACKNOWLEDGMENTS

I acknowledge the support and guidance of my doctoral advisor, Dr. William E. Lear Jr. He was helpful throughout my time at the University of Florida, encouraging me to think in all different directions and study challenging research topics with the right path. I would like to thank my other doctoral committee members for their time and patience spent with me. I appreciate the dynamic and control engineering knowledge Dr. Oscar Crisalle has given me. In addition, I must thank Dr. Sherif, Dr. Ingley, and Dr. Svoronos for their willingness to participate in my doctoral review process.

I thank many colleagues working with me in Dr. Lear's lab. Dr. Khan and Dr. Singh taught me many important lab procedures during my first days in the lab. David was always helpful to work together in the lab and he was a good English teacher. Minki was supportive during my last days in the lab. Additionally, I appreciate a friendship of long standing with Cheng-Chan and Sydni in fuel cell lab.

I also thank my family for their support and understanding. My parents always encourage me with their constant trust. My brother and sister-in-law have always helped me. I must thank my parents-in-law for their support to my wife and me during out stay in USA. I really appreciate patient understanding of my dear wife, Jiyeon Lee. She has always inspired me and been helpful. She is also my good discussion partner. I thank her for her constant love and support. Lastly, I thank my precious and adorable baby, Hahnerl.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 8

LIST OF FIGURES 9

NOMENCLATURE 13

ABSTRACT 19

CHAPTER

1 INTRODUCTION 21

Semi-Closed Cycle System 21
Vapor Absorption Refrigeration System 22
Statement of the Research Problem 23
Thermodynamic Properties of Ammonia-Water Mixtures 23
Process Models 24
Organization of This Document 24

2 LITERATURE REVIEW 26

Thermodynamic Properties of Ammonia-Water Mixtures 26
Process Models 28

3 THERMODYNAMIC PROPERTIES OF AMMONIA-WATER MIXTURES 30

Gibbs Free Energy for Ammonia-Water Mixture 30
Gibbs Free Energy for Pure Components 30
Vapor and Liquid Gibbs Free Energy for the Binary Mixtures 32
Pressure-Temperature-Composition Properties of the Binary Mixtures 33
Results and Discussion 35
Comparison with Gillespie's Data 35
Comparison with IGT Experimental Data 36
Saturated liquid phase 36
Saturated vapor phase 36
Comparison with IGT Smoothed Data 37
Temperature Discrepancies for Pure Components 38
Summary 39

4 SYSTEM DESIGN OF A REFRIGERATION SYSTEM 48

Second Law Analysis Using Constant Internal Temperatures 48

Thermodynamic Model 48
Solution Method 51
Results and Discussion 53
Summary 56
Second Law Analysis Using Variable External Temperatures 56
Thermodynamic Model 57
Solution Method 62
Results and Discussion 64
Summary 67

5 DYNAMIC MODELING WITH CONSTANT AMMONIA MASS FRACTION APPROACH 89

Thermodynamic Model 89
Heat Recovery Vapor Generator 91
Rectifier 91
Condenser, Absorber, and Solution Heat Exchanger 92
Evaporator 92
Pump 93
Thermal Expansion Valve 93
Refrigerant Heat Exchangers 93
Ice Maker and Storage 96
Solution Method 96
Results and Discussion 98
Consistency of First Law and Second Law Models 100
Summary 101

6 DYNAMIC MODELING FOR TWO-COMPONENT FLUIDS 113

Introduction to Dynamic Heat Exchanger Modeling 113
Moving Boundary Model for Single Component Fluid 113
Two-Component Fluids 117
Two-Phase Model Options 118
Thermodynamic Models of VARS for Two-Component Fluids 120
Condenser 121
Absorber 128
Generator 133
Evaporator 137
Solution Heat Exchanger 140
Results and Discussion 142
Condenser 143
Absorber 144
Generator 145
Evaporator 145
Solution Heat Exchanger 146
Summary 147

7 CONCLUSIONS 162

Thermodynamic Properties of Ammonia-Water Mixtures 162
Preliminary Design of VARS 163
Dynamic Modeling of Heat Exchangers 164
Future Work 165

APPENDIX

A COMPUTER CODE FOR THEMODYNAMIC PROPERTIES OF AMMONIA-WATER MIXTURES 167

B COMPUTER CODE FOR DYNAMIC MODELING WITH CONSTANT AMMONIA MASS FRACTION APPROACH 175

C GOVERNING EQUATIONS AND COMPUTER CODE FOR MOVING BOUNDARY MODEL 178

Governing Equations for Moving Boundary Model 178
Derivations and Coefficients of Single-Component Evaporator Model 178
Coefficients of Two-Component Condenser Model 183
Coefficients of Two-Component Absorber Model 185
Coefficients of Two-Component Generator Model 186
Derivations and Coefficients of Two-Component Evaporator Model 188
Coefficients of Two-Component SHX Model 189
Computer Code for Moving Boundary Model 190
Condenser 191
Absorber 200
Generator 210
Evaporator 219
Solution Heat Exchanger 227

LIST OF REFERENCES 235

BIOGRAPHICAL SKETCH 239

LIST OF TABLES

Table		page
3-1	Coefficients for equations (3-1) and (3-2) for pure water and ammonia	40
3-2	Coefficient for Gibbs excess energy function	40
3-3	Comparison of dew point temperatures	40
4-1	Input/Output parameters of the HPRTE system.	69
4-2	Comparison of results	69
4-3	Input parameters of the VARS	70
4-4	Input/output parameters of the HPRTE system.	70
4-5	Ice produced a day [Ton/day (L/day)].	70
5-1	Design values or states at each state point	103
5-2	Results for higher air conditioning load	104
5-3	Results for lower air conditioning load	105
5-4	Consistency of the results between Chapters 4 and 5	106
6-1	Parameters of the condenser	148
6-2	Parameters of the absorber	148
6-3	Parameters of the generator	148
6-4	Parameters of the evaporator	149
6-5	Parameters of SHX	149
A-1	Matlab files for thermodynamic properties of ammonia-water mixtures	167
B-1	Matlab and Simulink files for dynamic modeling with constant ammonia mass fraction approach.	175

LIST OF FIGURES

Figure		page
3-1	Comparison P-T-x relations with Gillespie's data at 313.15K	41
3-2	Comparison P-T-x relations with Gillespie's data at 405.9K	41
3-3	Comparison for saturated liquid enthalpy	42
3-4	Comparison with IGT data for saturated vapor temperature at 1654.74kPa	42
3-5	Comparison with IGT data for saturated vapor enthalpy at 1654.74kPa	43
3-6	Comparison with IGT data for saturated vapor and liquid temperature	43
3-7	Comparison with IGT data for saturated vapor and liquid enthalpy	44
3-8	Saturated temperatures for high ammonia mass fraction at 344.74kPa	44
3-9	Saturated temperatures for high ammonia mass fraction at 1034.21kPa	45
3-10	Saturated temperatures for high ammonia mass fraction at 1723.69kPa	45
3-11	Temperature discrepancies at pure components	46
3-12	Discrepancy and oscillation of El-Sayed and Tribus's P-T-x	46
3-13	Discrepancy and oscillation of Pátek and Klomfar's P-T-x	47
4-1	Gas turbine flowpath of the PoWER system	71
4-2	The multi-stage vapor absorption refrigeration system	71
4-3	The multi-stage simple vapor absorption refrigeration system	72
4-4	Air conditioner hourly load for typical summer day	72
4-5	Typical variations of outdoor air temperature at 42° north latitude on July 1	73
4-6	Carnot COP variations for the three evaporators	73
4-7	COP variations for three evaporators at η_R=0.5	74
4-8	COP variations for three evaporators at η_R=0.7	74
4-9	Total COP variations of the system at η_R=0.5	75
4-10	Total COP variations of the system at η_R=0.7	75

4-11 Heat removed from surroundings at η_R=0.5, R_{evp}=0.3 76
4-12 Allocated heat supply from generator at η_R=0.5, R_{evp}=0.3 76
4-13 Heat removed from surroundings at η_R=0.7, R_{evp}=0.3 77
4-14 Allocated heat supply from generator at η_R=0.7, R_{evp}=0.3 77
4-15 Heat removed from surroundings at η_R=0.5, R_{evp}=0.5 78
4-16 Heat removed from surroundings at η_R=0.7, R_{evp}=0.5 78
4-17 Allocated heat supply from generator at η_R=0.5, R_{evp}=0.5 79
4-18 Allocated heat supply from generator at η_R=0.7, R_{evp}=0.5 79
4-19 Heat removed from surroundings at η_R=0.5, R_{evp}=0.7 80
4-20 Heat removed from surroundings at η_R=0.7, R_{evp}=0.7 80
4-21 Allocated heat supply from generator at η_R=0.5, R_{evp}=0.7 81
4-22 Allocated heat supply from generator at η_R=0.7, R_{evp}=0.7 81
4-23 Ice produced hourly for various R_{evp} at η_R=0.5..... 82
4-24 Ice produced hourly for various R_{evp} at η_R=0.7..... 82
4-25 The multi-temperature simple vapor absorption refrigeration system 83
4-26 Four cases of external temperatures 83
4-27 COP for a small control volume when A) generator temperature varies and B) Evaporator 1 temperature varies 83
4-28 Carnot COP for Evaporator 1 84
4-29 Carnot COP for Evaporator 2 84
4-30 Carnot COP for Evaporator 3 85
4-31 Heat removed at Evaporator 2..... 85
4-32 Heat removed at Evaporator 3..... 86
4-33 Heat required for Sub-system 1 86
4-34 Heat required for Sub-system 2..... 87
4-35 Heat required for Sub-system 3..... 87

4-36 Efficiency ratio based on Case (a) efficiency 88
5-1 Extended vapor absorption refrigeration system of the PoWER system 107
5-2 Simple shell and tube heat exchanger 107
5-3 Lumped wall model 108
5-4 Control volumes for A) RHX1 and B) RHX2 108
5-5 Extended VARS Simulink model 108
5-6 Air conditioning hourly load 109
5-7 Mass change of the stored ice [kg] 109
5-8 Chilled water use [kg/s] 110
5-9 Heat gained from surroundings at three evaporators 110
5-10 Coefficient of performance (COP) 111
5-11 Temperature of RHX1 111
5-12 Temperature of RHX2 112
6-1 A schematic of an evaporator model for single component flow 150
6-2 Two-phase flow temperature distribution in an evaporator 150
6-3 Saturated vapor ammonia mass fraction in an evaporator 151
6-4 Saturated liquid ammonia mass fraction in an evaporator 151
6-5 Enthalpy-Ammonia mass fraction (h-x) diagram 152
6-6 A schematic of a condenser 152
6-7 A schematic of an absorber 153
6-8 A schematic of a generator 153
6-9 A schematic of an evaporator 154
6-10 A schematic of SHX 154
6-11 Inlet ammonia mass fraction step increase for the condenser 155
6-12 Inlet mass flow rate step increase for the condenser 155

6-13 Secondary fluid inlet mass flow rate step increase for the condenser 156
6-14 Inlet ammonia mass fraction step increase for the absorber 156
6-15 Inlet enthalpy step increase for the absorber..... 157
6-16 Inlet mass flow rate step increase for the absorber 157
6-17 Secondary fluid inlet mass flow rate step increase for the absorber..... 158
6-18 Inlet ammonia mass fraction step increase for the generator 158
6-19 Inlet mass flow rate step increase for the generator 159
6-20 Inlet ammonia mass fraction step increase for the evaporator 159
6-21 Inlet enthalpy step increase for the evaporator..... 160
6-22 Inlet mass flow rate step increase for the evaporator 160
6-23 Colder side inlet ammonia mass fraction step increase for SHX 161
6-24 Colder side inlet mass flow rate step increase for SHX..... 161
B-1 Simulink model for Chapter 5. 177
C-1 Condenser Simulink model..... 191

NOMENCLATURE

A	Area [m^2]
C_p	Specific heat capacity at constant pressure [kJ/kg·K]
COP_{carnot}	Carnot COP
D	Diameter of tube [m]
G	Specific Gibbs free energy [kJ/kg]
H	Heat transfer coefficient between tube wall and fluid [kW/m^2]
L	Heat exchanger length [m]
L^*	Imaginary moving boundary position [m]
P	Pressure [kPa]
$\dot{Q}$	Heat transfer rate [kW]
$\dot{Q}''$	Heat transfer rate per area [kW/m^2]
R	Gas constant [kJ/kmole·K]
R_{evp}	Refrigeration ratio
T	Temperature [K] or Transpose of matrix
UA	Area universal heat transfer coefficient [kW/K]
UD	Length universal heat transfer coefficient [kW/K·m]
V	Volume [m^3]
$\dot{W}_p$	Pump work [kW]
c	Concentration
d	Vapor quality or dryness
f	Body force [N]
h	Specific enthalpy [kJ/kg]
l	Interface position or length [m]

m	Mass [kg]
$\dot{m}$	Mass flow rate [kg/s]
q_{latent}	Latent heat of fusion of water [kJ/kg]
s	Specific entropy [kJ/kg·K]
t	Time [sec]
u	Fluid velocity [m/s]
$\mathbf{u}$	Input matrix
v	Specific volume [m^3/kg]
x	Total ammonia mass fraction or Ammonia mass fraction in liquid
$\mathbf{x}$	State variable matrix
y	Ammonia mass fraction in vapor
$\mathbf{y}$	Output matrix
z	Length [m]

Greek

α	Stress tensor [N/m^2]
α_i	Heat transfer coefficient between tube wall and refrigerant [kW/m^2]
α_o	Heat transfer coefficient between tube wall and ambient [kW/m^2]
ρ	Density [kg/m^3]
η_p	Pump efficiency
η_R	Second law efficiency
η_{th}	Thermal efficiency
τ	Characteristic time [sec] or Shear stress tensor [N/m^2]

Superscripts

*	Non-dimensional variables or imaginary moving boundary
L	Liquid phase
G	Gas phase
E	Excess energy

Subscripts

AC	Air conditioning
E	Evaporator
ICE	Ice storage
R	Refrigeration
a	Ambient, Absorber, or Ammonia
$a12$	Interface in absorber
aa	Ambient for absorber
ai	Absorber inlet
ao	Absorber outlet
ar	Refrigerant in absorber
avg	Length average
b	Bubble point
$c12$	Interface in condenser
c	Condenser or Carnot
ca	Ambient for condenser
ci	Condenser inlet
co	Condenser outlet
cr	Refrigerant in condenser

cw	Critical point for water or Chilled water
d	Dew point
e	Equilibrium or Evaporator
ea	Ambient for evaporator
eg	Saturated vapor for thermodynamic equilibrium
ei	Evaporator inlet
el	Saturated liquid for thermodynamic equilibrium
eo	Evaporator outlet
er	Refrigerant in evaporator
g	Gas phase or Generator
*g*12	Interface in generator
ga	Ambient for generator
gi	Generator inlet
go	Generator outlet
gr	Refrigerant in generator
i	Inlet
int	Interface between two-phase and single-phase
m	Mixture
o	Reference values, Outlet, or Ambient
p	Pump
r	Refrigerant or Reduced values
rc	Colder side in RHX
rci	Colder side inlet in RHX
rco	Colder side outlet in RHX

rh	Hotter side in RHX
rhi	Hotter side inlet in RHX
rho	Hotter side outlet in RHX
s	SHX or Steady state
sc	Colder side in SHX
sci	Colder side inlet in SHX
sco	Colder side outlet in SHX
sg	Saturated vapor
sh	Hotter side in SHX
shi	Hotter side inlet in SHX
sho	Hotter side outlet in SHX
sl	Saturated liquid
w	Wall or Water

Abbreviations

ABS	Absorber
AC	Air Conditioning
CGC	Cold Gas Cooler
CM	Combustor
COND	Condenser
COP	Coefficient of Performance
CW	Chilled Water
DE	Distributed Energy
EVAP	Evaporator
GP	Gas Path

HGC	Hot Gas Cooler
HPC	High Pressure Compressor
HPRTE	High Pressure Regenerative Turbine Engine
HPT	High Pressure Turbine
HRVG	Heat Recovery Vapor Generator
h-x	Enthalpy-Ammonia Mass Fraction
IGT	Institute of Gas Technology
LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
ODE	Ordinary Differential Equation
P	Pump
PDE	Partial Differential Equation
PoWER	Power, Water Extraction, and Refrigeration
P-T-x	Pressure-Temperature-Composition
RECT	Rectifier
RHX	Refrigerant Heat Exchanger
SHX	Solution Heat Exchanger
SOFC	Solid Oxide Fuel Cell
TXV	Thermal Expansion Valve
VARS	Vapor Absorption Refrigeration System
WGC	Warm Gas Cooler

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

MODELING AND CONTROL OF A NOVEL SEMI-CLOSED GAS TURBINE-ABSORPTION COMBINED CYCLE

By

Choon Jae Ryu

May 2011

Chair: William E. Lear Jr.
Cochair: Oscar Crisalle
Major: Mechanical Engineering

The Power, Water Extraction, and Refrigeration (PoWER) engine has been investigated for several years as a distribute energy (DE) system, among other applications, for civilian or military use. Previous literature describing its modeling and experimental demonstration have indicated several benefits, especially when the underlying semi-closed cycle gas turbine is combined with a vapor absorption refrigeration system (VARS), the PoWER system described herein. The benefits include increased efficiency, high part-power efficiency, small lapse rate, compactness, decreased emissions, reduced air and exhaust flows (which decrease filtration and duct size) and condensation of fresh water.

In this research the preliminary design and modeling of a modified version of this system as applied to DE is the focus, a system especially useful in regions prone to major grid interruptions due to hurricanes, undercapacity, or terrorism. In such cases, the DE system should support most or all services within an isolated service island, including ice production, so that the influence of the power outage is limited in scope. This research also describes the rather straightforward system modifications necessary

for ice production and the use of this ice-making capacity to achieve significant load-leveling during the summer utility peak, hence reducing the electrical capacity requirements for the grid Load-leveling strategies are also discussed.

For this complex system, maximum efficiency and safe operation at any instant of time is the ultimate goal of an advanced control algorithm. To develop such advanced system controls, dynamic modeling is necessary. As part of this research, thermodynamic properties of ammonia-water mixtures and their behavior in a VARS are improved and applied to dynamic modeling of a VARS unit. A conventional moving boundary dynamic heat exchanger model is extended for two-component two-phase flow which exists in the VARS.

CHAPTER 1
INTRODUCTION

Semi-Closed Cycle System

Currently, most countries generate their electricity in large centralized facilities, such as coal, nuclear, hydropower or gas powered plants. These plants have excellent economies of scale, but usually transmit electricity long distances and/or are often considered to be too far away for their waste heat to be used effectively. To avoid these problems, construction of intermediate-size district heating plants near a city is one possible solution, often employed in northern European countries.

Distributed generation is another popular approach, in which small distributed energy resources such as wind turbines, fuel cells, solar cells, reciprocating engines, and microturbines installed near loads can produce electricity without significant transmission loss. Some of these can also take advantage of waste heat to meet local heating requirements. Of the various distributed energy systems, gas turbines are more reliable and require less maintenance than others. They are compact and produce low emissions and high quality heat but are expensive and the efficiencies of small gas turbines are typically lower than some other options.

Combined cycle power generation and/or cogeneration are used in larger plants to increase efficiency. The waste heat can drive steam turbines, absorption chillers, or other cycles. The current paper deals with a novel type of semi-closed cycle gas turbine engine which inherently favors coupling to bottoming cycles or to cogeneration. Anxionnaz [1-2] proposed semi-closed gas turbine cycles due to their potential advantages in low fuel consumption over their power range and significantly reduced airflow requirements. Lear and Sherif [3] patented the concept of integrating a semi-

closed gas turbine system, called High Pressure Regenerative Turbine Engine (HPRTE), with a Vapor Absorption Refrigeration System (VARS) to generate the Power, Water Extraction, and Refrigeration (PoWER) system. This system is the focus of the current research study.

In order to understand this complex system and control it optimally, accurate process models for each component must be developed and physical properties of working fluids must be well understood. After the dynamic behavior of the coupled VARS and HPRTE is properly modeled, various control logic can be applied to the integrated system.

Vapor Absorption Refrigeration System

VARS units using ammonia-water as the working fluid were popular and widely used before the development of the vapor compression refrigeration system, which has a higher coefficient of performance (COP) than the VARS. The VARS, however, can use low-grade waste heat energy rather than electricity, so the lower COP is unsurprising. Solar or geothermal energy or waste heat from power plants is of sufficiently high temperature to drive a VARS unit.

Various refrigerant-absorbent pairs can be used for the VARS refrigeration system, depending on the temperatures to be controlled. Two common working fluids for the VARS are lithium bromide-water (LiBr-H_2O) and ammonia-water (NH_3-H_20) mixtures. Lithium bromide-water refrigeration systems use water as their refrigerant. Therefore, these systems cannot be used below freezing, but are suitable for air conditioning or other systems operated above freezing temperatures. For ammonia-water refrigeration system, ammonia is the refrigerant and water is the absorbent. Ammonia is an excellent refrigerant not only for air conditioning but also for freezing. Since the absorbent, water,

is volatile, distillation columns need to be included in the cycle in order to purify the ammonia sufficiently to attain high efficiency.

Even though the ammonia-water VARS has been under development for over 100 years, there is still the potential to improve performance. Better understanding of the ammonia-water mixture behavior and a thermodynamically consistent property model would significantly improve the accuracy and robustness of process models. These mathematical models are necessary in order to create an advanced controller for the multivariate system.

Statement of the Research Problem

In this research, dynamic modeling of a VARS and the control the system, thermodynamic properties of ammonia-water mixtures, modeling methods, and control schemes have been explored. All aspects of the research are in support of the development of advanced physics-based control for the PoWER integrated system, including consideration of multiple applications with time-varying objective functions.

Thermodynamic Properties of Ammonia-Water Mixtures

There are various mathematical models for ammonia-water mixtures in common use for research or design purposes. Fugacity and chemical potential-based methods, combined with the energy equation, have been used to obtain pressure-temperature-composition (P-T-x) relationships, which are the bubble and dew point temperatures. Simpler curvefits for P-T-x are also widely used to save computing time, suitable for design. However, fugacity and chemical potential-based methods need time-consuming iteration and curvefit models using polynomial equations have oscillatory behavior so that they are thermodynamically inconsistent in some regimes.

For dynamic analysis or accurate iteration in the solution of ammonia-water systems, a faster and smoother property model is necessary. A set of tabulated data with an interpolation method for P-T-x are chosen to overcome the problems from the above-mentioned relationships.

Process Models

Since terrestrial gas turbine systems gained widespread use, utilizing waste heat has been an important option for increasing the system efficiency. Steam turbines, absorption chillers or cogeneration can be integrated, depending on the local demands. Absorption chillers are appropriate when the ambient temperature is always high enough to run air conditioners or for the food processing or storage industry. A semi-closed microturbine system is combined with a double expansion VARS, which has three absorbers for cooling hot air or gas and making ice.

This system, however, has not yet been studied in the literature. In this research, the VARS unit has been conceptually modeled at multiple levels and analyzed. Thermodynamic models based on the First Law and on the Second Law have been developed in this work. To model two-phase, two-component flow in heat exchangers, the conventional moving boundary heat exchanger model has been extended to enforce species balances and phase equilibrium.

Organization of This Document

The literature for the relevant research areas is reviewed in Chapter 2. In Chapter 3, a study of thermodynamic properties of ammonia-water mixtures is presented, in which existing commonly-used mathematical models and an improved interpolation approach are introduced and compared with experimental data.

Chapters 4 to 6 address thermodynamic modeling of semi-closed cycle systems including both the VARS and HPRTE. A second law VARS model is first discussed in Chapter 4, and dynamic modeling of single effect systems for constant and variable ammonia mass fraction are presented in Chapters 5 and 6, respectively.

The overall results are summarized in Chapter 7 including recommendations for future research.

CHAPTER 2
LITERATURE REVIEW

This section discusses the literature on thermodynamic properties of ammonia-water mixtures, process models.

Thermodynamic Properties of Ammonia-Water Mixtures

Ammonia-water mixtures are considered as a highly effective working fluid for vapor absorption refrigeration operations, which take advantage of waste heat, and for other energy applications such as Kalina cycle power-generation plants [4] which can be driven by low-temperature heat sources. To design these systems for high-efficiency operation, it is important to have access to the thermodynamic properties of vapor-liquid mixtures of ammonia and water. In principle, it is of interest to ensure that these variables are thermodynamically and mathematically consistent. Furthermore, for system-design purposes it is also important to ensure that the mathematical models providing the physical data are easily programmable for numerical execution in a computing platform, and also that they are time-efficient [5] in terms of execution time.

The literature contains reports of ammonia-water mixture presented in terms of fundamental thermodynamic models, curve-fitting approximations, and tabulated data. Of particular interest is the work of Tillner-Roth et al. [6] who present a comprehensive set that includes most of the experimental data available. They include the data of Gillespie et al. [7], which has been used extensively by others to develop thermodynamic modeling equations because the set includes a very wide range of temperatures and compositions. Macriss et al. [8] present data measured at high ammonia concentrations.

Park et al. [9] present a thermodynamic model for the ammonia-water binary system based on equations of state for the Helmholtz free energy. This approach is capable of predicting mixture properties up to 20MPa and 650K, a temperature and pressure range where experiments are particularly challenging to carry out.

Another proposition based on the Helmholtz free energy is presented by Tillner-Roth et al. [10], a work that includes a large set of experimental data [6]. These authors use density as an input variable rather than the more common choice of using pressure, hence avoiding the need to include in the model separate equations for the pure components and for the mixture. Very accurate density data, therefore, are necessary for thermodynamic consistency. In addition to the Helmholtz free energy, various equations of state have been developed. For example, thermodynamic models based on the Gibbs free energy are also frequently encountered.

Bubble and dew-point temperature equations and P-T-x equations, developed as polynomial functions of pressure and ammonia concentration, can be found in El-Sayed et al. [11] and Pátek et al. [7]. Using such polynomial P-T-x equations, iteration processes can be removed to get equilibrium temperature. Even though thermodynamically inconsistent, a set of five equations provided by Pátek et al. is easy to program for modeling a VARS, which operate in a pressure range from 0.2 to 2MPa. Tabulated P-T-x data are also available. Bogart [12] presents various mixture data and tabulated ammonia mass fraction and enthalpy data for saturated liquid and vapor according to temperature up to a pressure of approximately 2MPa. Using polynomial P-T-x equations, rather than numerical iterations for phase equilibrium calculations by the fugacity or chemical-potential equality method, is more convenient for determining

mixture properties such as temperature, dryness, and physical state (superheated, sub-cooled, and equilibrium). Unlike any other tabulated data, Bogart provided temperature data for very high ammonia compositions.

Xu et al. [13] and Tamm [14] combined El-Sayed's P-T-x relations with the equations of state for the Gibbs free energy developed by Ibrahim et al. [15]. Ryu et al. [16] used Pátek's bubble and dew point temperatures with Ibrahim's enthalpy and density relations for their dynamic simulation of PoWER system, a highly efficient semi-closed gas turbine system [3].

Process Models

Lear and Sherif [3] patented a combined cooling and power cycle, which produces power, refrigeration, and water extraction, in addition to heat. Khan [17] studied several cases of operation modes and optimized the system. His steady-state HPRTE model is used for this research.

Adewusi et al. [18] compared a single-stage VARS with a double-stage VARS by second law based thermodynamic analysis. The entropy generation of each component and the total entropy generation of all the system components as well as the COPs of the VARSs are calculated from the thermodynamic properties of the working fluids at various operating conditions.

A lumped parameter model of VARS was established by Kim and Park [19]. The dynamic two-phase thermal-hydraulic characteristics are investigated during its start-up operation period, which is most challenging to understand and model. In addition to their literature, MacArthur [20] gave clear method of discretized dynamic heat exchanger model for compressible two-phase flow.

He et al. [21] and Jensen et al. [22] modeled vapor compression refrigeration systems using moving boundary heat exchanger model to meet reasonable two-phase physics and control application at once. They used Wedekind's [23] time-invariant mean void fraction idea for two-phase dynamics and Zivi's [24] void fraction model.

CHAPTER 3
THERMODYNAMIC PROPERTIES OF AMMONIA-WATER MIXTURES

In this chapter, we show that the polynomial equations of El-Sayed et al. and Pátek et al. and a non-monotonic behavior at near-pure compositions, an effect that becomes particularly pronounced at near pure-ammonia compositions and caused by the nature of polynomial equations (Runge's phenomenon) rather than as a consequence of the thermodynamics. In addition, frequently used P-T-x relationships and Bogart's P-T-x tabulated data are compared and used with Ibrahim's equations, and the best method for developing controller using memory chips for VARSs and less mathematical error will be discussed. The data in Tillner-Roth et al. [6] and Gillespie et al. [7] is used as a source of comparison in this work.

Gibbs Free Energy for Ammonia-Water Mixture

Schulz [26] made two separate equations for the liquid and the vapor phases of the ammonia-water mixtures. His model is limited to a maximum pressure of 2.5MPa, which is extended by Ziegler et al. [26] to 5MPa and 500K. Ibrahim and Klein [15] modified the coefficients for the liquid Gibbs excess energy to include the experimental data reported by Gillespie et al. [7]. By doing so, their correlations cover vapor-liquid equilibrium pressures of 11MPa and temperature of 600K.

Gibbs Free Energy for Pure Components

The equations of Gibbs free energy for the pure ammonia or pure water are as shown below for the liquid and vapor states.

Liquid phase:

$$G_r^L = h_{r,o}^L - T_r s_{r,o}^L + B_1\left(T_r - T_{r,o}\right) + \frac{B_2}{2}\left(T_r^2 - T_{r,o}^2\right) + \frac{B_3}{3}\left(T_r^3 - T_{r,o}^3\right)$$
$$- B_1 T_r \ln\left(\frac{T_r}{T_{r,o}}\right) - B_2 T_r\left(T_r - T_{r,o}\right) - \frac{B_3}{2} T_r\left(T_r^2 - T_{r,o}^2\right) \qquad (3\text{-}1)$$
$$+\left(A_1 + A_3 T_r + A_4 T_r^2\right)\left(P_r - P_{r,o}\right) + \frac{A_2}{2}\left(P_r^2 - P_{r,o}^2\right)$$

Gas Phase:

$$G_r^G = h_{r,o}^G - T_r s_{r,o}^G + D_1\left(T_r - T_{r,o}\right) + \frac{D_2}{2}\left(T_r^2 - T_{r,o}^2\right) + \frac{D_3}{3}\left(T_r^3 - T_{r,o}^3\right)$$
$$- D_1 T_r \ln\left(\frac{T_r}{T_{r,o}}\right) - D_2 T_r\left(T_r - T_{r,o}\right) - \frac{D_3}{2} T_r\left(T_r^2 - T_{r,o}^2\right)$$
$$+ T_r \ln\left(\frac{P_r}{P_{r,o}}\right) + C_1\left(P_r - P_{r,o}\right) + C_2\left(\frac{P_r}{T_r^3} - 4\frac{P_{r,o}}{T_r^3} + 3P_{r,o}\frac{T_r}{T_{r,o}^4}\right) \qquad (3\text{-}2)$$
$$+ C_3\left(\frac{P_r}{T_r^{11}} - 12\frac{P_{r,o}}{T_{r,o}^{11}} + 11P_{r,o}\frac{T_r}{T_{r,o}^{12}}\right) + \frac{C_4}{3}\left(\frac{P_r^3}{T_r^{11}} - 12\frac{P_{r,o}^3}{T_{r,o}^{11}} + 11P_{r,o}^3\frac{T_r}{T_{r,o}^{12}}\right)$$

where the reduced thermodynamic properties are defined as follows:

$$T_r = \frac{T}{T_B} \qquad (3\text{-}3)$$

$$P_r = \frac{P}{P_B} \qquad (3\text{-}4)$$

$$G_r = \frac{\tilde{G}}{RT_B} \qquad (3\text{-}5)$$

The reference values for the reduced properties are R = *8.314* kJ/kmole·K, T_B = 100 K, and P_B = 10 bar. Table 3.1 reports the numerical values of the coefficients appearing in equations (3-1) and (3-2) for the cases of pure ammonia and pure water.

The Gibbs free energy equations (3-1)-(3-5) can in turn be used to calculate the molar specific enthalpy, entropy, and volume of the pure components in terms of reduced variables through the following suite of equations.

$$\tilde{h} = -RT_B T_r^2 \left[\frac{\partial}{\partial T_r} \left(\frac{G_r}{T_r} \right) \right]_{P_r} \tag{3-6}$$

$$\tilde{s} = -R \left[\frac{\partial G_r}{\partial T_r} \right]_{P_r} \tag{3-7}$$

$$\tilde{v} = \frac{RT_B}{P_B} \left[\frac{\partial G_r}{\partial P_r} \right]_{T_r} \tag{3-8}$$

Vapor and Liquid Gibbs Free Energy for the Binary Mixtures

Vapor phase ammonia-water mixtures are assumed to be ideal solutions. Hence, the following thermodynamic properties can be obtained through simple relationships involving only the gas-composition of ammonia:

$$\tilde{h}_m^G = \tilde{y}\tilde{h}_a^G + (1-\tilde{y})\tilde{h}_w^G \tag{3-9}$$

$$\tilde{s}_m^G = \tilde{y}\tilde{s}_a^G + (1-\tilde{y})\tilde{s}_w^G + \tilde{s}^{mix} \tag{3-10}$$

$$\tilde{s}^{mix} = -R\left\{\tilde{x}\ln(\tilde{x}) + (1-\tilde{x})\ln(1-\tilde{x})\right\} \tag{3-11}$$

$$\tilde{v}_m^G = \tilde{y}\tilde{v}_a^G + (1-\tilde{y})\tilde{v}_w^G \tag{3-12}$$

For liquid mixtures, the Gibbs excess energy

$$G_r^E = (1-\tilde{x})\left\{F_1 + F_2(2\tilde{x}-1) + F_3(2\tilde{x}-1)^2\right\} \tag{3-13}$$

must be included, where

$$F_1 = E_1 + E_2 P_r + \left(E_3 + E_4 P_r\right) T_r + \frac{E_5}{T_r} + \frac{E_6}{T_r^2} \tag{3-14}$$

$$F_2 = E_7 + E_8 P_r + \left(E_9 + E_{10} P_r\right) T_r + \frac{E_{11}}{T_r} + \frac{E_{12}}{T_r^2} \tag{3-15}$$

$$F_3 = E_{13} + E_{14} P_r + \frac{E_{15}}{T_r} + \frac{E_{16}}{T_r^2} \tag{3-16}$$

Then, the liquid-mixture thermodynamic properties can be postulated as follows:

$$\tilde{h}_m^L = \tilde{x}\tilde{h}_a^L + \left(1-\tilde{x}\right)\tilde{h}_w^L + \tilde{h}^E \tag{3-17}$$

$$\tilde{s}_m^L = \tilde{x}\tilde{s}_a^L + \left(1-\tilde{x}\right)\tilde{s}_w^L + \tilde{s}^E + \tilde{s}^{mix} \tag{3-18}$$

$$\tilde{v}_m^L = \tilde{x}\tilde{v}_a^L + \left(1-\tilde{x}\right)\tilde{v}_w^L + \tilde{v}^E \tag{3-19}$$

Pressure-Temperature-Composition Properties of the Binary Mixtures

Pressure-temperature-composition (P-T-x) relationships of a vapor-liquid equilibrium state for the ammonia-water binary mixture can be obtained by solving a system of thermodynamic equations that imposes equality of pressures, temperatures and component fugacities across the two phases. Such approach requires an iterative numerical solution method, as carried out for example by Ibrahim et al. [15] and Park and Sonntag [9]. The latter reference also discusses an approach for calculating equilibrium temperatures using flow charts.

The calculation of thermodynamic equilibrium properties via numerical methods involves a significant computational cost. Hence, once the calculations are carried out, it is customary to report selected values in a table for future access, or alternatively, to approximate the results through a mathematical correlation often of polynomial form.

The user then needs to compute the values of a polynomial expression or carry out an interpolation in a table to obtain specific thermodynamic values, such as the dew point at a given pressure and mixture composition.

Using a curve-fitting approach, El-Sayed and Tribus developed the following bubble (T_b) and dew point (T_d) temperature polynomial correlations:

$$T_b = T_c - \sum_{i=1}^{7}\left(c_i + \sum_{j=1}^{10} c_{ij}\tilde{x}^j \right)\left(\ln\left(\frac{P_c}{P}\right)\right)^i \tag{3-20}$$

$$T_d = T_c - \sum_{i=1}^{6}\left(a_i + \sum_{j=1}^{4} A_{ij}\left(\ln\left(1.0001-\tilde{y}\right)\right)^j \right)\left(\ln\left(\frac{P_c}{P}\right)\right)^i \tag{3-21}$$

where

$$T_c = T_{cw} - \sum_{i=1}^{4} a_i \tilde{x}^i \tag{3-22}$$

$$P_c = P_{cw}\exp\left(\sum_{i=1}^{8} b_i \tilde{x}^i \right) \tag{3-23}$$

where P is the mixture pressure, $\tilde{x}$ and $\tilde{y}$ are ammonia mole fraction of saturated liquid and vapor, respectively, T_{cw} and P_{cw} are critical temperature and pressure for water, and T_c and P_c are critical temperature and pressure of ammonia-water mixtures.

Pátek and Klomfar propose equations for describing the thermodynamic properties of the ammonia-water binary system with pressure ranges up to 2MPa. A particularly useful equation calculates the vapor composition as a function of the pressure and the liquid composition without requiring an iterative computational method. Even though the vapor enthalpy equation is not available for superheated region and bubble and dew

point temperatures and vapor composition equations are not thermodynamically consistent, the equation set is in widespread use in industry [27] where rapid calculations are needed. The bubble point and dew point temperature correlations proposed by these authors are the following equations:

$$T_b = T_o \sum_{i=1}^{14} a_i \left(1-\tilde{x}\right)^{m_i} \left(\ln\left(\frac{P_o}{P}\right)\right)^{n_i} \tag{3-24}$$

$$T_d = T_o \sum_{i=1}^{17} a_i \left(1-\tilde{y}\right)^{m_i/4} \left(\ln\left(\frac{P_o}{P}\right)\right)^{n_i} \tag{3-25}$$

where T_o and P_o are reference temperature and pressure.

Bogart [12] tabulated calculated data. Unlike other published data, his data are available at very high ammonia composition as function of temperature at given pressures. The data is a compilation of various experimental results. Obtaining thermodynamic information from Bogart's tables reduces to an exercise in numerical interpolation, for example, using cubic polynomials, which entails a relatively low computational cost.

Results and Discussion

In this study, three different P-T-x relations with Ibrahim's Gibbs free energy method are compared with most used experimental and smoothed data.

Comparison with Gillespie's Data

Figures 3-1 and 3-2 show that Bogart tabulated data with cubic interpolation best follow the experimental data from Gillespie for saturated liquid while Pátek's equations are generally better for saturated vapor. This trend is shown for a wide range of temperatures. However, the differences between three P-T-x relations and Gillespie's

experimental data are not significantly large except the El-Sayed's at almost pure ammonia compositions.

Pátek and Klomfar have their own enthalpy equations for saturated liquid and saturated vapor. Their saturated enthalpy equations with temperature equations show very good agreements at most regions. However, their saturated vapor model is not available for the superheated region. So, Ibrahim's Gibbs free energy method will be used to get enthalpies of the mixtures for the comparison.

Comparison with IGT Experimental Data

Macriss et al. [8] provided both experimental and calculated data in their research bulletin for the Institute of Gas Technology (IGT). Their experimental data are compared with three P-T-x methods and explained, mainly while their smoothed data are used for figures to compare.

Saturated liquid phase

For low ammonia concentrations (0.0475 ~ 0.383 of ammonia mass fraction, IGT Table C-3 ~ C-7 [8]), three P-T-x methods show good agreement overall for liquid enthalpy (Figure 3-3). Bogart's and Pátek's are somewhat better at lower pressure and El-Sayed's relations are better at higher pressure (from about 1500kPa).

Saturated vapor phase

The behaviors of saturated vapor temperatures and enthalpies of three methods show the same trends at high pressure ranges (1378.95 (200), 1654.74 (240), and 2068.43 (300) kPa (psia), IGT Table A-1 and C-2 [8])) for a refrigeration unit and at high ammonia mass-fraction regions. The percentage errors that Pátek's and El-Sayed's temperature equations make at those regions are always lower than those of Bogart's method. In particular, Pátek's temperatures best follow the IGT experimental data.

Bogart's temperatures with cubic interpolation are about 5 K (approximately 1.5 % error) higher than the IGT temperature data, while El-Sayed's temperature drops fast and fluctuates at near pure ammonia compositions.

With Ibrahim's saturated vapor enthalpy relation, Pátek's, El-Sayed's, and Bogart's show good agreement at lower, middle, and higher ammonia compositions, respectively at high ammonia concentration. The percentage errors are less than approximately 1 percent.

Comparison with IGT Smoothed Data

IGT smoothed (calculated) data for 344.74kPa, 1034.21kPa, and 1723.69kPa (IGT Table C-8, 9, 11, 13, and 14 [8]) were compared with three different P-T-x models. As shown at Figure 3-6, all of three models follow well the IGT data. Pátek's temperature is somewhat more precise at higher pressure, and Bogart's temperature predictions are close to the experimental data at lower pressure. As a resolution of Figure 3-7, three P-T-x methods show almost identical enthalpy results.

When near pure component regions are inspected in closer detail, two significant problems are exposed. At lower pressure (344.74kPa = 50psia), the vapor temperatures are lower than liquid temperature for both El-Sayed's and Pátek's correlations, even though it is difficult to visually observe the effect even at the high resolution of Figure 3-8. As the pressure increases, El-Sayed's dew point temperature fluctuates and never meets the bubble point temperature in the limit of pure ammonia.

A temperature discrepancy between dew and bubble point temperatures of pure components is also observed in Pátek's correlation. At very high ammonia concentration, dew point temperatures become lower than bubble point temperatures, which is physically impossible. Bogart's data with interpolation, however, do not show

these problems. The oscillations appear at enthalpy-composition Figures 3-9 and 3-10, also.

Temperature Discrepancies for Pure Components

Figure 3-11 shows temperature differences between dew and bubble point temperatures at pure water and pure ammonia. Generally, the higher pressure the more discrepancy. Although at very high ammonia concentrations (more than 99.9%), pure water and pure ammonia are not of high interest in vapor-absorption refrigeration processes, when continuous dynamic calculations and high accuracy iteration calculations are necessary, the small fluctuations at very high ammonia compositions and temperature discrepancies may compromise the validity of the resulting calculations.

For example, consider the saturated vapor composition at a given temperature and liquid composition at the equilibrium state. If the P-T-x correlation of El-Sayed and Tribus is used as shown in Figure 3-12, it is concluded that the saturated vapor composition will be higher than 100 percent when the mixture (pure ammonia) starts to boil. Because the dew point temperature fluctuates, three different vapor compositions exist for one liquid composition. To avoid these problems, it is necessary for the maximum vapor composition to be limited by the minimum dew point temperature. At 1700kPa, concentrations cannot be higher than 0.99965 of saturated vapor composition and 0.98938 of saturated liquid composition.

When the P-T-x correlations by Pátek and Klomfar is deployed at high pressures, it is important for the user to keep in mind that for a certain composition range the predicted dew point is lower than the bubble point. Therefore, saturated vapor composition by given liquid composition and temperature cannot be obtained beyond

the temperature matching point as shown at Figure 3-13. El-Sayed's also has same problem at lower pressure.

Summary

Various vapor liquid equilibrium methods were studied in this paper. To avoid time-consuming iterative numerical calculation, three P-T-x methods were compared. Bubble and dew point temperature equations are easy to program and show very good agreement with experimental data. However, there are regions where the predicted thermodynamic variables exhibit local oscillatory patterns characteristic (Runge's phenomenon) because of the nature of polynomial equations. This thermodynamically inconsistent behavior leads to erroneous predictions, including the observations that the predicted dew and bubble point temperatures for pure components do not match, that dew point temperatures are lower than bubble point temperatures, and that there are several different compositions for one given dew point temperature. Under these conditions, the use of polynomial correlations of high order may lead to erroneous thermodynamic calculations.

The alternative of using well developed tabulated data with interpolation can ensure that the problems discussed in the previous paragraph are avoided. Given the memory capabilities of modern computing platforms, it is possible to store and access very large tables at very little cost of execution time. However, to achieve this objective, more accurate wide range experimental data are necessary. Thermodynamic consistency tests might also be useful and necessary.

Table 3-1. Coefficients for equations (3-1) and (3-2) for pure water and ammonia

Coefficient	Ammonia	Water
A_1	3.971423e-2	2.748796e-2
A_2	-1.790557e-5	-1.016665e-5
A_3	-1.308905e-2	-4.452025e-3
A_4	3.752836e-3	8.389246e-4
B_1	1.634519e1	1.214557e1
B_2	-6.508119	-1.898065
B_3	1.448937	2.911966e-1
C_1	-1.049377e-2	2.136131e-2
C_2	-8.288224	-3.169291e1
C_3	-6.647257e2	-4.634611e4
C_4	-3.045352e3	0
D_1	3.673647	4.01917
D_2	9.989629e-2	-5.17555e-2
D_3	3.617622e-2	1.951939e-2
$h^L_{r,o}$	4.878573	21.821141
$h^G_{r,o}$	26.468879	60.965058
$s^L_{r,o}$	1.644773	5.733498
$s^G_{r,o}$	8.339026	13.45343
$T_{r,o}$	3.2252	5.0705
$P_{r,o}$	2	3

Table 3-2. Coefficient for Gibbs excess energy function

E_1	-41.733398	E_5	63.608967	E_9	0.387983	E_{13}	-3.553627
E_2	0.024140	E_6	-62.490768	E_{10}	-0.004772	E_{14}	0.000904
E_3	6.702285	E_7	1.761064	E_{11}	-4.648107	E_{15}	24.361723
E_4	-0.011475	E_8	0.008626	E_{12}	0.836376	E_{16}	-20.736547

Table 3-3. Comparison of dew point temperatures

P [kPa]	x	IGT	Bogart	% error	El-Say	% error	Pátek	% error
1530.6	0.9953	333.872	339.256	-1.6126	331.804	0.6195	333.333	0.1615
1916.7	0.9953	339.706	345.104	-1.5892	338.256	0.4267	338.877	0.2439
1551.3	0.9907	344.761	348.796	-1.1703	343.806	0.2770	342.689	0.6012
1868.4	0.9907	348.761	353.607	-1.3896	348.737	0.0070	347.232	0.4385
1634.1	0.9824	355.039	360.086	-1.4216	357.682	-0.7444	354.332	0.1991
1903.0	0.9824	360.206	364.322	-1.1428	361.633	-0.3962	358.141	0.5732
1544.4	0.9641	366.928	371.682	-1.2956	371.013	-1.1132	366.768	0.0434

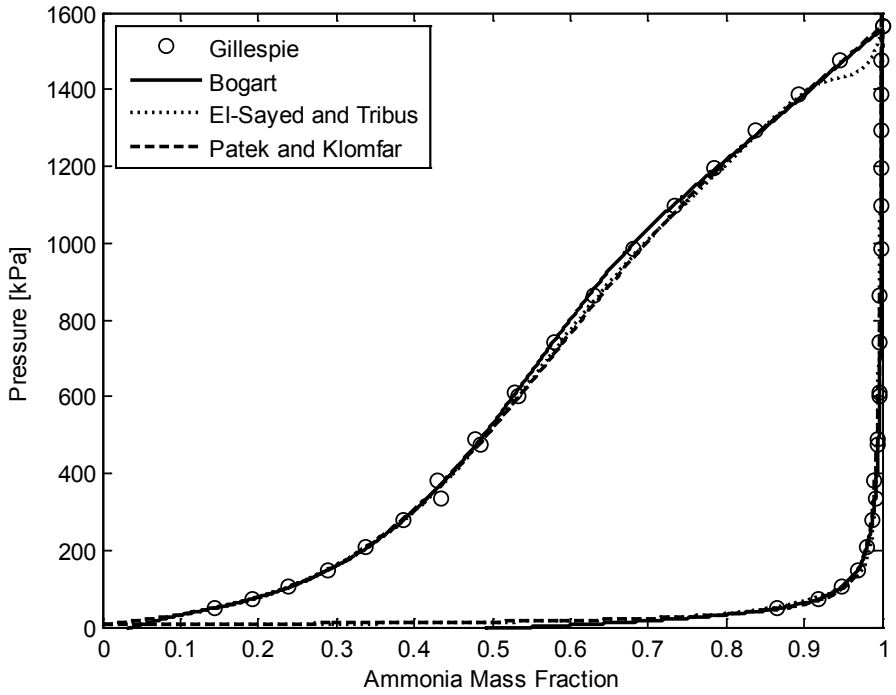

Figure 3-1. Comparison P-T-x relations with Gillespie's data at 313.15K

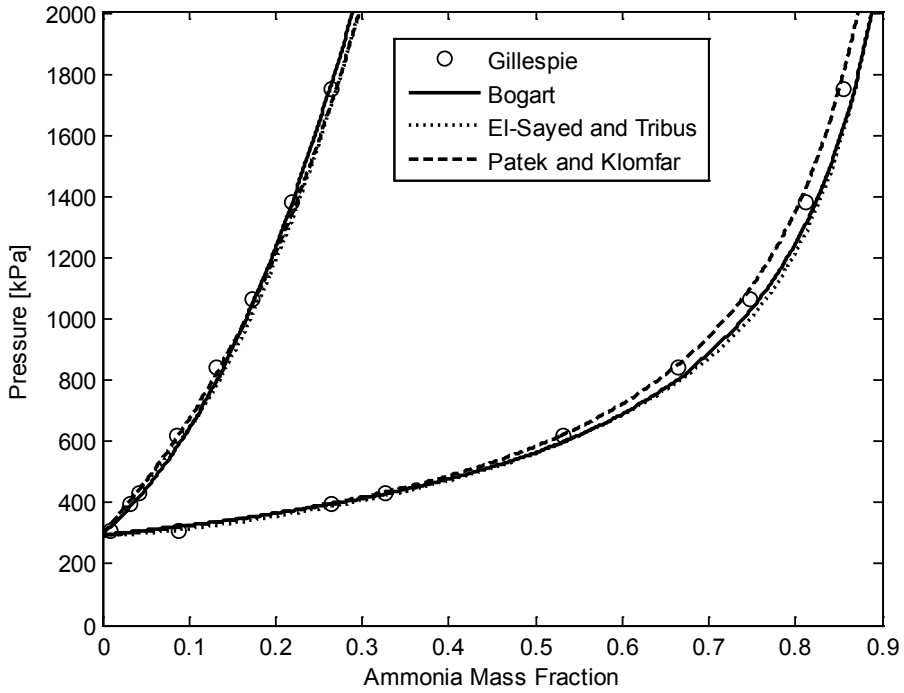

Figure 3-2. Comparison P-T-x relations with Gillespie's data at 405.9K

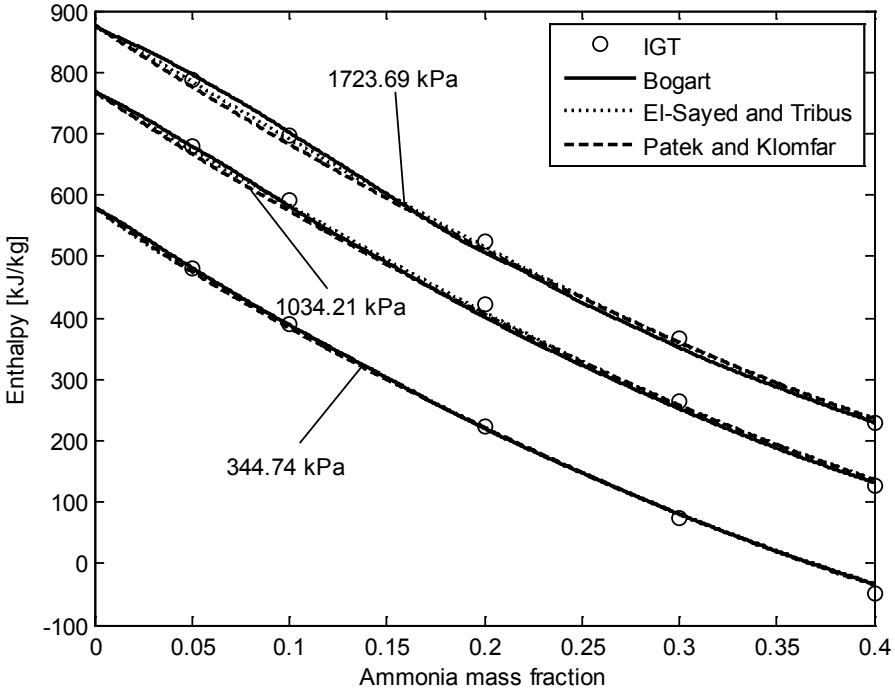

Figure 3-3. Comparison for saturated liquid enthalpy

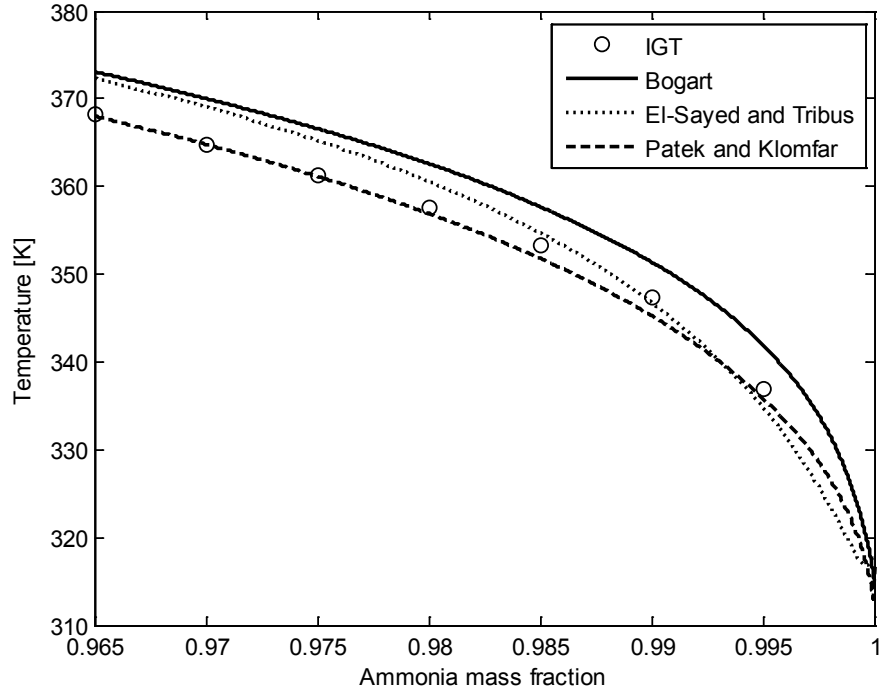

Figure 3-4. Comparison with IGT data for saturated vapor temperature at 1654.74kPa

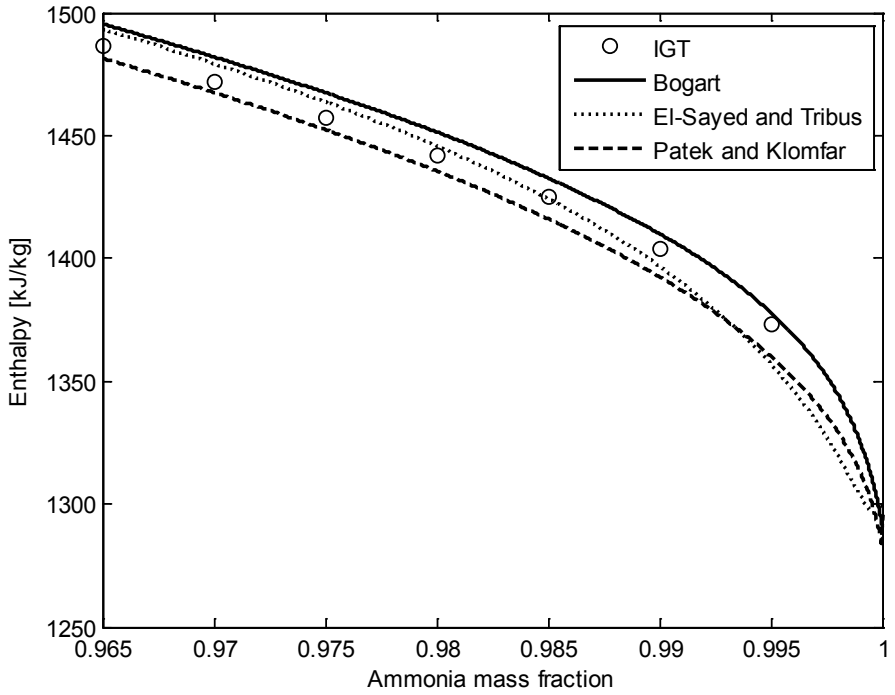

Figure 3-5. Comparison with IGT data for saturated vapor enthalpy at 1654.74kPa

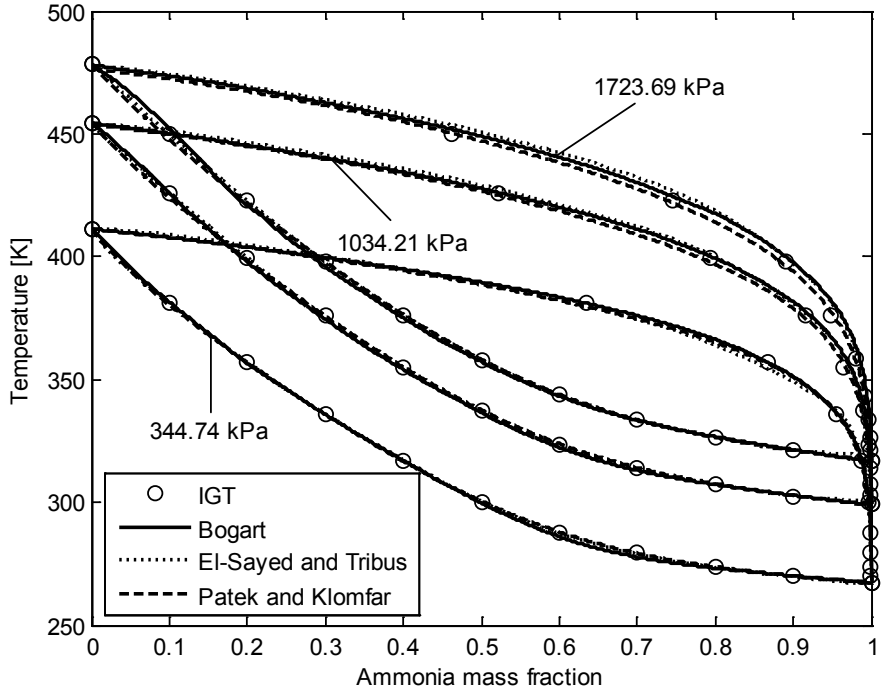

Figure 3-6. Comparison with IGT data for saturated vapor and liquid temperature

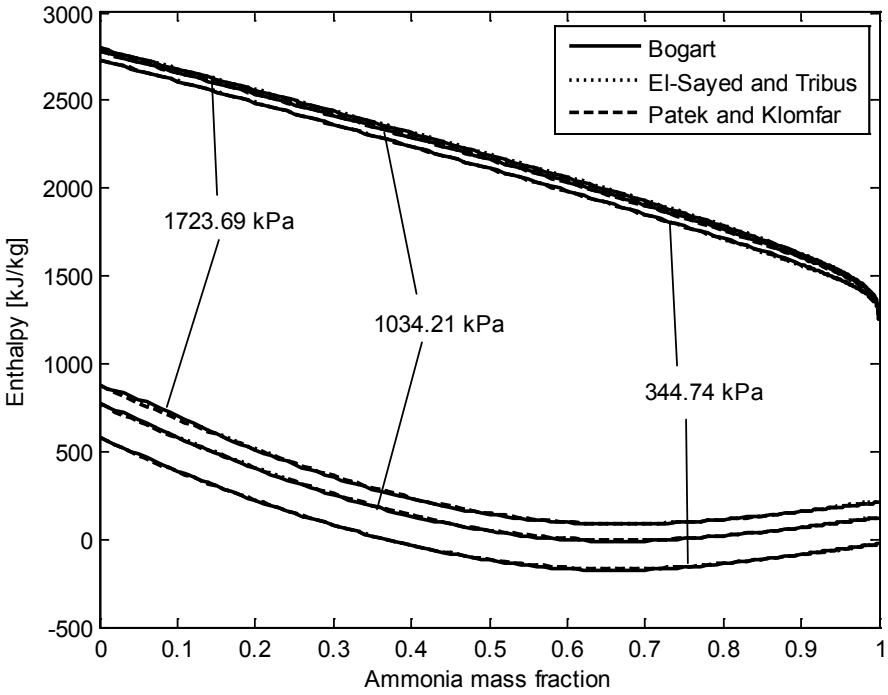

Figure 3-7. Comparison with IGT data for saturated vapor and liquid enthalpy

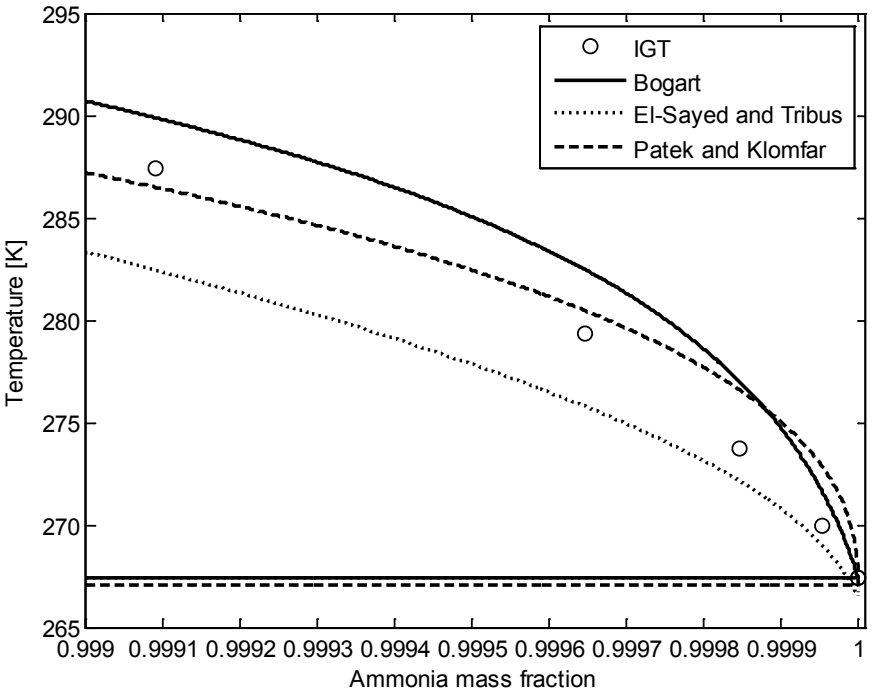

Figure 3-8. Saturated temperatures for high ammonia mass fraction at 344.74kPa

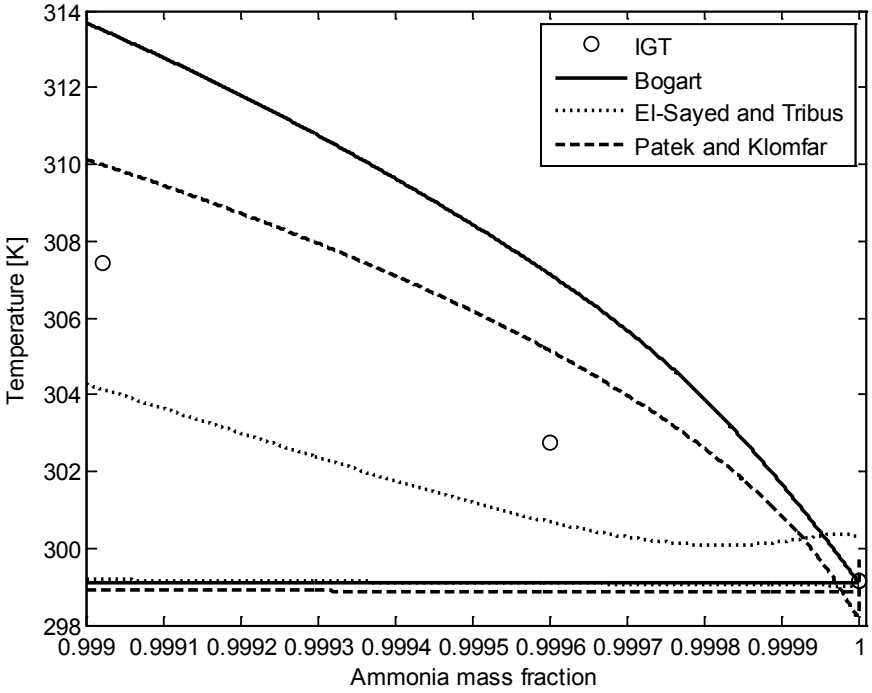

Figure 3-9. Saturated temperatures for high ammonia mass fraction at 1034.21kPa

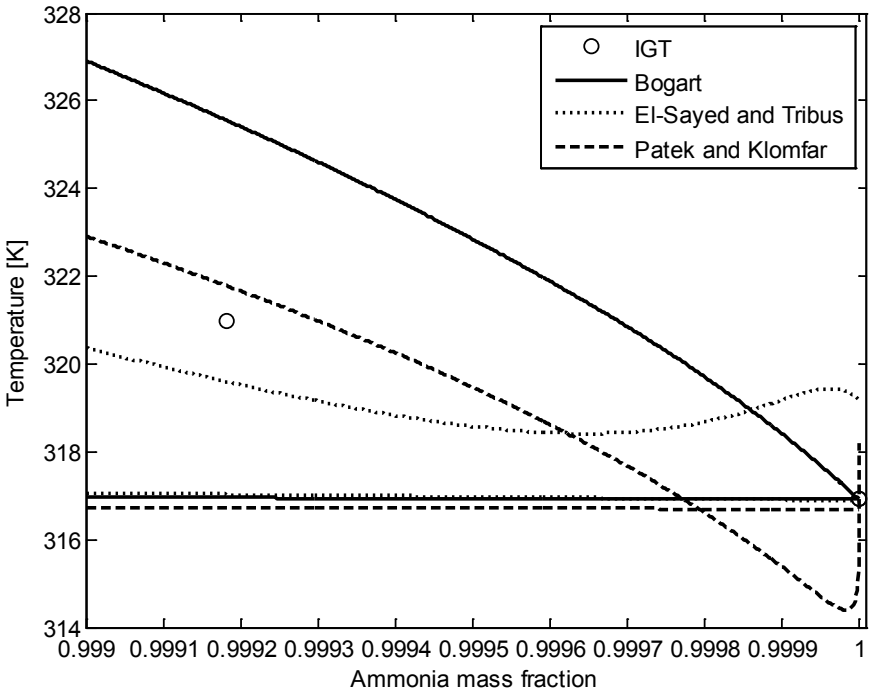

Figure 3-10. Saturated temperatures for high ammonia mass fraction at 1723.69kPa

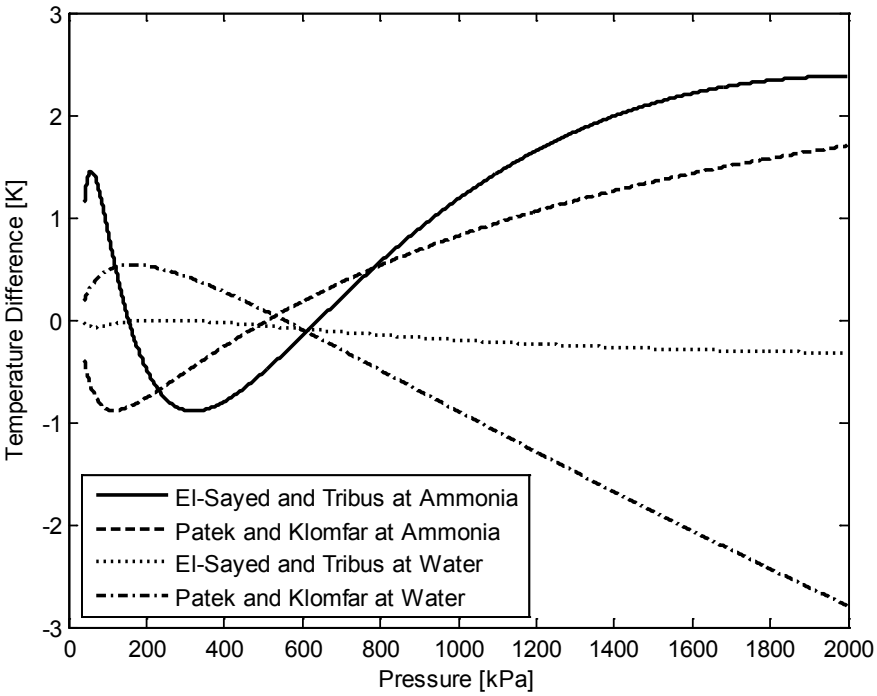

Figure 3-11. Temperature discrepancies at pure components

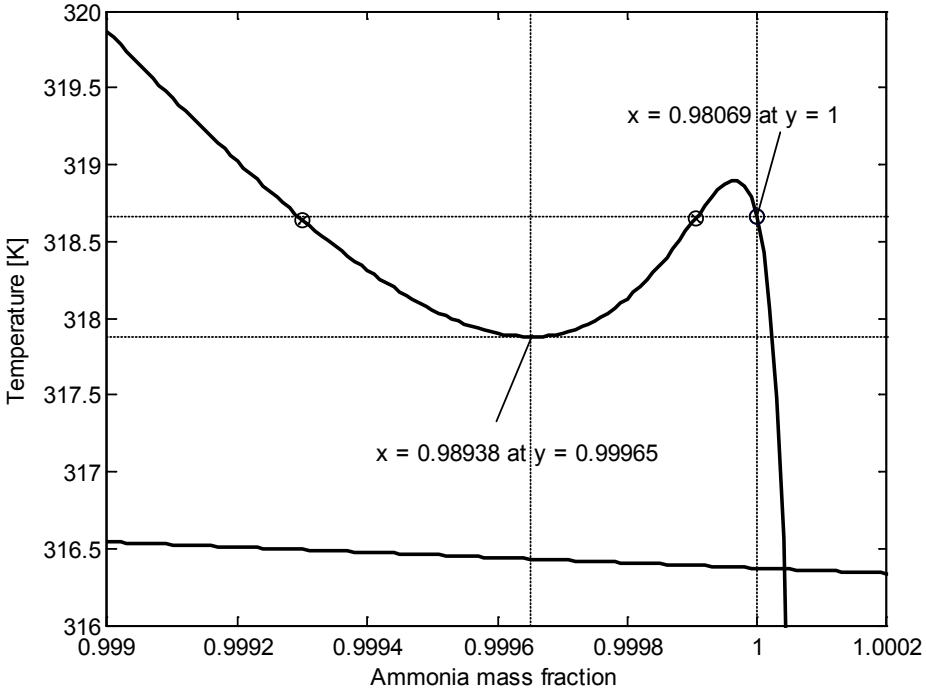

Figure 3-12. Discrepancy and oscillation of El-Sayed and Tribus's P-T-x

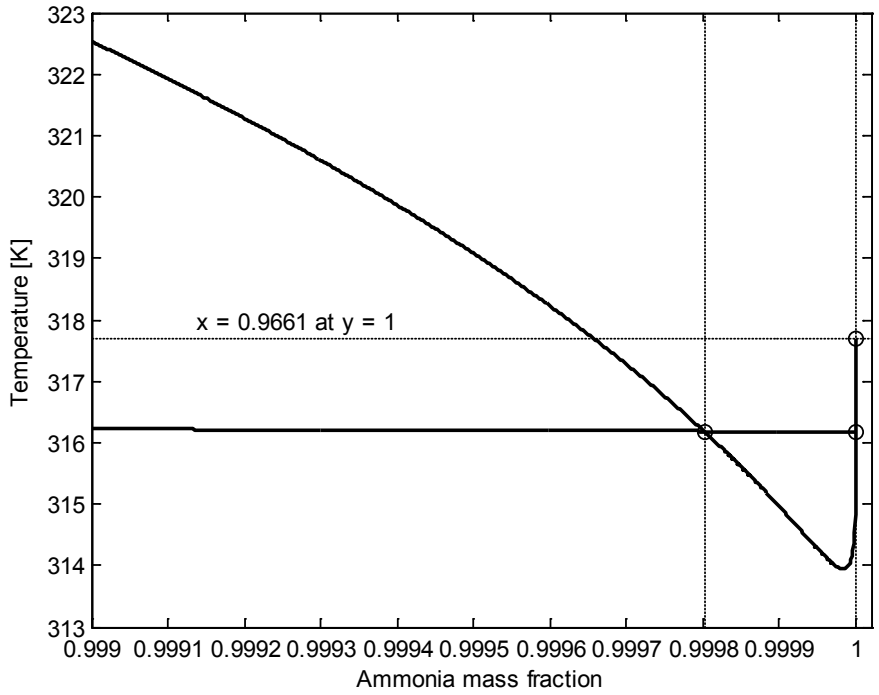

Figure 3-13. Discrepancy and oscillation of Pátek and Klomfar's P-T-x

CHAPTER 4
SYSTEM DESIGN OF A REFRIGERATION SYSTEM

In this research, the focus is on a variant of the original PoWER system, one in which the VARS is designed to produce refrigeration at two temperatures. The primary evaporator would operate above freezing, cooling the gas path as well as a local air conditioning load. During the summer peak electrical demand in hot climates, the nighttime excess refrigerating capacity would be used to produce ice in a closed thermal storage subsystem. The ice would then be available during the afternoon demand peak to provide supplemental air conditioning, thus providing a demand-side management benefit.

Second Law Analysis Using Constant Internal Temperatures

For preliminary design of double expansion, single effect VARS using the Second Law approach, temperatures for main components need to be selected. Most evaporators show a single temperature for their refrigerant in the refrigeration side. Therefore, constant internal temperatures at three evaporators have been used to obtain the Second Law efficiency.

Thermodynamic Model

Figures 4-1 and 4-2 show the PoWER system, split for convenience into a gas turbine flow path (Figure 4-1) and the coupled vapor absorption refrigeration system (Figure 4-2). The gas turbine operates on a semi-closed cycle, in that a substantial fraction of the flow from the hot side recuperator exit is mixed with the fresh air leaving the low-pressure compressor. The majority of the waste heat of the engine is removed from the mixture at an intermediate pressure (typically several atmospheres), allowing much smaller heat exchangers than conventional ones at the exhaust. The captured

waste heat powers the VARS in this implementation. The VARS cooling capacity is used in part to chill the gas at the entrance to the high-pressure compressor, thereby significantly increasing the efficiency.

The gas turbine flow path model has been reported by Lear and Laganelli [28]. The component models are conventional for such cycle calculations except that the extraction of water condensate from the evaporator is taken into account, along with the influence of water concentration on thermodynamic properties. Results from Khan et al. [29] were utilized in combination with a conceptual refrigeration system optimized for this application. Since the current analysis is in the conceptual design stage, it would be incorrect to specify the design of the VARS. Instead, we have chosen to calculate the ideal COP for a reversible refrigeration system, then parameterize the fraction of that ideal to be achieved by an actual system. Stated differently, this analysis utilizes a Second Law efficiency to allow calculation of the VARS performance.

For the Second Law analysis, it is recognized that a control volume enclosing the two-coolant temperature VARS is equivalent to one enclosing two conventional VARS systems, one for each low-temperature reservoir as shown at Figure 4-3.

For each conventional VARS unit, the First and Second Laws may be written as

$$\dot{Q}_{in} = \sum_{i=1}^{3} \left(\dot{Q}_{Gen,i} + \dot{Q}_{E,i} \right) + \dot{W}_p \tag{4-1}$$

$$\dot{Q}_{out} = \sum_{i=1}^{3} \dot{Q}_{0,i} = \dot{Q}_{in} \tag{4-2}$$

$$\dot{P}_{S,i} = -\frac{\dot{Q}_{Gen,i}}{T_{Gen}} - \frac{\dot{Q}_{E,i}}{T_{E,i}} + \frac{\dot{Q}_{o,i}}{T_o} \geq 0 \tag{4-3}$$

Combining Equations (4-1), (4-2), and (4-3) with negligible $\dot{W}_p$

$$\dot{Q}_{E,i}\frac{T_{E,i}-T_o}{T_{E,i}} \geq \dot{Q}_{Gen,i}\frac{T_o-T_{Gen}}{T_{Gen}} \tag{4-4}$$

For the reversible case, the entropy production is zero, so the equality in Equation (4-4) is appropriate, yielding

$$COP_{Carnot1,2} = \frac{\dot{Q}_{E1,2}}{\dot{Q}_{Gen1,2}} = \frac{T_E(T_{Gen}-T_O)}{T_{Gen}(T_O-T_E)} \tag{4-5}$$

$$COP_{Carnot3} = \frac{\dot{Q}_{E,3}}{\dot{Q}_{Gen,3}} = \frac{T_{ICE}(T_{Gen}-T_O)}{T_{Gen}(T_O-T_{ICE})} \tag{4-6}$$

where $T_{E,i}$ is T_E, for Evaporators 1 and 2, and T_{ICE} is the temperature for Evaporator 3. For each evaporator, the refrigeration efficiency is specified as a Second Law-based efficiency parameter. Using the conventional COP definition,

$$COP_i = \frac{\dot{Q}_{E,i}}{\dot{Q}_{Gen,i}} \tag{4-7}$$

enables solving for the COP values, given the Carnot COP values and refrigeration efficiency, from

$$\eta_{R,i} = \frac{COP_i}{COP_{Carnot,i}} \tag{4-8}$$

Solution Method

A computer code described previously by Khan et al. [29] was written in FORTRAN to simulate the performance of the PoWER system, using traditional cycle analysis methods. The code is capable of calculating engine performance (network output, water extraction flow rate and thermal efficiency) as a function of the input parameters (turbine inlet temperature, recuperator inlet temperature, generator temperature, evaporator exit temperature, low-pressure compressor ratio, recirculation ratio, turbo machinery efficiencies, heat exchanger effectiveness, equivalence ratio, fuel type, and pressure drops).

A wide range of VARS technologies exist, typically trading off between complexity and COP. The intent of the current paper is to quantify the potential for ice-making and load leveling, not to analyze one particular design choice. Therefore, the VARS was studied by a Second Law analysis, using as input the results of the HPRTE code. The refrigeration efficiency is taken as a parameter to represent the maturity of the VARS technology chosen.

The temperatures used for determining the Carnot coefficient of performance, COP_{Carnot}, in this research are tabulated in Table 4-1, all temperatures are either results from the HPRTE code or the specified environmental temperature. The variation of the outdoor temperature is shown in Figure 4-3, and the values of generator and evaporator temperatures are taken from Goswami et al. [30]. T_{ICE} was chosen by referring to the ASHRAE Handbook [31]. Figure 4-4 shows the maximum air condition load at 4 PM, while Figure 4-5 shows the maximum outdoor air temperature at 3 PM [32]. This phase lag is because of thermal storage of the building, the movement of heat sources such as people, and time-dependent use of lights in the building [33]. Figure 4-6 shows the

hourly maximum COPs (or COP_{Carnot}) calculated by using the variation of outdoor temperature, T_o as shown Figure 4-3.

The cycle input and output parameters – which will be used for a multistage VARS study – are given in Table 4-1. The low-pressure compressor pressure ratio was based on prior studies [34] and is chosen for maximum thermal efficiency and reasonable water extraction. The inlet temperature of the gaseous mixture entering the high-pressure compressor is constant and is equal to 10°C, which is a practical limit taking into account the fact that Evaporator 1 temperature is fixed at 5°C. The turbo machinery polytrophic efficiencies were based on state-of-the-art values as reported by Mattingly [35] and Carcasci and Facchini [36]. The pressure drops in the recuperator and the combustor are similar to the values reported by Fiaschi et al. [37]. The equivalence ratio Φ was set to 0.9 based on previous studies with the HPRTE as given in Boza et al. [38]. It was assumed to be equal to 0.9, because it was expected that at least 10% excess air was necessary to ensure complete combustion [34]. The gas entering the combustor is diluted with excess air and the recirculated combustion products, to lower the adiabatic flame temperature towards the specified turbine inlet temperature (This provides benefits in emissions as well as combustion efficiency).

The effectiveness of the recuperator is assumed to be 0.85 and the recuperator inlet temperature (on the hot side) is assumed to be 800°C. These are conservative values based on existing commercial equipment [39]. Turbine blade cooling is considered for higher turbine inlet temperatures, and the model for turbine blade cooling used is similar to that given by Massardo et al. [40]. This model gives a relation between the turbine inlet temperature and the percentage of main air flow used for turbine blade

cooling. Note that the intent here is to model a future system with advanced cooling and recuperator technology. This is the conservative choice from the standpoint of cooling and icemaking provided by the VARS, since the high firing and recuperator temperatures lead to high gas turbine efficiency, hence reduced waste heat for driving the VARS. That in turn limits the icemaking potential and the load leveling capability due to thermal energy storage. This conservative approach is chosen in order to balance the simplifications made in the thermal energy storage system, primarily the omission of parasitic power requirements and heat leaks.

Results and Discussion

The computer code of the HPRTE cycle and the second law approach of the multistage VARS were used to calculate hourly air conditioning and ice-making capacities. Given HPRTE data, which are calculated by Khan's model [30], together with the outdoor air temperature, air conditioning hourly required load, and refrigeration efficiency, the COP_{Carnot}, COPs, heat supplied to the generator, and heat removed from the surroundings for all three evaporators are calculated below and illuminated by various plots. In addition to those results, the amount of ice produced daily and hourly, and the air conditioned floor area are shown. These values were determined for six cases, varying the ratio of the heat added from the air conditioner (Evaporator 2), and ice maker (Evaporator 3) to the heat added at Evaporator 1 for two different refrigeration efficiencies,

$$R_{evp} = \frac{\dot{Q}_{E,2} + \dot{Q}_{E,3}}{\dot{Q}_{E,1}} \tag{4-9}$$

To produce additional heat to run Evaporators 2 and 3, the heat capacity of the generator needs to be higher than that of normal HPRTE system, which has only one evaporator. This higher generator capacity leads to making the required capacity of the cold gas cooler smaller to fix the properties at Point 9.1 and Point 3.2. Because there is a limitation, however, for the generator to be bigger, 0.7 is chosen as the maximum R_{evp} in this study.

Figure 4-6 shows the maximum COPs based on varying ambient temperatures, as well as constant maximum and minimum temperatures (generator and evaporator). Because the CGC (Evaporator 1) and the air conditioner (Evaporator 2) use the same temperatures, as shown in Figure 4-2, the Carnot COPs for the two evaporators are same. However, the ice maker needs colder refrigerant, and this reduces the Carnot COP of Evaporator 3. As expected, all Carnot COPs have minimum values at the hottest time of 3 PM.

Figures 4-7 and 4-8 show the COP variations of the three evaporators, obtained from Equation (4-8). The COPs have the same trend as the Carnot COPs since the refrigeration efficiency is held constant. The total COP variations of the system are shown in Figures 4-9 and 4-10 for η_R of 0.5 and 0.7. Obviously, the higher refrigeration efficiency the higher COP_{sys}. The minimum COP_{sys}s occur at different times, but they are in the day time between 12:00PM and 15:00PM.

For R_{evp}=0.3, Figures 4-11 through 4-14 represent the results when the ratio is 0.3 for η_R equal to 0.5 and 0.7. Heat removed from the surroundings at Evaporator 1 is fixed at all times to maintain HPC inlet gas temperature. While Evaporator 1 cools the HPC inlet gas, Evaporator 2 is used as an air conditioner which maintains the indoor

temperature of 27°C in a conditioned space of 671.1m^2. As the air conditioning required load decreases, Evaporator 3 makes more ice. Because the COP for Evaporator 3 is lower than that of Evaporator 2, the allocated heat supply required from the generator for the ice maker is relatively higher even though the heat rejected from the surroundings is of the same order for Evaporators 1 and 2.

For R_{evp}=0.5: As the capacity of the heat rejected to Evaporator 1 is lower than that of the 0.3 R_{evp} case, as shown in Figures 4-15 and 4-16, the heat supplied to Evaporator 1 is also somewhat less while the heat absorbed in Evaporators 2 and 3 are somewhat more. Clearly, when the refrigeration efficiency is higher as in Figures 4-17 and 4-18, the loads are smaller for Evaporators 1 and 2, so the system can make more ice.

For R_{evp}=0.7, Figures 4-19 through 4-22 show that the allocated heat supply to Evaporator 1 is reduced, while those of Evaporators 2 and 3 are increased. Because the HPC inlet gas temperature must be maintained, more generator heat is needed for air conditioning and icemaking. Approximately 9.85% more heat is needed to increase R_{evp} from 0.3 to 0.7. For this increase of generator heat capacity, almost double the floor area can be cooled, and 28% and 19% more ice can be made for η_R of 0.5 and 0.7, respectively.

Figures 4-23 and 4-24 and Table 4-2 show the comparison of results with various R_{evp} values. The amount of ice produced hourly is shown in Figures 4-9 and 4-18. As R_{evp} is increased, larger amounts of ice are produced, but the rate of increase is reduced. The more the generator cools the gas in the gas turbine system, the less the required heat capacity of Evaporator 1, while the total amount of ice produced per day and conditioned area are increased.

Summary

A steady-state gas path model and a second law VARS analysis were used for a thermodynamic performance study of a novel cooling and power cycle that combines a semi-closed cycle gas turbine called the HPRTE with a VARS.

As shown above, if the generator capacity is higher, Evaporator 1 can be downsized, and additional heat from generator is better used in different applications, such as air conditioning and ice-making. The ice can be used to chill water to cool hot fluids inside the HPRTE/VARS the next day – which can increase the total thermal efficiency of the system – or for additional air conditioning. The amount of ice available can ideally produce twice the total daily cooling capacity in air conditioning. This cooling capacity could be used to cool a second building of a size similar to the design building, or be used in a load leveling manner to downsize the entire system (thus reducing capital costs). Ice can also be used for other applications, and if the air conditioning load is limited, much more ice can be produced. However, if the ratio of Evaporator 2 and 3 loads to Evaporator 1 load needs to change, the process of re-sizing the heat exchangers is non-linear and needs to be taken into account. There is also a limitation on the upsizing possible for the generator, thus requiring an optimization.

Considering all the issues/points mentioned above it can be concluded that the combined HPRTE/VARS, or PoWER, cycle seems to be a promising candidate for distributed power generation especially in hot climates where the summer load-leveling benefits are important.

Second Law Analysis Using Variable External Temperatures

As discussed above, most evaporators have a single temperature theoretically in the refrigeration side. However, the fluids being cooled down by evaporators or

generators do not have to have a single value depending on heat exchanger types. In this sub chapter, variable or constant external temperatures are selected to obtain Carnot COP.

Thermodynamic Model

The Second Law model for the VARS is based on the system defined by the dashed line in Figure 4-25. The system receives high temperature heat from the gas turbine, rejects heat to the ambient, and receives low temperature heat from three reservoirs – the gas turbine, building air conditioning, and the ice storage subsystem. Pump and fan power are neglected, though those parasitic losses may be incorporated into the gas turbine efficiency calculation. In utilizing a refrigeration efficiency, the first step is to calculate the performance of a reversible system operating among these thermal reservoirs. It may be shown that the reversible system indicated is equivalent to three traditional Carnot refrigerators (heat actuated, not vapor compression), each cooling one of the three loads. The Carnot COP of such systems is a well-known function of the three temperatures. Since the cooling environments and temperatures differ for each of the three evaporators, each will have a unique Carnot COP. In this work, the Second Law refrigeration efficiencies of each of the three refrigeration systems are assumed equal, corresponding to an equivalent maturity of the technology. The relationship between cooling rate and heating rate for each of the three refrigeration systems is set by the assumed refrigeration efficiency and the value of the Carnot COP determined from the reservoir temperatures.

However, there are several reasonable approaches to calculating the Carnot COP, based on different choices of reservoir temperatures. One approach is to perform VARS cycle calculations for a particular configuration and to use the generator temperature for

the maximum value, the ambient as the intermediate value, and the three evaporator temperatures of the refrigerant as the low temperature values. This is considered an internal Carnot COP as in Lear et al. [41], and it does not include the irreversibilities in the heat exchangers.

A second approach is to take the maximum and minimum temperatures necessary for calculation of Carnot COP to be at the hot gas inlet (State 9.1 in Figure 4-1) and the cold gas exit (State 3.2), respectively. A variant of this approach is to take into account that the external reservoirs are not at fixed temperature, but rather change temperature as a result of the heat transfer. The COP for a reversible cycle under these circumstances has been derived in the present work.

The temperatures used for the Carnot COP calculations are given in Table 4-3. T_H, T_{L1}, T_{L2}, and T_{L3} are the internal working fluid temperatures, which are constants at all times. The calculation involving variable reservoir temperatures require knowledge of both the inlet and the exit temperature, so these are also listed. However, the external temperatures of the air conditioner (Evaporator 2) and ice maker (Evaporator 3) are taken to be constants.

In this paper, external temperatures are used for the generator and the evaporators. Figure 4-26 shows four cases using different external temperatures for the system:

- Constant generator and evaporator temperatures
- Variable generator temperature with constant evaporator temperature
- Constant generator temperature with variable evaporator temperature
- Variable generator and evaporator temperatures

For Case (a), the higher temperature, T_{Gi} is used for the constant reservoir temperature while the lower temperature, T_{Eo} is used for the lower-reservoir temperature (Figure 4-26 (a)). The First and Second laws for the case may be written as

$$\dot{Q}_{in,1} = \dot{Q}_{H,1} + \dot{Q}_{L,1} + \dot{W}_{p,1} \approx \dot{Q}_{H,1} + \dot{Q}_{L,1} \tag{4-10}$$

$$\dot{Q}_{out,1} = \dot{Q}_{o,1} = \dot{Q}_{in,1} \tag{4-11}$$

$$\dot{P}_{s,1} - \frac{\dot{Q}_{H,1}}{T_{Gi}} - \frac{\dot{Q}_{L,1}}{T_{Eo}} + \frac{\dot{Q}_{o,1}}{T_o} \geq 0 \tag{4-12}$$

where $\dot{Q}_{in,1}$ is energy input to the system, $\dot{Q}_{H,1}$ is allocated heat input for the Evaporator 1 from the generator, $\dot{Q}_{L,1}$ is heat input from the Evaporator 1, and $\dot{W}_{p,1}$ is pump work for Evaporator 1, neglected here. $\dot{Q}_{out,1}$ and $\dot{Q}_{o,1}$ are the heat released outside of the system and $\dot{P}_{s,1}$ is the entropy production for Sub-system 1.

Combining Equations (4-10), (4-11), and (4-12), we obtain

$$\dot{Q}_{L,1} \frac{T_{Eo} - T_o}{T_{Eo}} \geq \dot{Q}_{H,1} \frac{T_o - T_{Gi}}{T_{Gi}} \tag{4-13}$$

For the reversible case, the entropy production is zero, so the equality in Equation (4-13) is appropriate, yielding the Carnot COP for Sub-system 1 and Case (a):

$$COP_{C1,a} = \frac{\dot{Q}_{L,1}}{\dot{Q}_{H,1}} = \left(\frac{T_{Eo}}{T_o - T_{Eo}} \right) \left(\frac{T_{Gi} - T_o}{T_{Gi}} \right) \tag{4-14}$$

For Case (b), in which the generator temperature changes from T_{Gi} to T_{Go} as shown at Figure 4-27 (A), the local COP at Point x can be written as

$$COP_{dx} = \frac{d\dot{Q}_{Lx,1}}{d\dot{Q}_{Hx,1}} \tag{4-15}$$

If mass flow rate and heat capacity are constants, the heat inputs to the system are,

$$\dot{Q}_{H,1} = \int_{x=1}^{x=0} d\dot{Q}_{Hx,1} = \int_{T_{Go}}^{T_{Gi}} \dot{m} C_p dT_x = \dot{m} C_p (T_{Gi} - T_{Go}) \tag{4-16}$$

$$\dot{Q}_{L,1} = \int_{x=1}^{x=0} d\dot{Q}_{Lx,1} = \int_{1}^{0} COP_{dx} d\dot{Q}_{Hx,1} \tag{4-17}$$

When the Carnot COP is introduced, we get

$$\dot{Q}_{L,1} = \int_{T_{Go}}^{T_{Gi}} \dot{m} C_p \left(\frac{T_x - T_o}{T_x} \right) \left(\frac{T_{Eo}}{T_o - T_{Eo}} \right) dT_x \tag{4-18}$$

Combining Equations (4-16) and (4-18) with some algebra, we obtain

$$COP_{C1,b} = \left(\frac{T_{Eo}}{T_o - T_{Eo}} \right) \left[1 - \frac{T_o}{T_{Gi} - T_{Go}} \ln \left(\frac{T_{Gi}}{T_{Go}} \right) \right] \tag{4-19}$$

With similar analysis as shown in Figure 4-27 (B), the Carnot COP for Case (c) can be easily determined

$$\dot{Q}_{L,1} = \int_{y=1}^{y=0} d\dot{Q}_{Ly,1} = \int_{T_{Eo}}^{T_{Ei}} \dot{m}_E C_{p,E} dT_y = \dot{m}_E C_{p,E} (T_{Ei} - T_{Eo}) \tag{4-20}$$

$$\dot{Q}_{H,1} = \int_{y=1}^{y=0} d\dot{Q}_{Hy,1} = \int_{T_{Eo}}^{T_{Ei}} \frac{d\dot{Q}_{Ly,1}}{COP_{dy}} \tag{4-21}$$

When the Carnot COP expression is again introduced, we obtain

$$\dot{Q}_{H,1} = \int_{T_{Eo}}^{T_{Ei}} \left(\frac{T_{Gi}}{T_{Gi} - T_o} \right) \left(\frac{T_o - T_y}{T_y} \right) \dot{m}_E C_{p,E} dT_y \tag{4-22}$$

Finally, we can get the Carnot COP for Case (c) as

$$COP_{C1,c} = \frac{\left(\frac{T_{Gi} - T_o}{T_{Gi}} \right)}{\left[\frac{T_o}{T_{Ei} - T_{Eo}} \ln \left(\frac{T_{Ei}}{T_{Eo}} \right) - 1 \right]} \tag{4-23}$$

In the case that both the generator and evaporator temperatures vary, it is assumed that they are independent. Then the Carnot COP for Case (d) is determined as

$$COP_{C1,d} = \frac{\left[1 - \frac{T_o}{T_{Gi} - T_{Go}} \ln \left(\frac{T_{Gi}}{T_{Go}} \right) \right]}{\left[\frac{T_o}{T_{Ei} - T_{Eo}} \ln \left(\frac{T_{Ei}}{T_{Eo}} \right) - 1 \right]} \tag{4-24}$$

Because the external temperatures of Evaporators 2 and 3 are constant in this study, only cases (a) and (b) are considered. Therefore Equations (4-14) and (4-19), in which the lower temperature is replaced with the constant air conditioned room temperature and ice temperature, T_{AC} and T_{ICE}, respectively, are also used for Evaporators 2 and 3.

For each evaporator, the refrigeration efficiency is specified as a Second Law-based efficiency parameter. Using the conventional COP definition,

$$COP_i = \frac{\dot{Q}_{E,i}}{\dot{Q}_{H,i}} \tag{4-25}$$

enables solving for the COP values, given the Carnot COP values calculated from the specified temperatures, and refrigeration efficiency, from

$$\eta_{R,i} = \frac{COP_i}{COP_{Ci}} \tag{4-26}$$

Solution Method

A computer code described previously by Khan et al. [29] simulates the performance of the PoWER system, using traditional cycle analysis methods. The code is capable of calculating engine performance (network output, water extraction flow rate and thermal efficiency) as a function of the input parameters (turbine inlet temperature, recuperator inlet temperature, generator temperature, evaporator exit temperature, low-pressure compressor ratio, recirculation ratio, turbomachinery efficiencies, heat exchanger effectiveness, equivalence ratio, fuel type, and pressure drops).

The intent of the current paper is to quantify the potential for ice-making and load leveling, not to analyze one particular design choice. Therefore, the VARS was simulated by a Second Law analysis as described above, using as input the results of the HPRTE code of Khan et al. [29]. The refrigeration efficiency is taken as a parameter to represent the maturity of the VARS technology chosen.

The temperatures used for determining the Carnot coefficient of performance, COP_{Carnot}, in this paper are tabulated in Table 4-3. The variation of the air conditioning hourly load and outdoor temperature [32] are shown in Figures 4-4 and 4-5, and the values of generator and evaporator temperatures are taken from Goswami et al. [30].

T_{ICE} was chosen by Huang et al. [42]. Figure 4-5 shows the maximum air conditioning load at 4 pm, while Figure 4-4 shows the maximum outdoor air temperature at 3 pm. This phase lag is because of thermal storage of the building, the movement of heat sources such as people, and time-dependent use of lights in the building [33].

The cycle input and output parameters – which will be used for a multi-temperature VARS study – are given in Table 4-4. The low-pressure compressor pressure ratio was based on prior studies [34] and is chosen for maximum thermal efficiency and reasonable water extraction. The inlet temperature of the gaseous mixture entering the high-pressure compressor is constant and is equal to 10°C, which is a practical limit taking into account the fact that the internal temperature of Evaporator 1 is fixed at 5°C. The turbomachinery polytrophic efficiencies were based on state-of-the-art values as reported by Mattingly [35] and Carcasci and Facchini [36]. The pressure drops in the recuperator and the combustor are similar to the values reported by Fiaschi et al. [37]. The equivalence ratio Φ was set to 0.9 based on previous studies with the HPRTE as given in Boza et al. [38]. It was assumed to be equal to 0.9, because it was expected that at least 10% excess air was necessary to ensure complete combustion [34]. The gas entering the combustor is diluted with excess air and the recirculated combustion products, to lower the adiabatic flame temperature towards the specified turbine inlet temperature (this provides benefits in emissions as well as combustion efficiency).

The effectiveness of the recuperator is assumed to be 0.85 and its inlet temperature (on the hot side) is assumed to be 800°C. These are conservative values based on existing commercial equipment [39]. Turbine blade cooling is considered for

higher turbine inlet temperatures, and the model for turbine blade cooling used is similar to that given by Massardo et al. [40]. This model gives a relation between the turbine inlet temperature and the percentage of the main air flow used for turbine blade cooling. Note that the intent here is to model a future system with advanced cooling and recuperator technology. This is the conservative choice from the standpoint of cooling and ice-making provided by the VARS, since the high firing and recuperator temperatures lead to a higher gas turbine efficiency, hence reduced waste heat for driving the VARS. That in turn limits the ice-making potential and the load leveling capability due to thermal energy storage. This conservative approach is chosen in order to balance the simplifications made in the thermal energy storage system, primarily the omission of parasitic power requirements and heat leaks.

Results and Discussion

The computer code of the HPRTE cycle and the Second Law approach of the multistage VARS were used to calculate hourly air conditioning and ice-making capacities. Given HPRTE data, which are calculated by Khan's model [29], together with the outdoor air temperature, air conditioning hourly required load, and refrigeration efficiency, the COP_{Carnot}, COPs, heat supplied to the generator, and heat removed from the surroundings for all three evaporators were calculated and the results plotted. In addition to those results, the amount of ice produced daily and hourly and the air conditioned floor area are shown. These values were determined for six cases, varying the ratio of the heat added from the air conditioner (Evaporator 2), and ice maker (Evaporator 3) to the heat added at Evaporator 1 for two different refrigeration efficiencies.

Figures 4-28 to 4-30 show the hourly Carnot COP for the evaporators. All cases have same trend; when ambient temperature is high, the Carnot COP is low and vice versa. For all evaporators, the Carnot COPs using external temperatures are higher than those using internal temperatures. This is because the evaporator external temperatures are higher than their internal temperatures as shown at Table 4-3. The temperature differences between the external temperatures and ambient temperature are smaller before sunset. Therefore, the Carnot COP has a maximum at around 5am for Evaporators 2 and 3.

Around the same time, the ambient temperature is less than the air conditioned room temperature. This results in the Carnot COP for Evaporator 2 being less than zero because when only the ambient and room temperatures are compared, no refrigeration effect is required. To avoid this problem, the Carnot COP equations need certain conditions imposed for four cases (Equations (4-14), (4-19), (4-23), and (4-24));

$$T_{Gi} > T_o > T_{Eo} \tag{4-27}$$

$$\frac{T_{Gi} - T_{Go}}{\ln\left(\frac{T_{Gi}}{T_{Go}}\right)} > T_o > T_{Eo} \tag{4-28}$$

$$T_{Gi} > T_o > \frac{T_{Ei} - T_{Eo}}{\ln\left(\frac{T_{Ei}}{T_{Eo}}\right)} \tag{4-29}$$

$$\frac{T_{Gi} - T_{Go}}{\ln\left(\frac{T_{Gi}}{T_{Go}}\right)} > T_o > \frac{T_{Ei} - T_{Eo}}{\ln\left(\frac{T_{Ei}}{T_{Eo}}\right)} \tag{4-30}$$

For Evaporator 2, conditions (4-27) and (4-28) above are used by replacing T_{Eo} with T_{AC}. In this paper, rather than using negative Carnot COP, it is considered that Evaporator 2 would not function as an air conditioner when the ambient temperature is lower than the temperature produced in Evaporator 2. As a result, the air conditioning load from 1 to 7 a.m. is zero.

Figures 4-31 and 4-32 show the cooling effect of the eight different cases for Evaporators 2 and 3. It is obvious that the higher the refrigeration efficiency, the higher the refrigerating effect. When a variable lower temperature, due to sensible heat addition from the gas path, for Evaporator 1 with a fixed generator temperature is considered, the heat gain from Evaporators 2 and 3 is always highest, while it is lowest when a variable generator temperature and a fixed evaporator temperature are considered instead.

Ice produced per day is indicated in Table 4-5 based on the results of Figure 4-32 when no load and no thermal loss are considered during the process. Consider Case (a) when η_R = 0.5. When the ice storage is used to replace conventional air conditioning having 2.5 of COP without loss, 8279 kWh of electric energy can be saved a day. In a practical ice storage system, however, there must be a loss associated with making ice and using ice for air conditioning or refrigeration, but ice can be made and used simultaneously. Therefore, the total volume of a real ice storage does not have to be so large as shown in Table 4-5.

The allocated heat input from the generator to each sub-system is shown in Figures 4-33 to 4-35. Because $\dot{Q}_{L,1}$ is 234.6 kW continuously, more heat is needed at lower refrigeration efficiencies to produce enough cooling. Cases (b) and (d) use a

variable generator temperature, which means that the average value of the inlet and exit temperatures is lower than T_{Gi}, so, more heat is necessary than those of Cases (a) and (c), respectively. However, the allocated generator heat inputs for Evaporator 2 and 3 show the exact opposite trends. More heat is necessary when the efficiency is higher and Evaporator 1 is using a variable temperature.

Figure 4-36 shows the efficiency ratio based on the efficiency of Case (a). To achieve the same COP for Case (a), about 45% higher efficiency is needed for Case (b). With lower efficiency, however, Cases (c) and (d) can achieve the same refrigerating effect.

Summary

A load-leveling implementation of the PoWER cycle has been analyzed in order to determine the potential for demand side management via cold thermal energy storage integrated with this novel distributed generation system. The analysis included an exploration of the appropriate methods of determining the performance of the refrigeration sub-system using Second Law efficiency as a parameter.

The results indicated that the daily storage of thermal energy in the form of ice is roughly equivalent to the air conditioning supplied directly to a building, for typical hot climate conditions. The sizing of the engine relative to the building depends on economic factors outside the scope of this analysis; however, it is clear that a very significant decrease in peak power requirement for air conditioning is achievable.

Results presented in this paper also illustrate that internal temperatures are best used when calculating Carnot COP values, convenient for conceptual design of such systems. Often the external temperatures would be better defined, and therefore they would be attractive as a means of simplifying the overall analysis, combining

irreversibilities of the heat exchangers with those of the refrigeration cycle. However, it is clear that non-physical results can easily occur when the ambient temperature approaches the low-temperature reservoir temperature, rendering that approach ineffective. It is therefore recommended that heat exchanger irreversibilities be included separately from cycle irreversibilities.

Table 4-1. Input/output parameters of the HPRTE system.

Parameters	Values
Turbine inlet temperature [°C]	1400
Recuperator inlet temperature [°C]	800
High pressure compressor ratio	9.02
Low pressure compressor ratio	2
Turbo-machinery polytropic efficiencies [%]	90
Combustor equivalence ratio, Φ	0.9
Effectiveness of the recuperator	0.85
Effectiveness of the intercooler	0.85
Relative humidity of inlet air,	0.9
Combustor pressure drop	0.05
Intercooler pressure drop	0.03
Generator pressure drop	0.03
Evaporator 1pressure drop	0.03
Recuperator pressure drop (HPT exit side)	0.06
Recuperator pressure drop (LPC exit side)	0.02
Generator temperature, T_G [°C]	100
Evaporator 1, 2 temperature, $T_{E1,2}$ [°C]	5
Evaporator 3 temperature, T_{ICE} [°C]	-25
Thermal efficiency, η_{th} [%]	40.4
Net power of system [kW]	455
Mass flow rate of fuel [kg/s]	0.0243
Mass flow rate of water extracted [kg/s]	0.0368
Mass flow rate of air input [kg/s]	0.43
Modified recirculation ratio, R_{mod}	2.45

Table 4-2. Comparison of results

R_{evp}	0.3	0.5	0.7
Generator heat capacity, $\dot{Q}_{Gen}$ [kW]	417.6	440.0 (5.36%)	458.73 (9.85%)
Evaporator 1 heat capacity, $\dot{Q}_{E}$ [kW]	257.0	234.6 (-8.72%)	215.9 (-16.0%)
Area to be Air Conditioned [m^2]	671.1	1021.2 (52.2%)	1315.3 (96.0%)
Ice Produced when η_R=0.5 [kg/day]	20235	23308 (13.2%)	25883 (24.2%)
Ice Produced when η_R=0.7 [kg/day]	40618	44782 (10.3%)	48273 (18.8%)

Table 4-3. Input parameters of the VARS

Parameters	Values [°C]
T_H, Internal temperature of the generator	100
T_{Gi}, Inlet temperature of the generator	452.3
T_{Go}, Exit temperature of the generator	68.46
T_{L1}, Internal temperature of the evaporator 1	5
T_{Ei}, Inlet temperature of the evaporator 1	29.25
T_{Eo}, Exit temperature of the evaporator 1	10
T_{L2}, Internal temperature of the evaporator 2	5
T_{AC}, Air conditioned room temperature	25
T_{L3}, Internal temperature of the evaporator 3	-7
T_{ICE}, Ice temperature	0

Table 4-4. Input/output parameters of the HPRTE system.

Parameters	Values
Turbine inlet temperature [°C]	1400
Recuperator inlet temperature [°C]	800
High pressure compressor ratio	9.02
Low pressure compressor ratio	2
Turbo-machinery polytropic efficiencies [%]	90
Combustor equivalence ratio, Φ	0.9
Effectiveness of the recuperator	0.85
Effectiveness of the intercooler	0.85
Relative humidity of inlet air,	0.9
Combustor pressure drop	0.05
Intercooler pressure drop	0.03
Generator pressure drop	0.03
Evaporator 1pressure drop	0.03
Recuperator pressure drop (HPT exit side)	0.06
Recuperator pressure drop (LPC exit side)	0.02
Thermal efficiency, η_{th} [%]	40.4
Net power of system [kW]	455
Mass flow rate of fuel [kg/s]	0.0243
Mass flow rate of water extracted [kg/s]	0.0368
Mass flow rate of air input [kg/s]	0.43
Modified recirculation ratio, R_{mod}	2.45

Table 4-5. Ice produced a day [Ton/day (L/day)].

	η_R = 0.7	η_R = 0.5
Case (a)	75.354 (81907)	52.159 (56695)
Case (b)	51.275 (55734)	34.960 (38000)
Case (c)	78.964 (85830)	55.769 (60618)
Case (d)	54.885 (59658)	38.570 (41924)

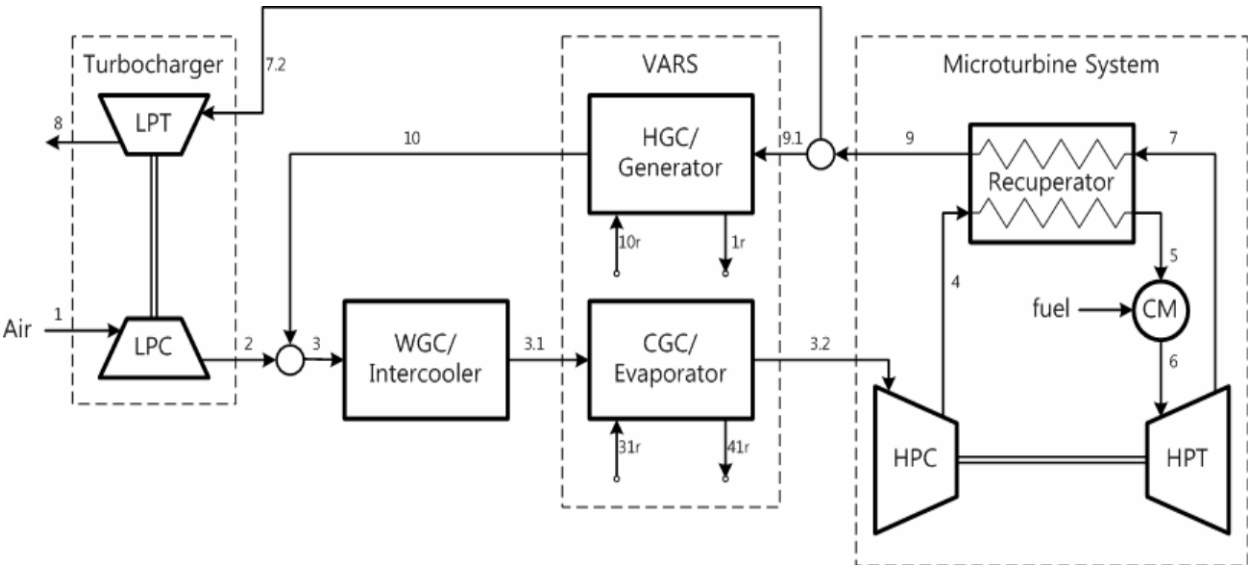

Figure 4-1. Gas turbine flowpath of the PoWER system

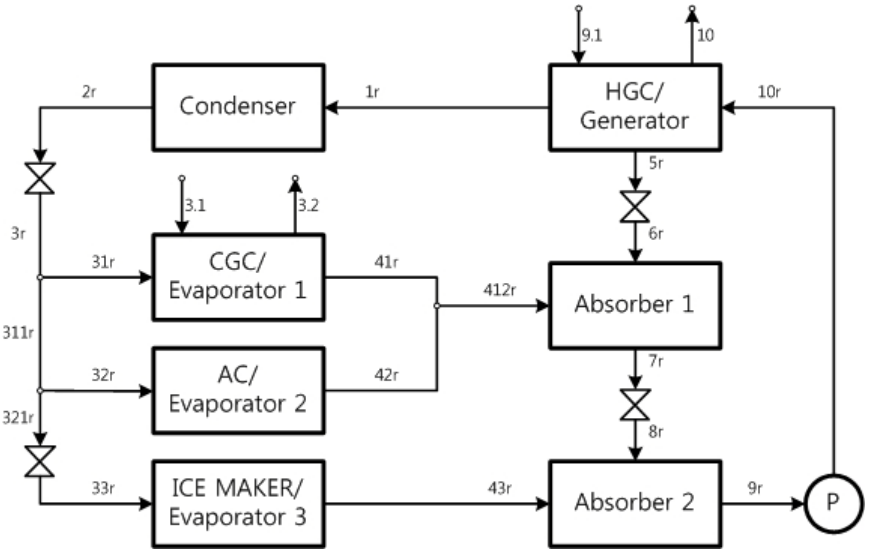

Figure 4-2. The multi-stage vapor absorption refrigeration system

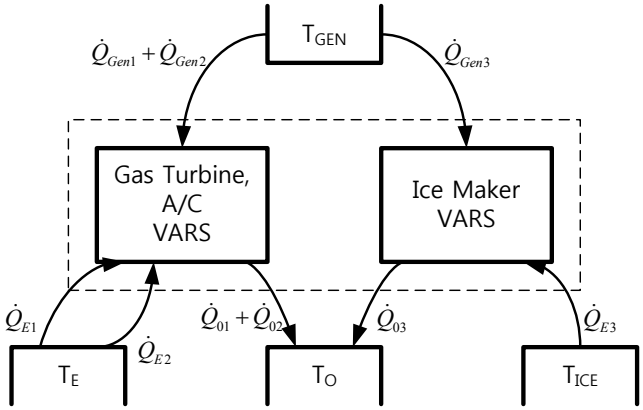

Figure 4-3. The multi-stage simple vapor absorption refrigeration system

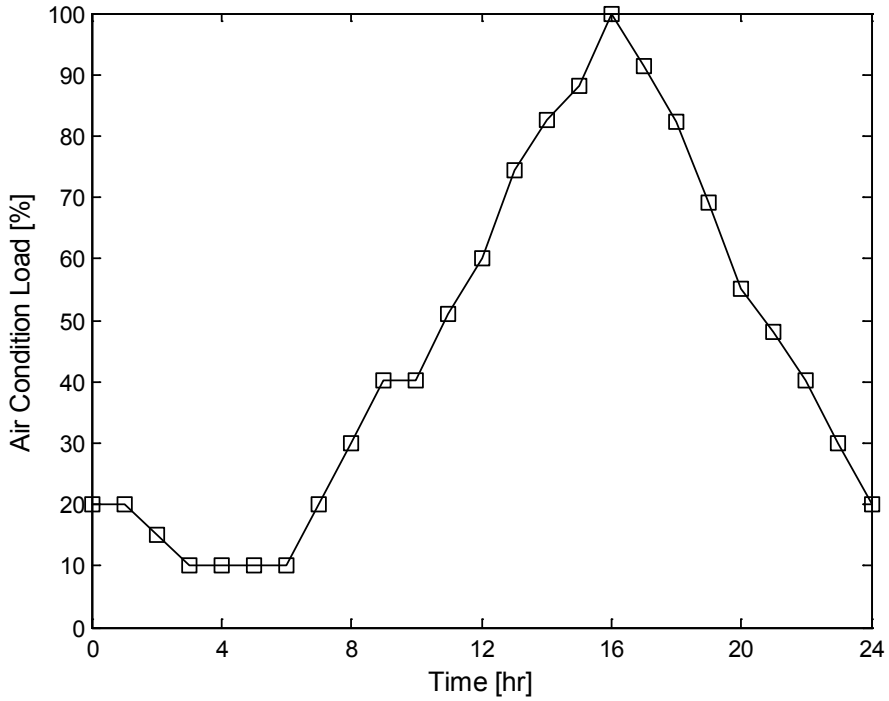

Figure 4-4. Air conditioner hourly load for typical summer day

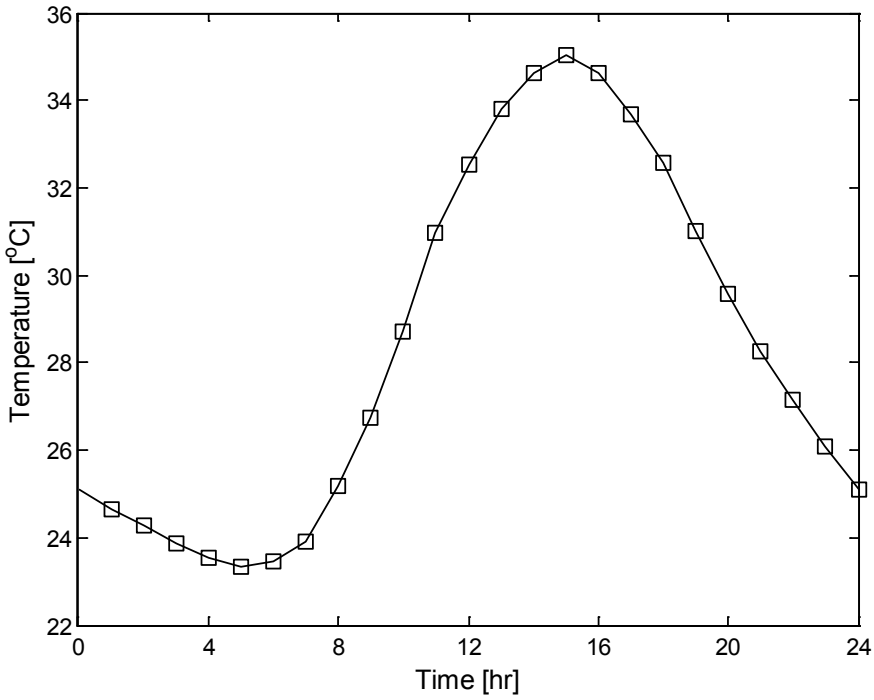

Figure 4-5. Typical variations of outdoor air temperature at 42° north latitude on July 1

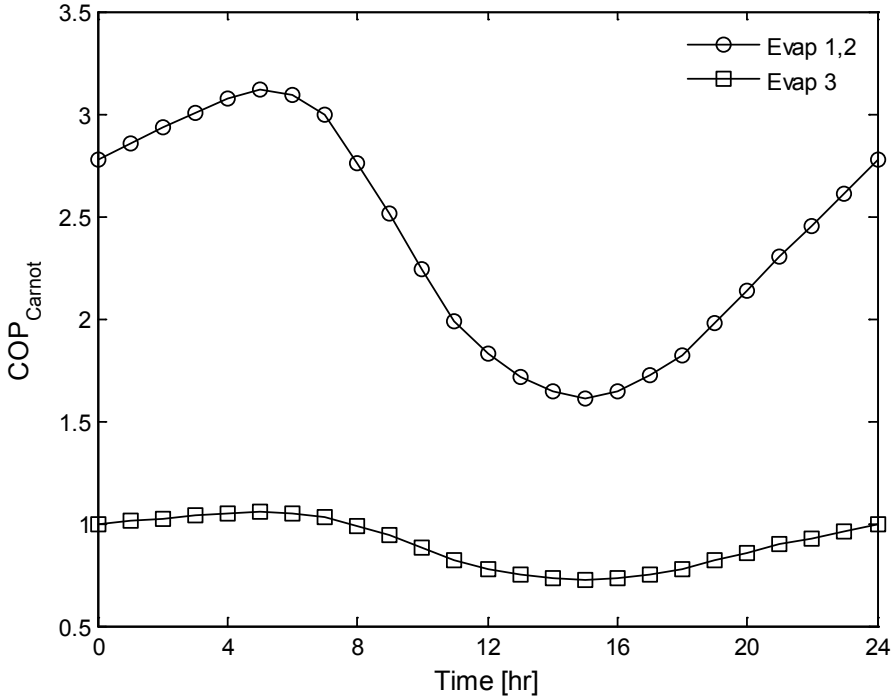

Figure 4-6. Carnot COP variations for the three evaporators

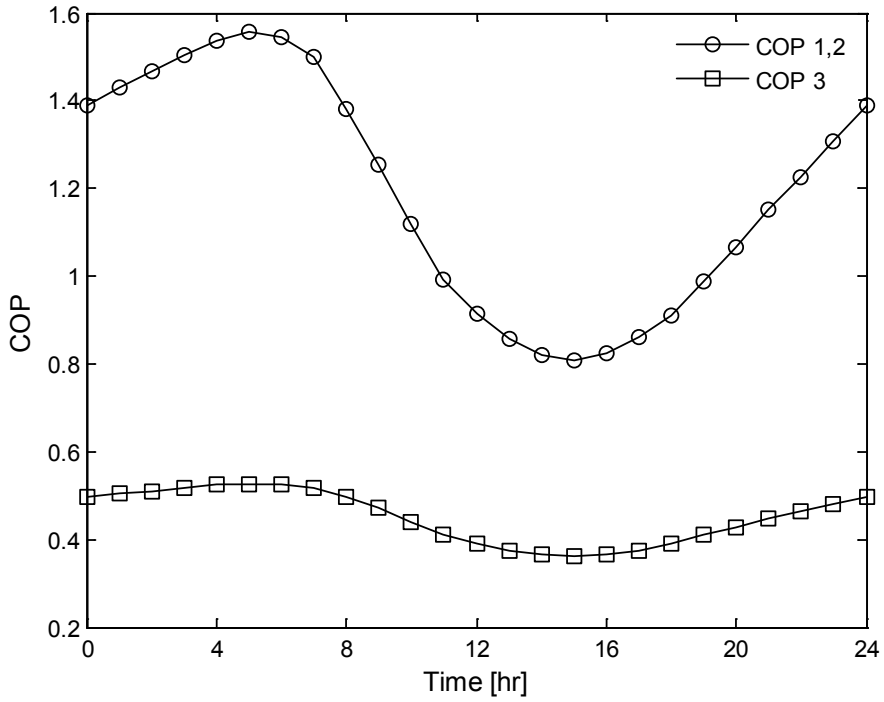

Figure 4-7. COP variations for three evaporators at η_R=0.5

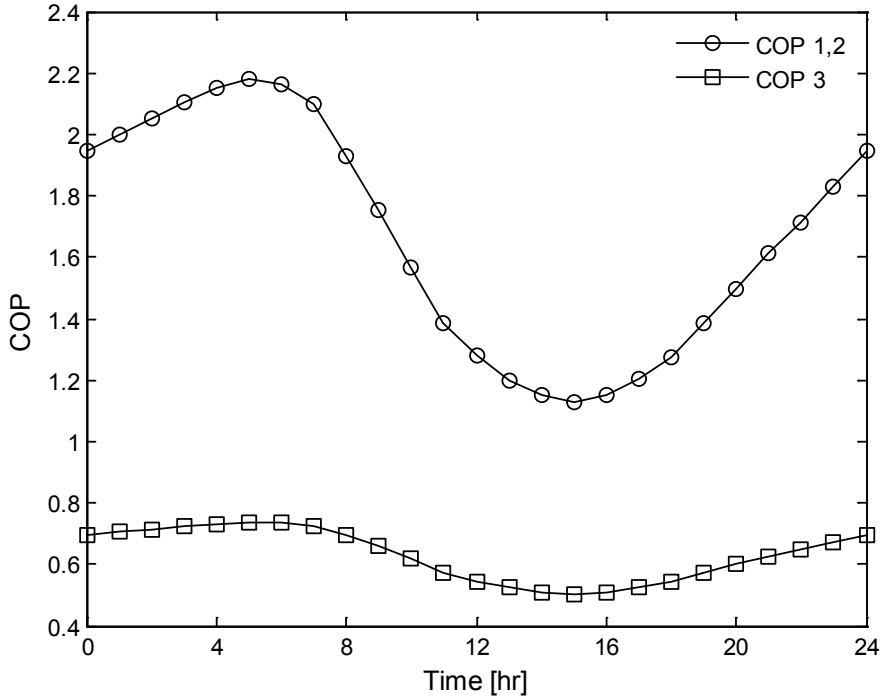

Figure 4-8. COP variations for three evaporators at η_R=0.7

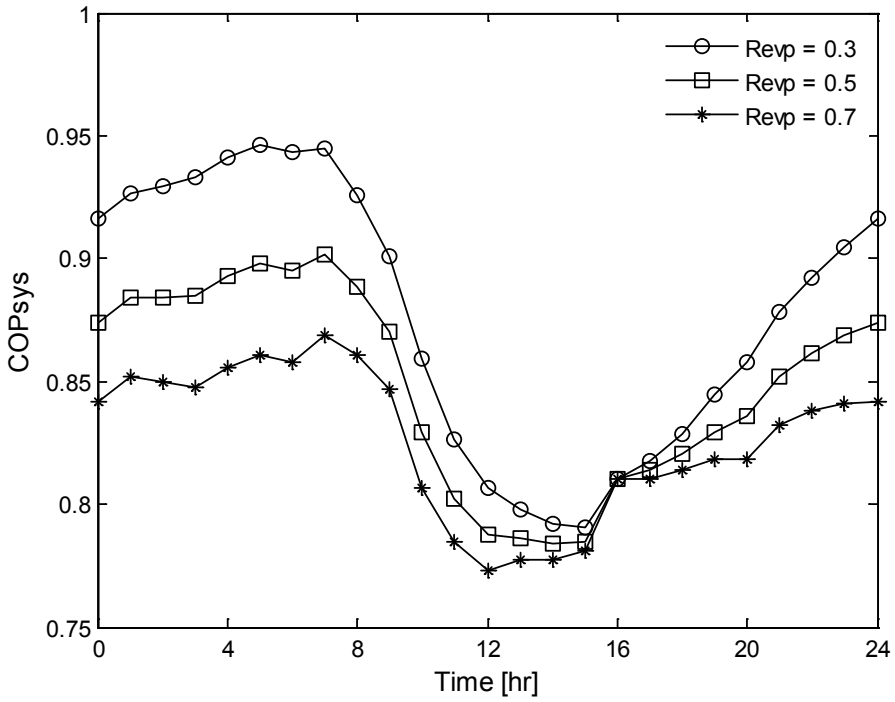

Figure 4-9. Total COP variations of the system at η_R=0.5

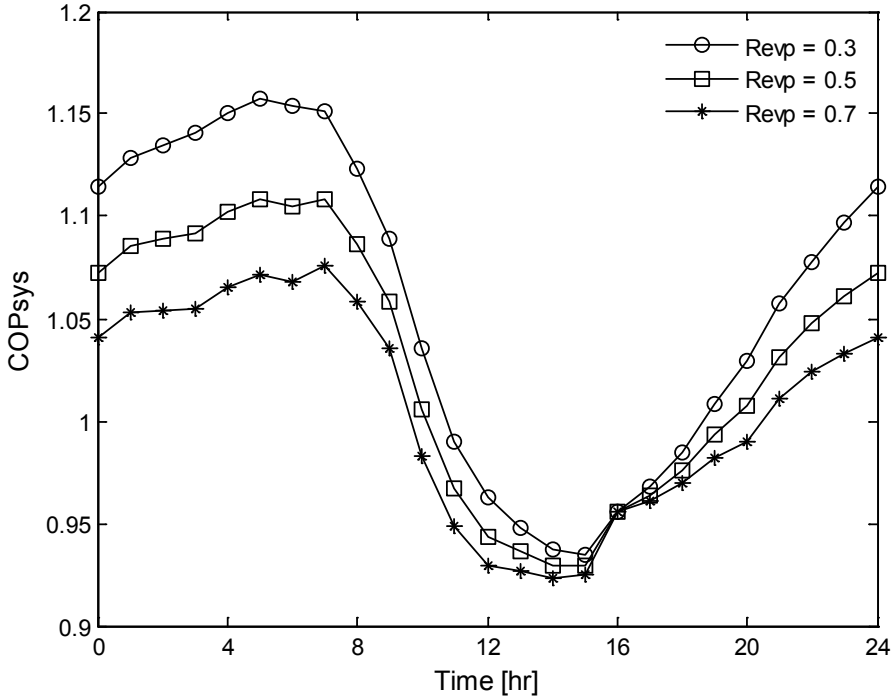

Figure 4-10. Total COP variations of the system at η_R=0.7

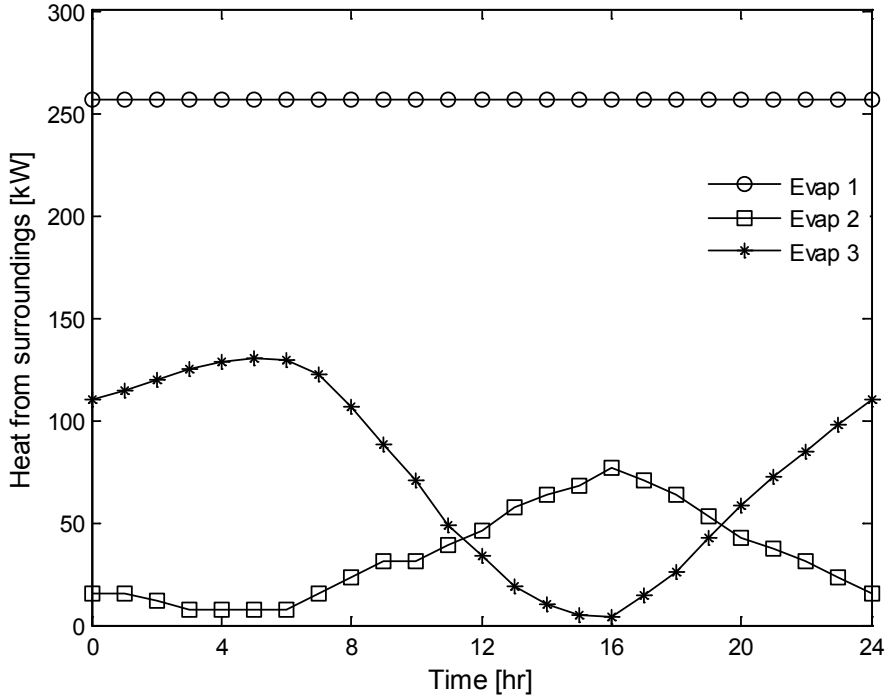

Figure 4-11. Heat removed from surroundings at η_R=0.5, R_{evp}=0.3

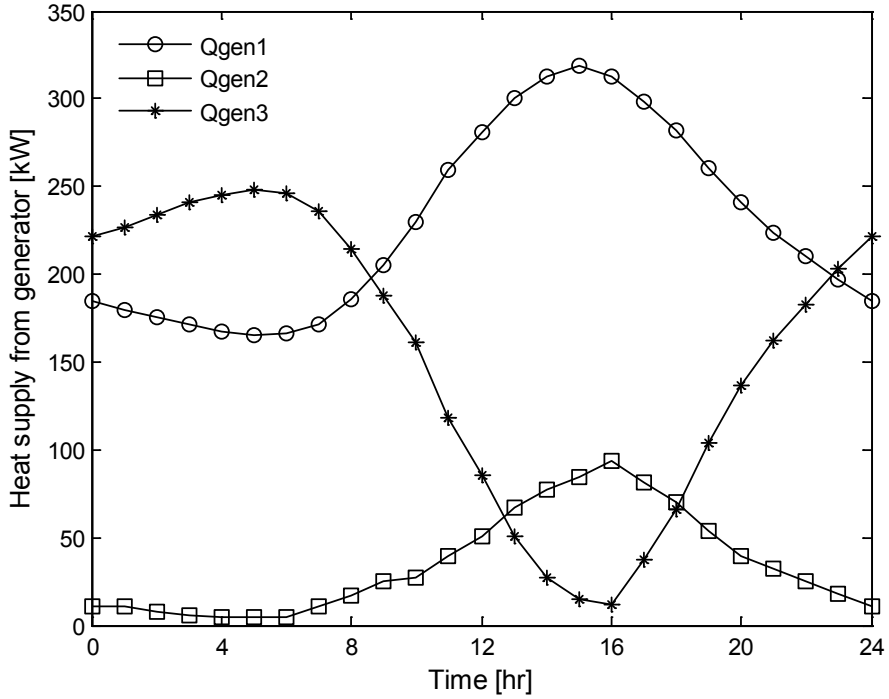

Figure 4-12. Allocated heat supply from generator at η_R=0.5, R_{evp}=0.3

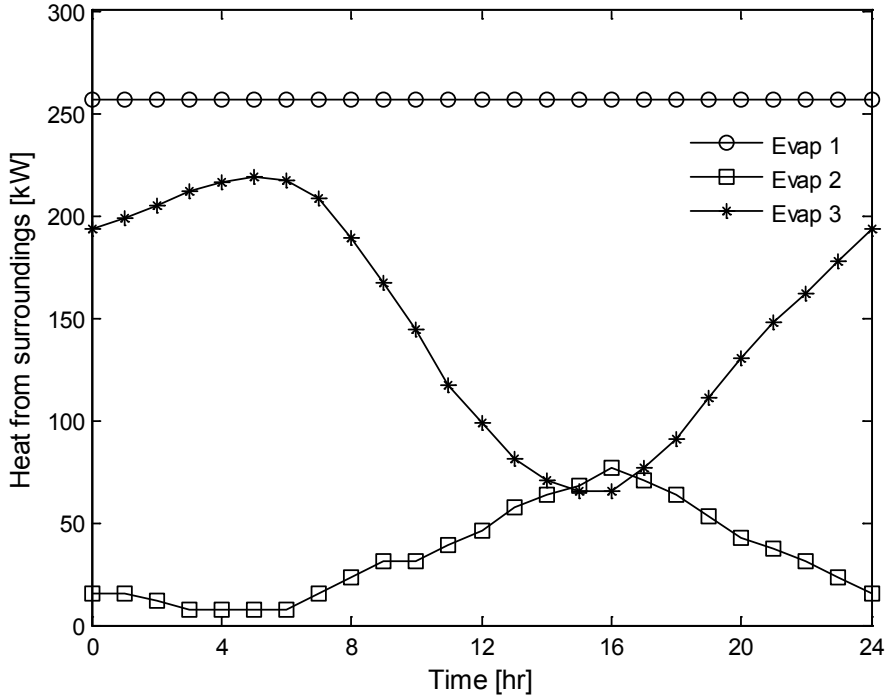

Figure 4-13. Heat removed from surroundings at η_R=0.7, R_{evp}=0.3

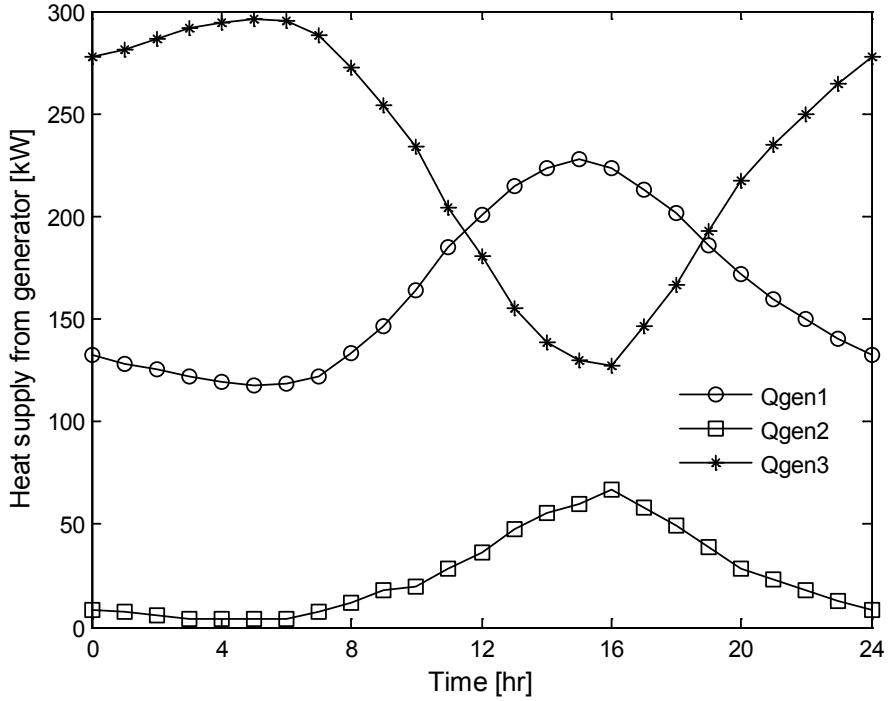

Figure 4-14. Allocated heat supply from generator at η_R=0.7, R_{evp}=0.3

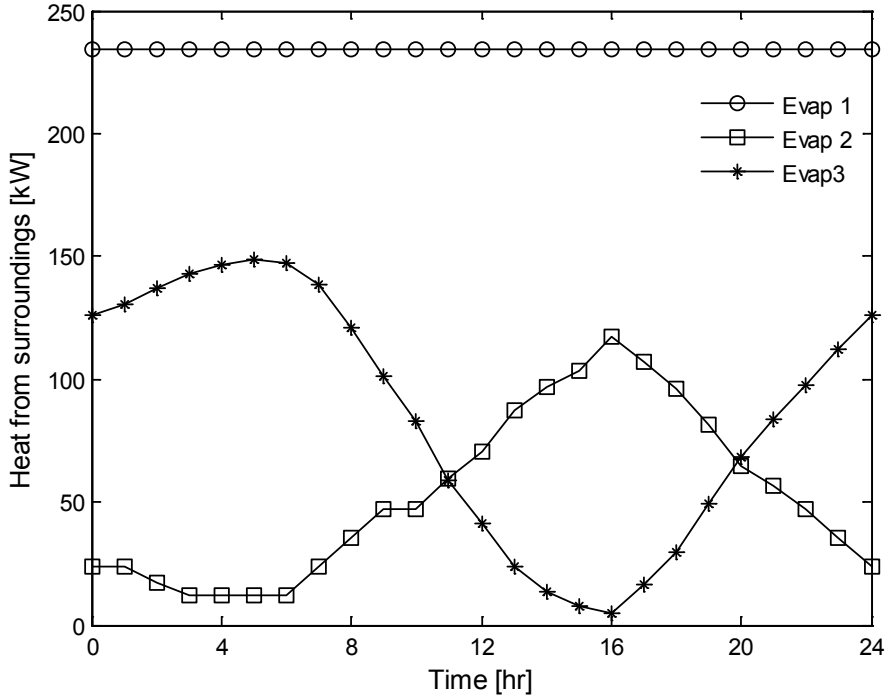

Figure 4-15. Heat removed from surroundings at η_R=0.5, R_{evp}=0.5

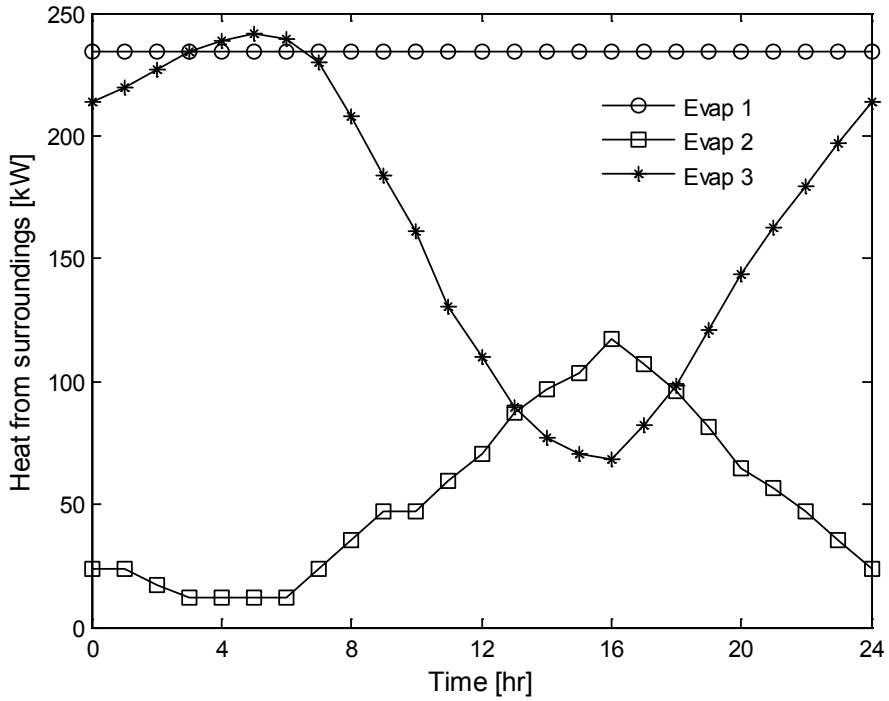

Figure 4-16. Heat removed from surroundings at η_R=0.7, R_{evp}=0.5

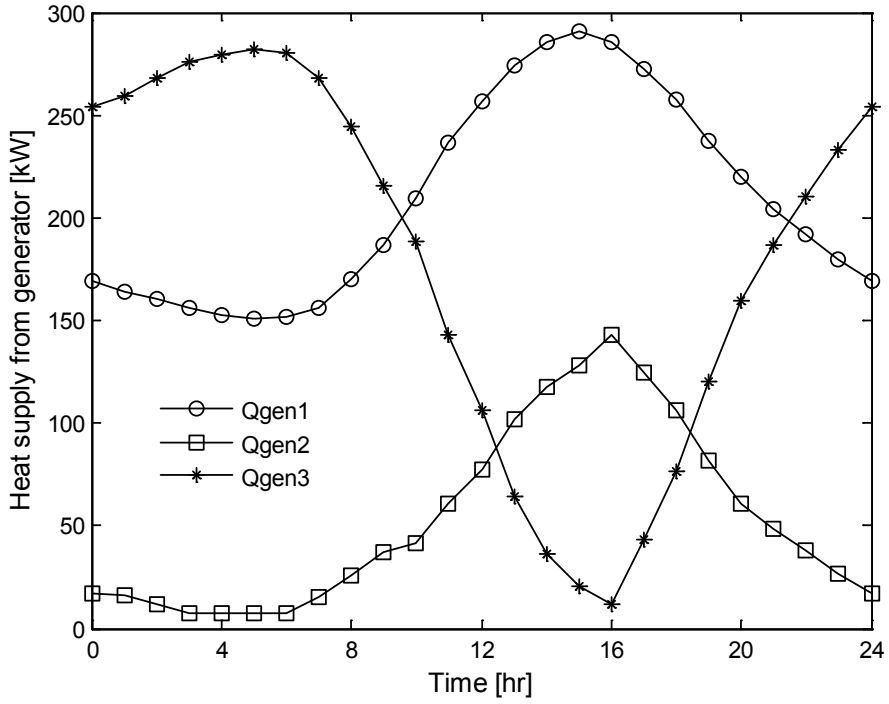

Figure 4-17. Allocated heat supply from generator at η_R=0.5, R_{evp}=0.5

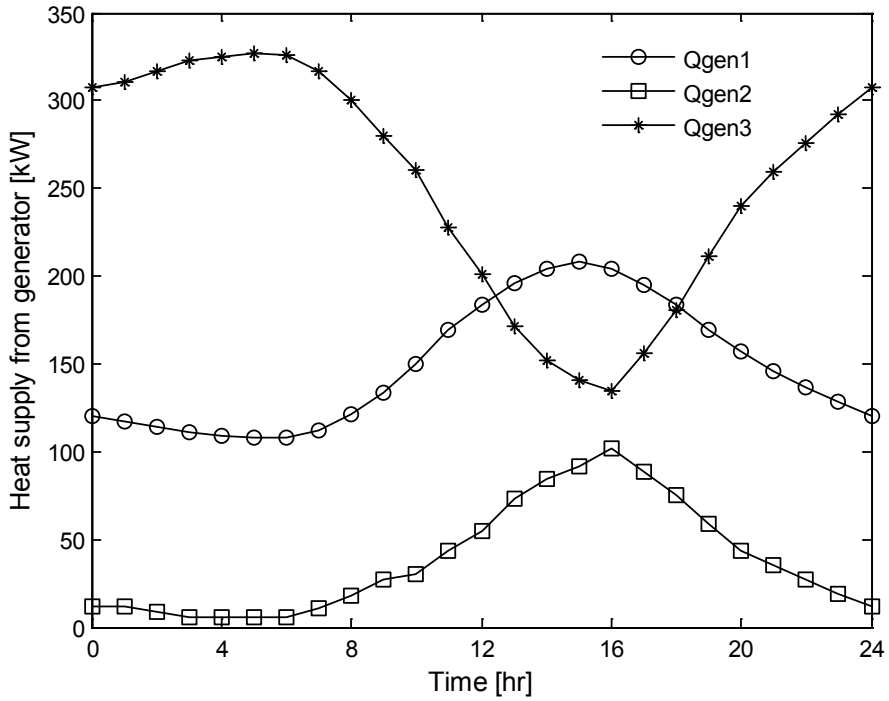

Figure 4-18. Allocated heat supply from generator at η_R=0.7, R_{evp}=0.5

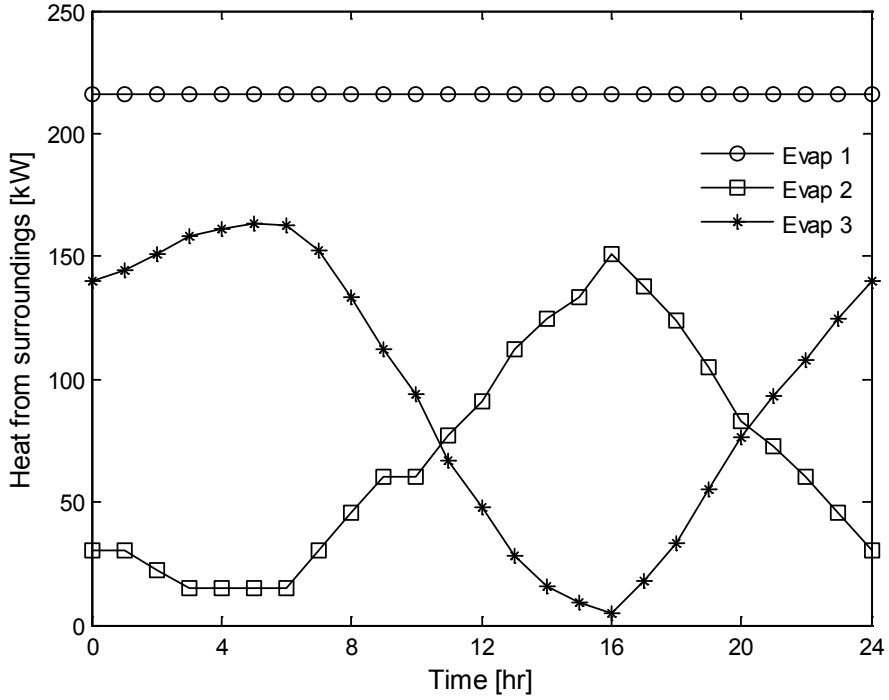

Figure 4-19. Heat removed from surroundings at η_R=0.5, R_{evp}=0.7

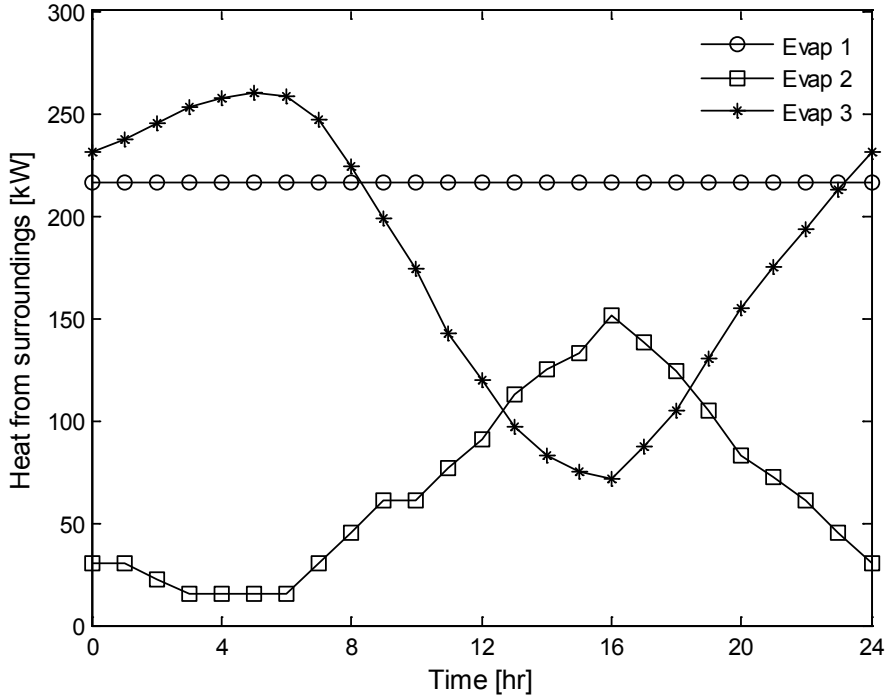

Figure 4-20. Heat removed from surroundings at η_R=0.7, R_{evp}=0.7

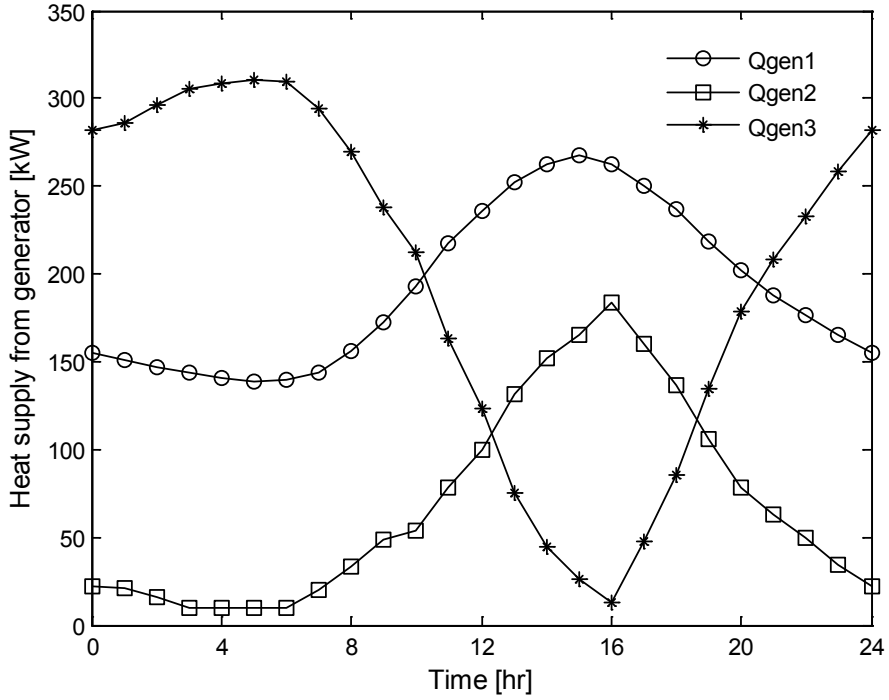

Figure 4-21. Allocated heat supply from generator at η_R=0.5, R_{evp}=0.7

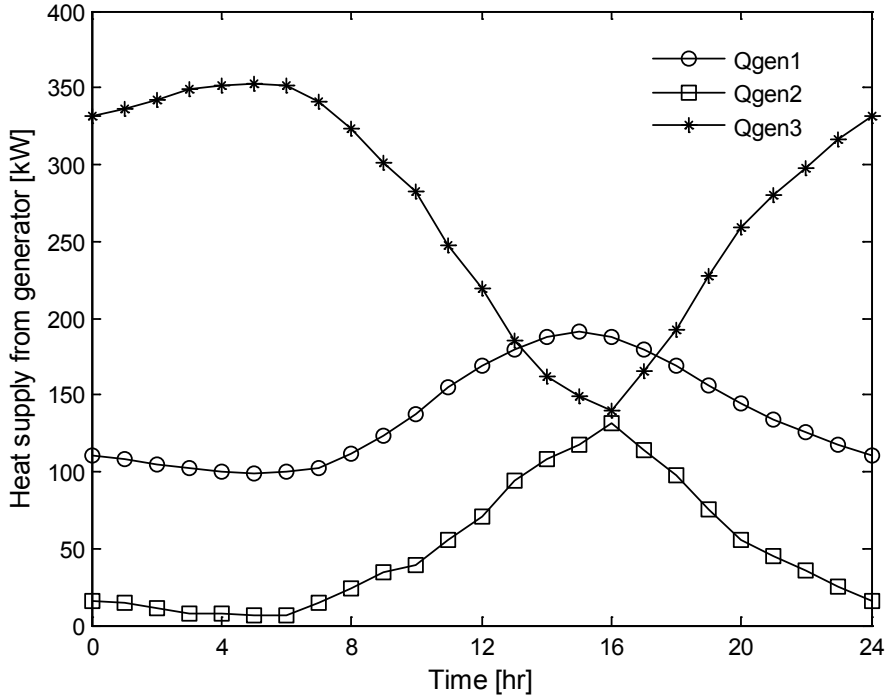

Figure 4-22. Allocated heat supply from generator at η_R=0.7, R_{evp}=0.7

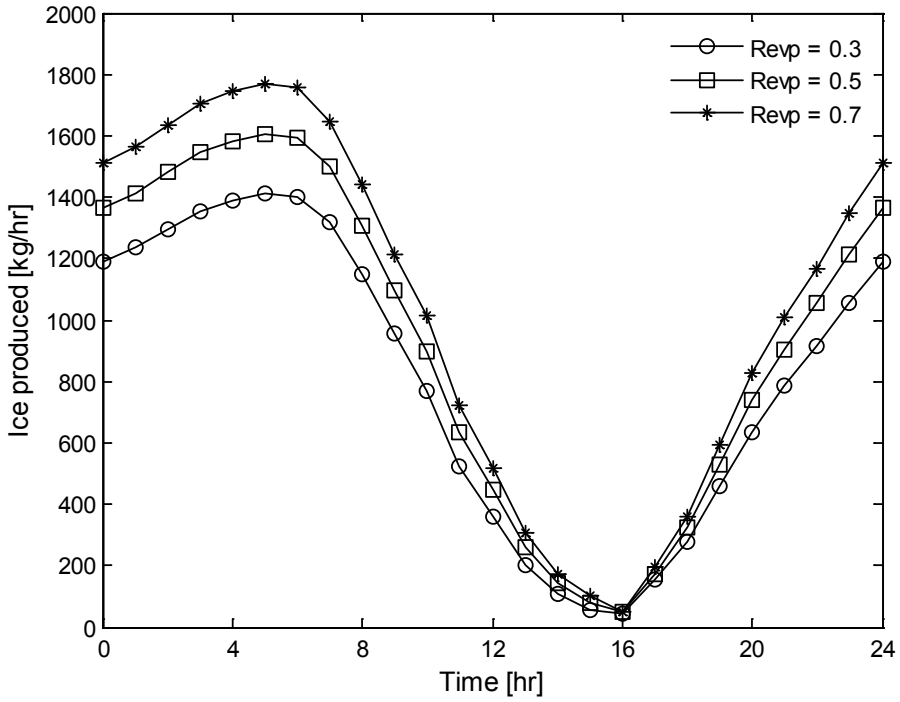

Figure 4-23. Ice produced hourly for various R_{evp} at η_R=0.5

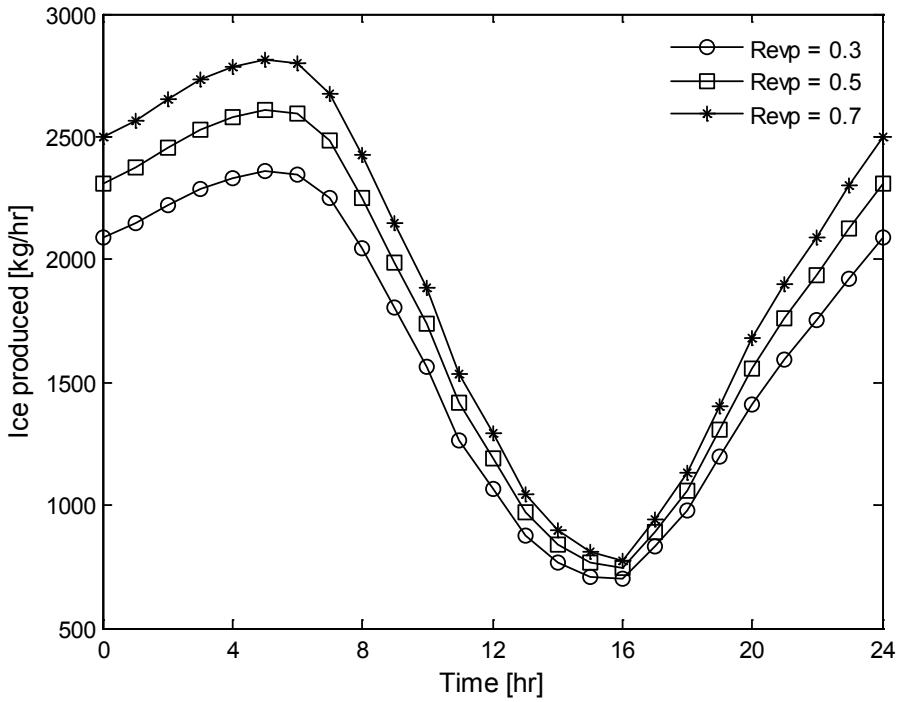

Figure 4-24. Ice produced hourly for various R_{evp} at η_R=0.7

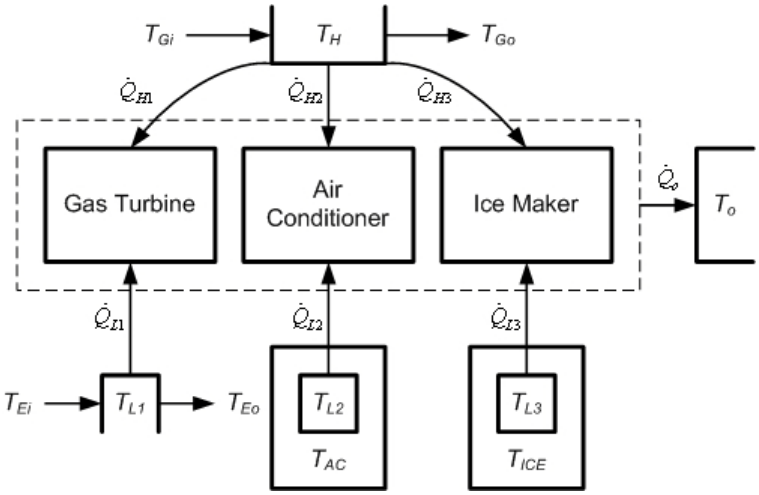

Figure 4-25. The multi-temperature simple vapor absorption refrigeration system

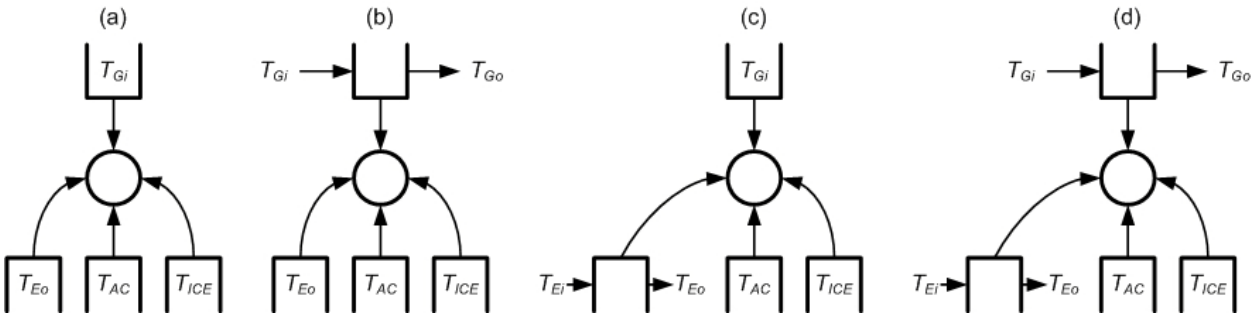

Figure 4-26. Four cases of external temperatures

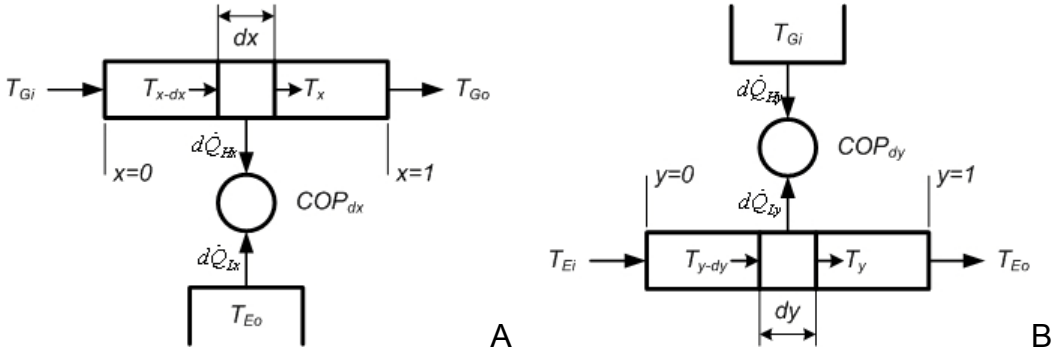

Figure 4-27. COP for a small control volume when A) generator temperature varies and B) Evaporator 1 temperature varies

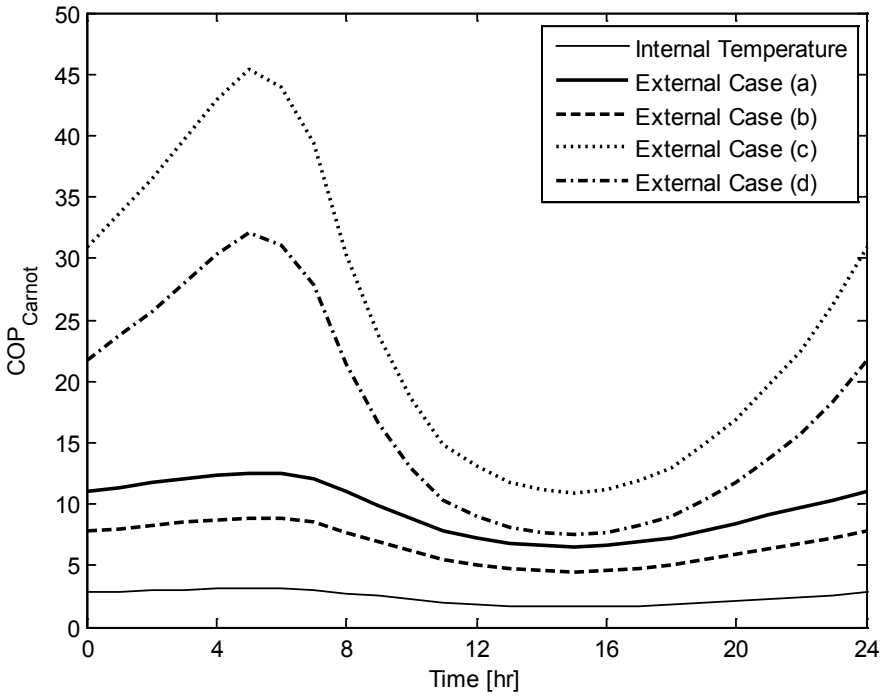

Figure 4-28. Carnot COP for Evaporator 1

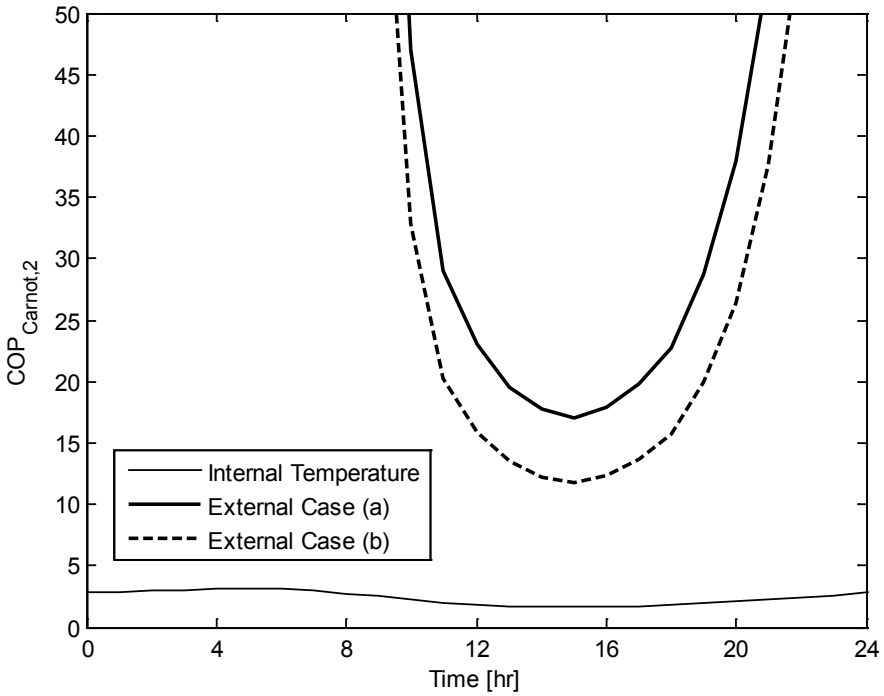

Figure 4-29. Carnot COP for Evaporator 2

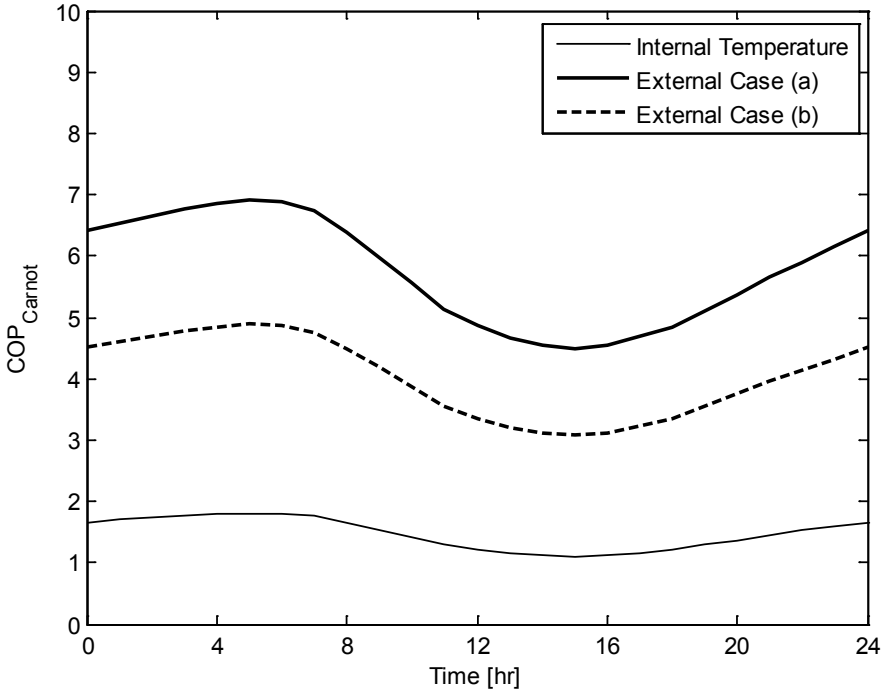

Figure 4-30. Carnot COP for Evaporator 3

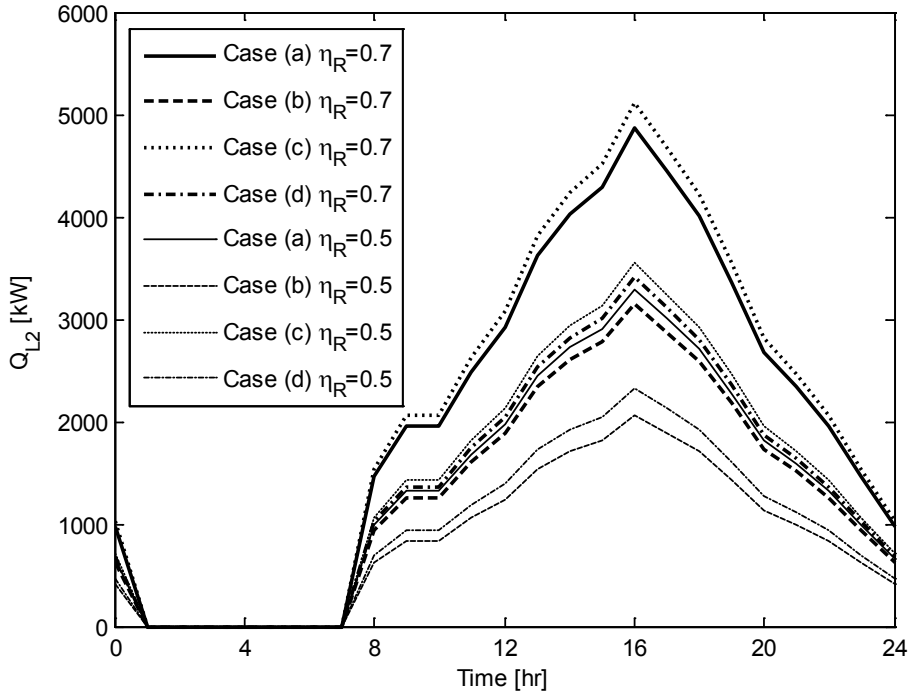

Figure 4-31. Heat removed at Evaporator 2

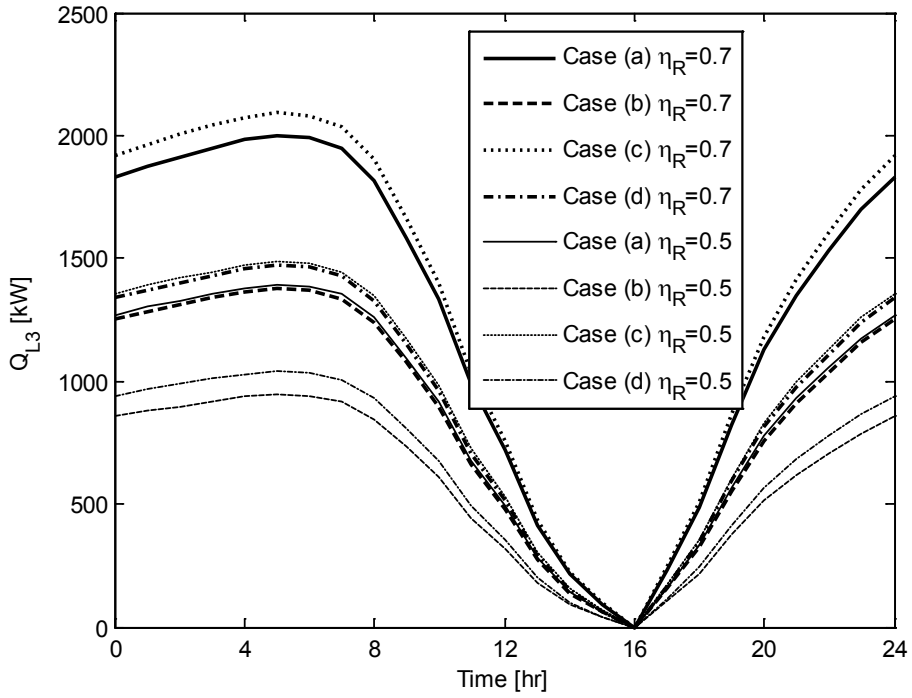

Figure 4-32. Heat removed at Evaporator 3

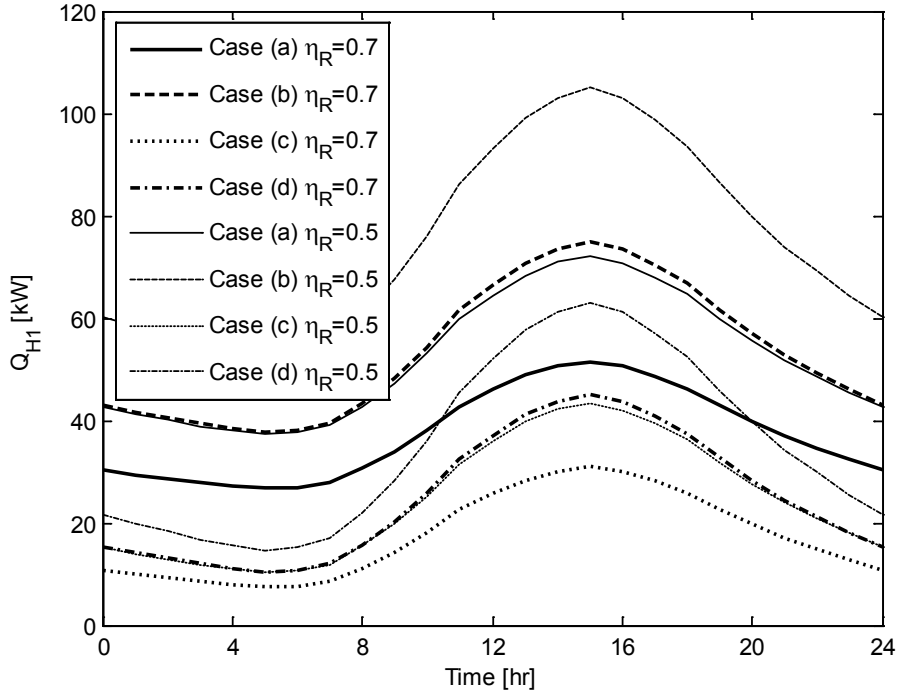

Figure 4-33. Heat required for Sub-system 1

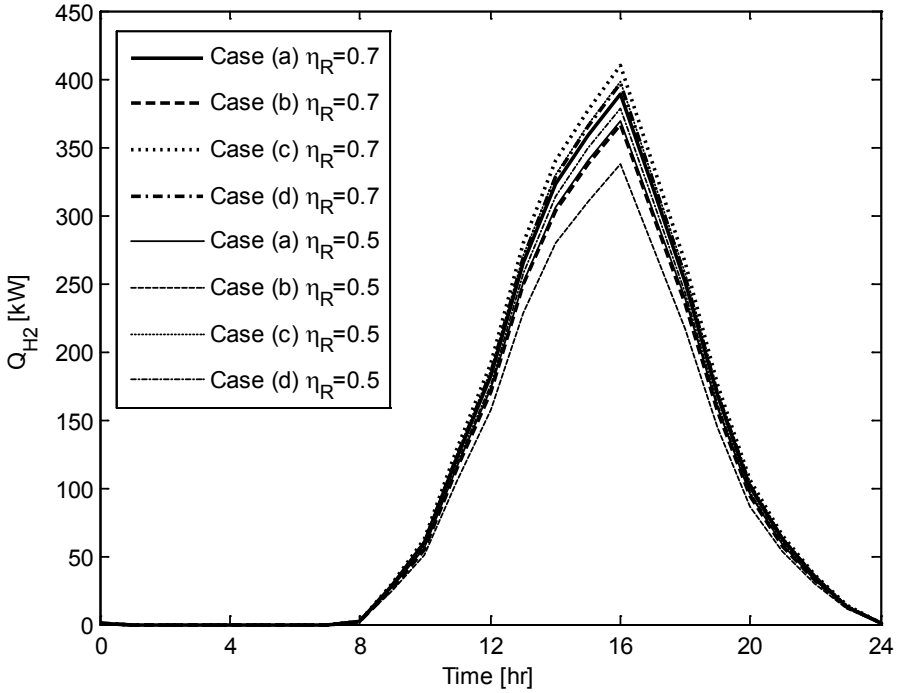

Figure 4-34. Heat required for Sub-system 2

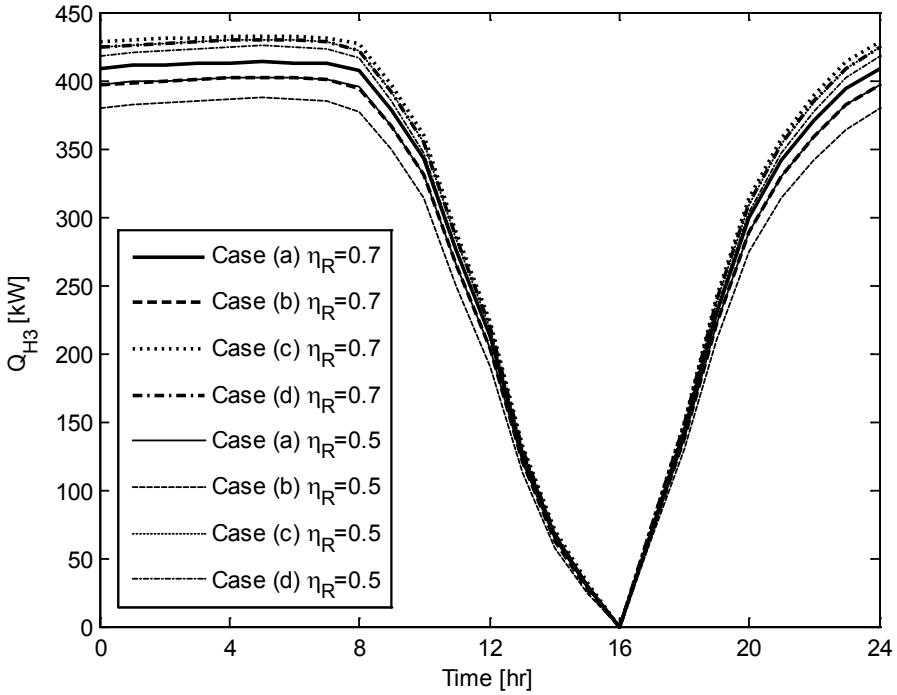

Figure 4-35. Heat required for Sub-system 3

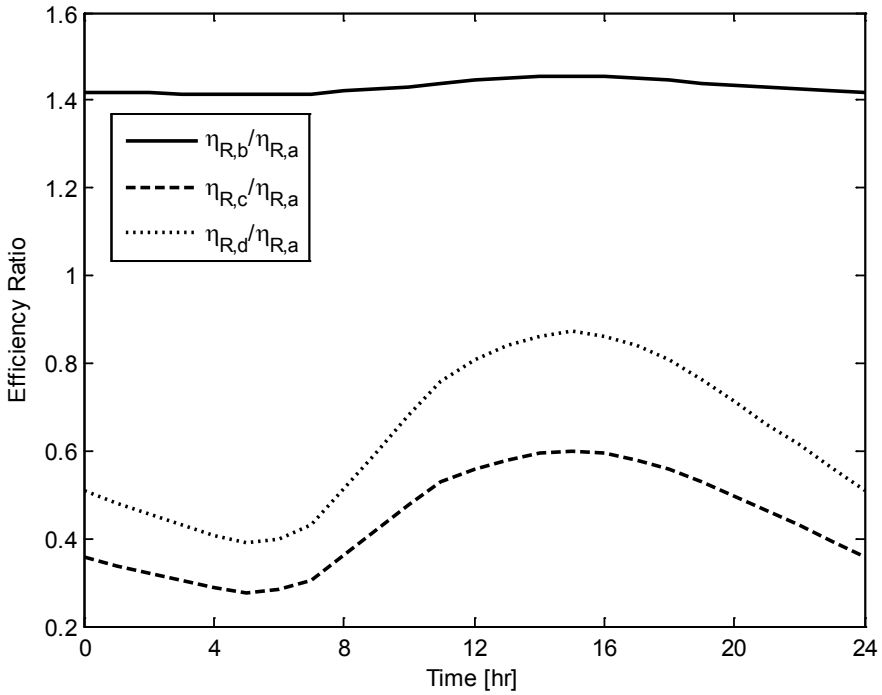

Figure 4-36. Efficiency ratio based on Case (a) efficiency

CHAPTER 5
DYNAMIC MODELING WITH CONSTANT AMMONIA MASS FRACTION APPROACH

As discussed previous section, the potential for the combined cycle PoWER plant in a configuration which allows ice-making for the purpose of load leveling was explored. However, the refrigeration system model used was based on the second law efficiency, which was parametrically varied in order to explore the potential as a function of the technology maturity and complexity in the VARS. In this research, the focus is on system performance using a specific design of a state-of-the-art VARS designed to produce refrigeration at two temperatures. The primary evaporator operates above freezing, cooling the gas path as well as a local air conditioning load. During the summer peak electrical demand in hot climates, the nighttime excess refrigeration capacity is used to produce ice in a closed thermal storage subsystem. The ice would then be available during the afternoon demand peak to provide supplemental air conditioning, thus providing a demand-side management benefit.

Thermodynamic Model

Schematic diagrams of the gas turbine and VARS portion of the PoWER system are shown in Figure 4-1 and Figure 5-1. Air enters the system at State Point 1 and is compressed by the low pressure compressor (LPC). It is adiabatically mixed with the recirculated combustion products from the recuperator at State Point 3. The mixture is then allowed to enter the intercooler (warm gas cooler, WGC) before passing on to the evaporator of the VARS. In the evaporator (EVAP) 1, the mixture is cooled and water produced as a product of combustion is extracted. The mixture then enters the high-pressure core where it is compressed in the high pressure compressor (HPC), heated in the recuperator and combustor (CM), and then expanded in the high pressure turbine

(HPT). After leaving the HPT, all of the combustion gases enter the recuperator where they lose heat to the gases entering the CM. At the exit of the recuperator, a portion of the gas recirculates to join the compressed air stream, and the remaining air is passed through a low pressure turbine (LPT) before exiting to the atmosphere. The recirculating gas passes the generator (hot gas cooler, HGC) of the VARS.

The VARS (Figure 5-1) in this study is an extended ammonia-water single-effect system. Part of the hot gas from the recuperator passes through the heat recovery vapor generator (HRVG), commonly called the generator. Liquid ammonia-water strong solution (State Point 19) from the solution heat exchanger (SHX) is heated and flows to the rectifier (RECT). The latter produces nearly pure ammonia vapor (State Point 1) by using heat from the HRVG and cold strong solution (State Point 16). The almost pure ammonia vapor is cooled in the condenser (COND), which uses chilled water as a coolant, and in the refrigerant heat exchanger (RHX) 1. The liquid phase refrigerant passes through the thermal expansion valve (TXV) 1, then, it flashes into a two-phase fluid at the intermediate pressure. The refrigerant is used both at EVAP1 to cool the HPC inlet and at EVAP2 as an air conditioner during typical summer days. The two-phase refrigerant leaving both EVAP1 and EVAP2 is cold enough to regeneratively cool the refrigerant leaving the condenser (State Point 2). Some refrigerant is further cooled through RHX2 and TXV2. This lowest temperature refrigerant is used to make ice. The return refrigerant, which is high quality or superheated vapor, mixes with strong solution at Absorber (ABS) 1 and ABS2 to produce week solution before entering the pump.

When the air conditioning load is reduced, some refrigerant is further expanded and supplied to EVAP3, which operates as an ice maker. During nighttime, this

evaporator makes ice for use as a coolant for the air conditioner at very high loads when EVAP2 could not meet the load on its own. Almost saturated vapor, but still two-phase refrigerant (State Points 6 and 83) leaves RHX1 and RHX2 to mix with refrigerant at State Points 24 and 114 from the SHX and ABS1, respectively. The saturated liquid strong solution (State Point 141) at the low pressure of ABS2 is pumped and distributed to the rectifier (RECT) and the SHX. The design values of all state points are shown in Table 5-1.

The following are the performance equations for each component considering mass, ammonia mass, and energy balances.

Heat Recovery Vapor Generator

$$\dot{m}_{20} = \dot{m}_{19} = \frac{\dot{Q}_{HRVG}}{q_{HRVG}} \tag{5-1}$$

$$\dot{m}_{20} c_{20} = \dot{m}_{19} c_{19} \tag{5-2}$$

$$\dot{Q}_{HRVG} = \dot{m}_{20} (h_{20} - h_{19}) \tag{5-3}$$

The HRVG is the only component that receives waste heat directly from the gas turbine system. Once $\dot{Q}_{HRVG}$ [kW], the amount of heat input from the gas turbine system, is known, the mass flow rate of the solution can be calculated based on q_{HRVG}, the specific heat transfer.

Rectifier

$$\dot{m}_1 + \dot{m}_{22} = \dot{m}_{16} + \dot{m}_{20} \tag{5-4}$$

$$\dot{m}_1 c_1 + \dot{m}_{22} c_{22} = \dot{m}_{16} c_{16} + \dot{m}_{20} c_{20} \tag{5-5}$$

$$\dot{m}_1 h_1 + \dot{m}_{22} h_{22} = \dot{m}_{16} h_{16} + \dot{m}_{20} h_{20} \tag{5-6}$$

Condenser, Absorber, and Solution Heat Exchanger

The COND, ABS1, and ABS2 use chilled water as a coolant. Even though the mechanical structure of typical absorbers is different from that of typical condensers, simple heat exchangers are applied for ABS1 and ABS2 by using mixing junctions before the components, yielding an equivalent thermodynamic process. The variations of the properties of State Points 18 and 22 are small, given the assumption that the operation of the VARS is well controlled at those points. So, these four components are considered as simple steady-state heat exchangers.

$$\dot{m}_i = \dot{m}_o \tag{5-7}$$

$$\dot{m}_i c_i = \dot{m}_o c_o \tag{5-8}$$

$$\dot{Q}_j = \dot{m}_i (h_i - h_o) = \dot{m}_{w,i} c_{p,H_2O} (T_{w,o} - T_{w,i}) \tag{5-9}$$

$$\dot{Q}_{SHX} = \dot{m}_{18} (h_{19} - h_{18}) = \dot{m}_{22} (h_{22} - h_{23}) \tag{5-10}$$

where the subscript i is 1, 13, 18, 22, 131, and the subscript o is 2, 14, 19, 23, 141 as input and output of the fluids, respectively. The subscript j indicates the components, COND, ABS1, and ABS2. The subscripts w,i and w,o are chilled water input and output, and these represent 7, 11, 111 as inputs, and 8, 12, 121 as outputs. The value of the specific heat of water is taken as $C_{p,H2O}$ = 4.184 [kJ/kg·K]

Evaporator

$$\dot{m}_i = \dot{m}_o \tag{5-11}$$

$$\dot{m}_i c_i = \dot{m}_o c_o \tag{5-12}$$

$$\dot{Q}_j = \dot{m}_i (h_o - h_i) \tag{5-13}$$

where the subscript i is 41, 42, 63, and the subscript o is 51, 52, 73. The subscript j indicates evaporators, EVAP1, EVAP2, and EVAP3.

Pump

Because State Points 141 and 15 are always liquid, densities and specific volumes at those points can be considered constant. This gives the pump work as

$$\dot{W}_P = \frac{\dot{m}_{141} v_{141} (P_{15} - P_{141})}{\eta_P} \tag{5-14}$$

$$\dot{W}_P = \dot{m}_{141} (h_{15} - h_{141}) \tag{5-15}$$

where η_p is the assumed efficiency of the pump.

Thermal Expansion Valve

There are four TXVs to decrease pressure of the fluids in VARS. For an ideal expansion valve the following holds:

$$\dot{m}_i = \dot{m}_o \tag{5-16}$$

$$c_i = c_o \tag{5-17}$$

$$h_i = h_o \tag{5-18}$$

The heading above shows that if you have a subheading of a certain level, you must have more than one. The rationale is that you cannot have a list of only one item.

Refrigerant Heat Exchangers

One of the assumptions in this study is that the system is well controlled. Therefore, steady-state analysis can be reasonably used for most components. However, dynamic analysis is also needed for RHX1 and RHX2 because as the mass flow rate at State Points 5 and 73 changes based on the air conditioning load, the

temperatures of the refrigerant leaving the RHXs will change. While a quasi-steady analysis may serve here for some purposes, we choose to lay the foundation for control algorithm development in future studies.

General conservation equations [42] may be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{5-19}$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = \rho \vec{f} + \nabla \cdot \sigma \tag{5-20}$$

where $\vec{f}$ and σ are body force vector and stress tensor, respectively.

$$\frac{\partial \rho \left(e + \frac{\vec{u} \cdot \vec{u}}{2} \right)}{\partial t} + \nabla \cdot \left\{ \rho \vec{u} \left(h + \frac{\vec{u} \cdot \vec{u}}{2} \right) \right\} = -\nabla \cdot \vec{q} + \rho \vec{f} \cdot \vec{u} + \nabla \cdot (\tau \cdot \vec{u}) + \dot{Q} \tag{5-21}$$

where $\vec{q}$ is the heat flux vector.

To simplify modeling, some reasonable assumptions are applied. The first well-used approximation is the one dimensional approach. This leads the velocity vector to be a scalar quantity. Spatial variations of pressure relating to body forces and viscous dissipation will be neglected. So, there is no need to consider the momentum equation in the analysis. Since convection effect is much greater than conduction, axial conduction will be neglected. Considering a simple shell and tube heat exchanger (Figure 5-2) with the above assumptions, and multiplying by the cross-sectional areas, we obtain

$$\frac{\partial (\rho A)_i}{\partial t} + \frac{\partial \dot{m}_i}{\partial x} = 0 \tag{5-22}$$

$$\frac{\partial(\rho Ah)_i}{\partial t}+\frac{\partial(\dot{m}h)_i}{\partial x}+H_i\pi D_i(T_i-T_w)=0 \tag{5-23}$$

where i is t for tube or s for shell. H_t and H_s are the heat transfer coefficient of the working fluids in the tube and shell, respectively.

As a first approximation, spatial temperature gradient of the wall is ignored as shown in Figure 5-3.

$$(\rho AC_p)_w\frac{\partial T_w}{\partial t}=H_t\pi D_t(T_t-T_w)+H_s\pi D_s(T_s-T_w) \tag{5-24}$$

$$A_w=\frac{V_w}{\partial x}=\frac{\pi}{4}(D_s{}^2-D_t{}^2) \tag{5-25}$$

For simplicity, a lumped heat exchanger model is used with the assumption that the working fluid is well mixed. Therefore, the properties of the working fluid in a control volume are the same as those at its outlet. Because analytical methods can be used only in special cases, a numerical approach is illustrated in this research [43]. A numerical approach always requires that either x, t or both x and t be discretized. Here both x and t are discretized to apply a trial and error approach, yielding the following for RHX1.

$$\dot{m}_3=\dot{m}_2+\frac{A_{t1}\cdot L_1}{\Delta t}\{\rho_3(0)-\rho_3\} \tag{5-26}$$

$$T_{w1}=T_3+\frac{\dot{m}_2h_2-\dot{m}_3h_3}{H_{t1}\pi D_{t1}L_1}+\frac{A_{t1}\{\rho_3h_3-\rho_3(0)h_3(0)\}}{H_{t1}\pi D_{t1}\Delta t} \tag{5-27}$$

$$T_6=T_{w1}+\frac{H_{t1}\pi D_{t1}(T_3-T_{w1})}{H_{s1}\pi D_{s1}}+\frac{(\rho AC_P)_{w1}}{H_{s1}\pi D_{s1}\Delta t}\{T_{w1}-T_{w1}(0)\} \tag{5-28}$$

$$\dot{m}_6 = \dot{m}_5 + \frac{A_{s1} \cdot L_1}{\Delta t}\{\rho_6(0) - \rho_6\} \tag{5-29}$$

$$T_6 = T_{w1} + \frac{\dot{m}_5 h_5 - \dot{m}_6 h_6}{H_{s1}\pi D_{s1} L_1} + \frac{A_{s1}\{\rho_6(0)h_6(0) - \rho_6 h_6\}}{H_{s1}\pi D_{s1}\Delta t} \tag{5-30}$$

For RHX2, the subscripts 2, 3, 5, 6, t1, s1, and w1 can be simply replaced to 43, 53, 73, 83, t2, s2, and w2 from above equations and their control volumes are shown at Figure 5-4.

Ice Maker and Storage

Heat gain to the ice storage unit is considered to be a part of the cooling load in this work. When the air conditioning load is less than or equal to the maximum cooling capacity of EVAP2, the ice maker produces ice while the accumulated ice melts when the air conditioning load exceeds the maximum cooling capacity of EVAP2. And the rate of producing ice is

$$\frac{dm_{ice}}{dt} = \frac{\dot{Q}_{EVAP3} - \dot{Q}_{AC,ice}}{q_{latent}} \tag{5-31}$$

where m_{ice} is the mass of ice [kg] and q_{latent} is latent heat of fusion of water [kJ/kg]. $\dot{Q}_{EVAP3}$ is heat transfer rate of EVAP3 and $\dot{Q}_{AC,ice}$ is additional air conditioning load, which exceeds the maximum cooling capacity of EVAP2 during daytime.

Solution Method

A computational model described previously by Khan et al. [30] was written in FORTRAN to simulate the performance of the PoWER system using traditional cycle analysis methods. The model is capable of calculating engine performance (net work output, water extraction flow rate, and thermal efficiency) as a function of the input

parameters (turbine inlet temperature, recuperator inlet temperature, generator temperature, evaporator exit temperature, low-pressure compressor ratio, turbomachinery efficiencies, heat exchanger effectiveness, equivalence ratio, fuel type, and pressure drops). The input data to the model were appropriate for a small-to-medium size gas turbine system. To fix the operating point in the gas path, the combined cooling capacity of the HGC (HRVG), the WGC (intercooler), and the CGC (EVAP1) has been held constant, even though the individual cooling capacity of each component is allowed to change.

For the VARS and air conditioner, a newly developed transient computational model was written in Simulink as shown in Figure 5-5. This model uses typical summer air conditioning hourly load data as the main input (Figure 5-6) [33] and assigns the required cooling quantities to EVAP2 and EVAP3. It also calculates the mass of ice storage at each time step. To determine the properties of every state point of the VARS, the equations presented in the previous section were used. From the discretized dynamic equation, two T6s for RHX1, and two T_{83}s for RHX2 are closely matched by using a trial and error approach. The values of the mass flow rate, temperature, pressure, concentration, enthalpy, entropy, specific volume, and quality at each state point were then found as a function of time.

During the simulation, some properties (temperature, pressure, concentration, and enthalpy) of the ammonia-water mixture were calculated by using the method described by Pátek and Klomfar [6]. Also, the method described by Ibrahim and Klein [16] was used for calculating entropy and specific volume (or density).

Results and Discussion

Two VARS modes were considered, the higher air conditioning load mode (maximum EVAP2 cooling capacity), and the lower air conditioning load mode (maximum ice making mode). During daytime and early night (11:00AM ~ 9:00PM), the remaining refrigerant (50% of mass at State Point 41) flows through the valve at State Point 42 to EVAP2 while the valve at State Point 43 is fully closed. At the same time, the stored ice helps EVAP2 to meet the required air conditioning load. The data at each state point during this mode are shown at Table 5-2. Saturated liquid refrigerant goes to both EVAP1 and EVAP2, two-phase refrigerant cools down the fluid from COND at RHX1. Finally, this refrigerant, whose temperature is still lower than that of fluid from COND (State Point 2), becomes high quality two-phase flow at the end of the RHX1 (State Point 6). During this mode, the ice maker stops making ice and the ice is melted to produce additional coolant to accommodate the additional air conditioning load (see Figure 5-7).

When the air conditioning load is minimum (3:00AM ~ 6:00AM), the values of each state point are shown at Table 5-3. At this time, the mass flow rate of refrigerant to EVAP2 is smaller than that of EVAP3. For this reason, the cooling effect at RHX1 is reduced, and the temperature of State Point 3 is higher than that of during daytime. Even though the VARS cannot produce fully saturated liquid after the TXV, low quality two-phase refrigerant has almost the same temperature in both EVAP1 and EVAP2 because the state of the refrigerant at the exit of the two evaporators still has a low quality. This two-phase refrigerant has enough cooling capacity at RHX1 to operate the VARS appropriately, and at State Point 6 this refrigerant is either saturated vapor or

superheated vapor. Notice that the temperature of State Point 6 is lower than the temperature of State Point 2.

Similarly, saturated liquid or sub-cooled liquid refrigerant leaving RHX2 is in the two-phase or saturated liquid regime before EVAP3. This fluid leaves as a high quality two-phase fluid at the exit of EVAP3 (State Point 73) and as saturated vapor at the exit of RHX2 (State Point 83).

The COP of the lower air conditioning mode is slightly higher than that of the other mode. Similarly, the differences of all other quantities except those of EVAP2, EVAP3, and ABS1, ABS2 for both modes are negligible. It is noticed that the sum of $\dot{Q}_{HRVG}$, $\dot{Q}_{EVAP1}$, $\dot{Q}_{EVAP2}$, and $\dot{Q}_{EVAP3}$ is approximately equal to the sum of $\dot{Q}_{ABS1}$, $\dot{Q}_{ABS2}$, and $\dot{Q}_{COND}$ for both modes with under 1% error. This result shows the significant role that the VARS coolant plays. As shown in Figure 5-8, 12.99 kg/s of chilled water is supplied to the COND, ABS1, and ABS2. If ambient temperature air or water were used as the coolant at the condenser and absorbers instead, high capacity compressors or pumps would have been needed, and it would not have been appropriate to neglect this power loss.

In Figure 5-9 and Figure 5-10, the heat gained from the surroundings at the three evaporators and their COPs as functions of time are shown. Both figures have similar trends as expected. Because the effect of the outdoor temperature on the PoWER system could be ignored due to the temperature control provided by the VARS, the EVAP1 cooling capacity and COP were unchanged during any one day. As mentioned above, during the daytime, all refrigerant flows to EVAP1 and EVAP2, so the COP of EVAP2 has the highest values at that time. Except for that time, the air conditioning load

changes with time. Therefore, the heat gain at EVAP2 and EVAP3 and their COPs change as functions of time.

These variations make the properties of refrigerant before each evaporator change. Figures 5-11 and 5-12 show the temperature differences at RHX1 and RHX2. Because the VARS in this paper is assumed to be well controlled, the temperature of the fluid leaving the COND is always fixed as shown at Figure 5-11. During late nighttime, T_3 is higher than that of the daytime because the mass flow rate at State Point 5 is decreased. However, the state of the fluid at State Point 5 before the RHX1 is saturated and not very high quality, so, T_5 is almost constant all day. Unlike T_5, T_6 changes because the ammonia-water mixture at State Point 6 has high quality even though it is saturated or because it is vapor.

RHX2 behaves in a similar manner. The temperature difference between T_{43} and T_{73} (the input temperatures of RHX2) is smaller than that of RHX1. Some daytime data are meaningless because EVAP3 is not used at that time as shown in Figure 5-12. EVAP3 turns on around 10:00PM, and because there is no flow at State Point 73 at that moment, T_{53} increases instantaneously and has a higher value than the design value.

Consistency of First Law and Second Law Models

The results from the cycle analysis in this chapter are expected to agree exactly with the results from the Second Law analysis in Chapter 4. To verify this, three cases are considered at 4AM, 14PM, and 24AM. Based on three evaporator temperatures, HRVG temperature (Table 5-1), and ambient temperature (Figure 4-5), Carnot COPs are calculated for each evaporator. Then the second law efficiencies, η_R, are determined by Equation (4-8). When these numbers are used as inputs for the Second Law model

in Chapter 4, we can determine the evaporator heat inputs, $\dot{Q}_{E,i}$, and the refrigeration ratios, R_{evp}, (Equation 4-9) for three different cases. Compared to the constant refrigeration ratio, 0.5, for the cycle analysis in this chapter, calculated refrigeration ratios show the similar results with less than 1% error as shown in Table 5-4. Therefore the Second Law analysis is determined to represent a valid and useful approach for preliminary design, when the details of the VARS have not yet been specified.

Summary

A steady-state gas path model and a dynamic VARS model were used to simulate the thermodynamic performance of the PoWER cooling and power cycle, which combines a semi-closed cycle gas turbine with a VARS in a novel way. Ice storage has been quantified over a typical daily variation in ambient conditions and building air conditioning load in order to show the effect on load leveling. As seen, the potential for reduction of the summer peak capacity requirement is significant.

Several design tradeoffs are possible in optimizing the PoWER system for particular applications. If the HRVG capacity is higher, EVAP1 can be downsized, and the additional heat from the HRVG can be better utilized in other applications, such as air conditioning and/or ice-making. The ice produced was used to meet the excessive air conditioning load at EVAP2 during the daytime. By doing so, part load operation of the total system can be eliminated, potentially providing economic and reliability advantages. The amount of ice available is ideally capable of producing as much as the total daily cooling capacity of EVAP2. This cooling capacity could be used to cool a second building of a size similar to the design building, or be used to downsize the entire system (thus reducing capital costs). If the air conditioning load is limited such as

during a cool day, much more ice can be produced, and it can also be used for other applications.

Considering the combination of attributes mentioned above, it can be concluded that the PoWER system seems to be a promising candidate for distributed power generation, especially in hot climates where the summer load-leveling benefits are important. It is also shown that the Second Law analysis is the most convenient approach and at the same time is effective for preliminary design, accomplished by comparing the results between the conceptual Second Law based model in Chapter 4 and the specific cycle model in this chapter.

Table 5-1. Design values or states at each state point

Parameters	Values and states
State Point 1 temperature [K]	326.254
State Point 1 mass fraction of ammonia [%]	99.8
State Point 1 state	Sat. vapor
Mass fraction; m_1/m_{20}	0.285
State Point 2 state	Sat. liquid
State Point 4 pressure [kPa]	521
EVAP1 specific heat gained [kJ/kg]	1100
EVAP1 temperature [K]	278.150
EVAP2 specific heat gained [kJ/kg]	1100
EVAP2 temperature [K]	278.150
EVAP3 specific heat gained [kJ/kg]	1110
EVAP3 temperature [K]	268.150
Mass fraction; $(m_{42} + m_{43}) / m_{41}$	0.5
State Point 63 pressure [kPa]	268.107
State Point 24 pressure [kPa]	500.368
State Point 114 pressure [kPa]	342.823
State Point 141 temperature [K]	303.256
State Point 15 pressure [kPa]	1792.637
Mass fraction; m_{16} / m_{18}	0.782
State Point 19 temperature [K]	365.070
HRVG specific heat gained [kJ/kg]	440
HRVG temperature [K]	373.15
Supplied chilled water temperature (7, 11, 111) [K]	290
State Point 8 temperature [K]	310
State Point 12, 121 temperature [K]	300
Pump efficiency	0.5
Pressure drop at each heat exchanger [%]	2
Pressure drop at EVAP 3 [%]	1

Table 5-2. Results for higher air conditioning load

State Point	Mass [kg/s]	Temperature [K]	Pressure [kPa]	NH_3 mass fraction	State
1	0.1888	326.2542	1723.6893	0.998	Sat. vap.
2	0.1888	316.0315	1689.2155	0.998	Sat. liq.
3	0.1888	298.8792	1655.4312	0.998	Liquid
4	0.1888	278.1543	521	0.998	Sat. liq.
5	0.1888	279.0330	510.5800	0.998	2-phase
6	0.1888	292.9011	500.3684	0.998	2-phase
13	1.1788	337.9796	500.3684	0.4381	2-phase
14	1.1788	314.9600	490.3610	0.4381	Sat. liq.
15	1.1788	304.0197	1792.6368	0.4381	Liquid
16	0.5171	304.0197	1792.6368	0.4381	Liquid
18	0.6617	304.0197	1792.6368	0.4381	Liquid
19	0.6617	365.0700	1756.7841	0.4381	Liquid
20	0.6617	408.0452	1721.6484	0.4381	2-phase
22	0.9900	376.2403	1723.6893	0.3313	Liquid
23	0.9900	335.3104	1689.2155	0.3313	Liquid
24	0.9900	335.3104	500.3684	0.3313	Liquid
41	0.1259	278.1543	521	0.998	Sat. liq.
42	0.0629	278.1543	521	0.998	Sat. liq.
43	0	-	-	-	-
51	0.1259	279.0330	510.5800	0.998	2-phase
52	0.0629	279.0330	510.5800	0.998	2-phase
53	0	-	-	-	-
63	0	-	-	-	-
73	0	-	-	-	-
83	0	-	-	-	-
114	1.1788	311.2367	343.8232	0.4381	2-phase
131	1.1788	311.4770	347.3316	0.4381	2-phase
141	1.1788	303.2563	340.3850	0.4381	Sat. liq.

$\dot{Q}_{HRVG}$ [kW]	440	$\dot{Q}_{COND}$ [kW]	215.6486	$\dot{Q}_{ABS1}$ [kW]	373.6421
$\dot{Q}_{EVAP1}$ [kW]	138.4798	$\dot{Q}_{EVAP2}$ [kW]	69.2399	$\dot{Q}_{EVAP3}$ [kW]	0
$\dot{W}_P$ [kW]	4.0499	$\dot{Q}_{SHX}$ [kW]	190.6864	$\dot{Q}_{ABS2}$ [kW]	62.4789
COP_1	0.3147	COP_2	0.1574	COP_3	0
COP_{total}	0.4721				

Table 5-3. Results for lower air conditioning load

State Point	Mass [kg/s]	Temperature [K]	Pressure [kPa]	NH_3 mass fraction	State
1	0.1888	326.2542	1723.6893	0.998	Sat. vap.
2	0.1888	316.0315	1689.2155	0.998	Sat. liq.
3	0.1888	303.8135	1655.4312	0.998	Liquid
4	0.1888	278.1558	521	0.998	2-phase
5	0.1888	280.0389	510.5800	0.998	2-phase
6	0.1399	299.5383	500.3684	0.998	Sat. vap.
13	1.1299	340.8979	500.3684	0.3981	2-phase
14	1.1299	314.9600	490.3610	0.3981	Liquid
15	1.1788	304.0169	1792.6368	0.4230	Liquid
16	0.5171	304.0169	1792.6368	0.4230	Liquid
18	0.6617	304.0169	1792.6368	0.4230	Liquid
19	0.6617	365.0700	1756.7841	0.4230	Liquid
20	0.6617	409.9966	1721.6484	0.4230	2-phase
22	0.9900	375.3206	1723.6893	0.3313	Liquid
23	0.9900	334.1742	1689.2155	0.3313	Liquid
24	0.9900	334.1742	500.3684	0.3313	Liquid
41	0.1259	278.1558	521	0.998	2-phase
42	0.0140	278.1558	521	0.998	2-phase
43	0.0489	278.1558	521	0.998	2-phase
51	0.1259	280.0389	510.5800	0.998	2-phase
52	0.0140	280.0389	510.5800	0.998	2-phase
53	0.0489	277.5856	510.5800	0.998	Sat. liq.
63	0.0489	268.0984	358	0.998	2-phase
73	0.0489	268.4098	354.42	0.998	2-phase
83	0.0489	272.5600	347.3316	0.998	Sat. vap.
114	1.1299	314.9600	343.8232	0.3981	Liquid
131	1.1788	316.4753	347.3316	0.4230	2-phase
141	1.1788	303.2563	340.3850	0.4230	Liquid

$\dot{Q}_{HRVG}$ [kW]	440	$\dot{Q}_{COND}$ [kW]	215.6486	$\dot{Q}_{ABS1}$ [kW]	305.5540
$\dot{Q}_{EVAP1}$ [kW]	138.4798	$\dot{Q}_{EVAP2}$ [kW]	15.4157	$\dot{Q}_{EVAP3}$ [kW]	54.3135
$\dot{W}_P$ [kW]	4.0250	$\dot{Q}_{SHX}$ [kW]	189.8940	$\dot{Q}_{ABS2}$ [kW]	131.0223
COP_1	0.3147	COP_2	0.035	COP_3	0.1234
COP_{total}	0.4732				

Table 5-4. Consistency of the results between Chapters 4 and 5.

Time		4AM			14PM			24PM	
Evaporator	1	2	3	1	2	3	1	2	3
COP	0.315	0.035	0.123	0.315	0.157	0	0.315	0.070	0.088
COP_{carnot}	3.073	3.073	1.925	1.643	1.643	1.180	2.778	2.778	1.788
η_R	0.102	0.011	0.064	0.192	0.096	0	0.113	0.025	0.049
Heat input	138.5	15.42	54.31	138.5	69.24	0	138.5	30.83	38.76
R_{evp}		0.503			0.500			0.502	

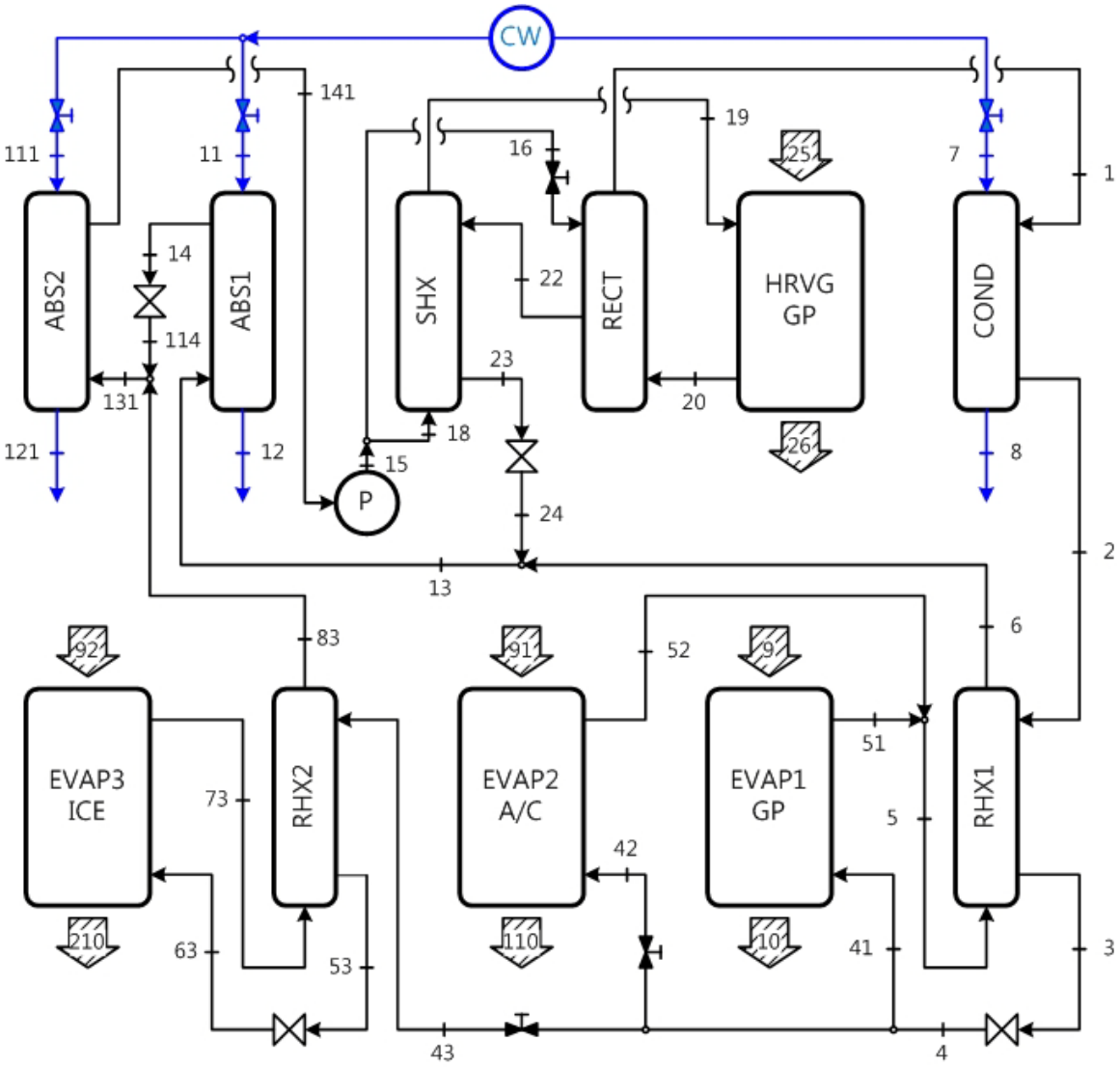

Figure 5-1. Extended vapor absorption refrigeration system of the PoWER system

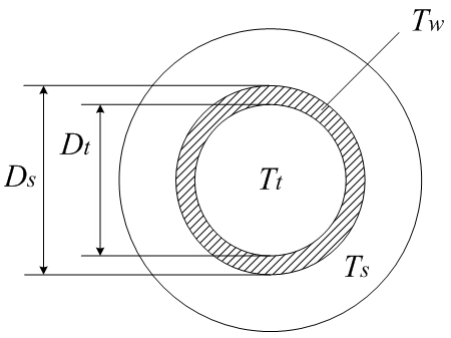

Figure 5-2. Simple shell and tube heat exchanger

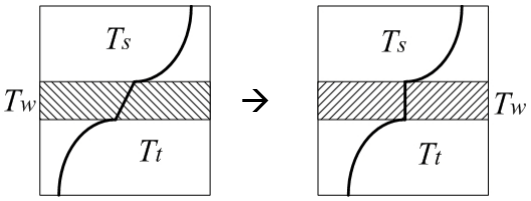

Figure 5-3. Lumped wall model

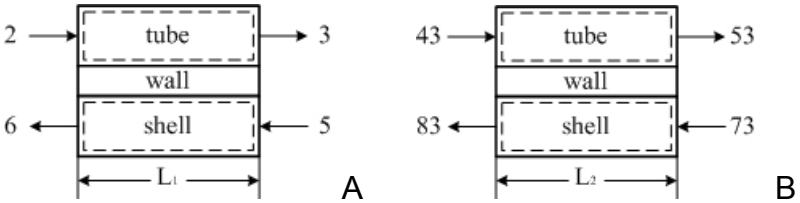

Figure 5-4. Control volumes for A) RHX1 and B) RHX2

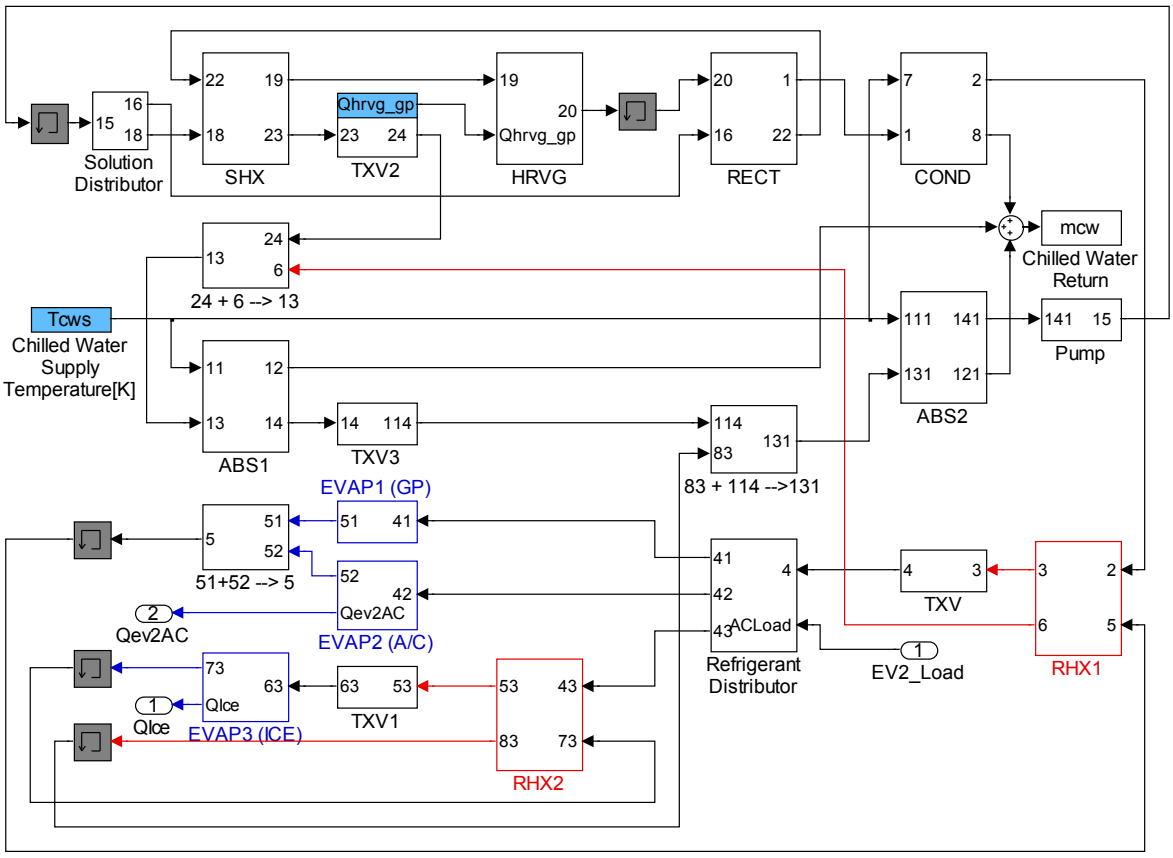

Figure 5-5. Extended VARS Simulink model

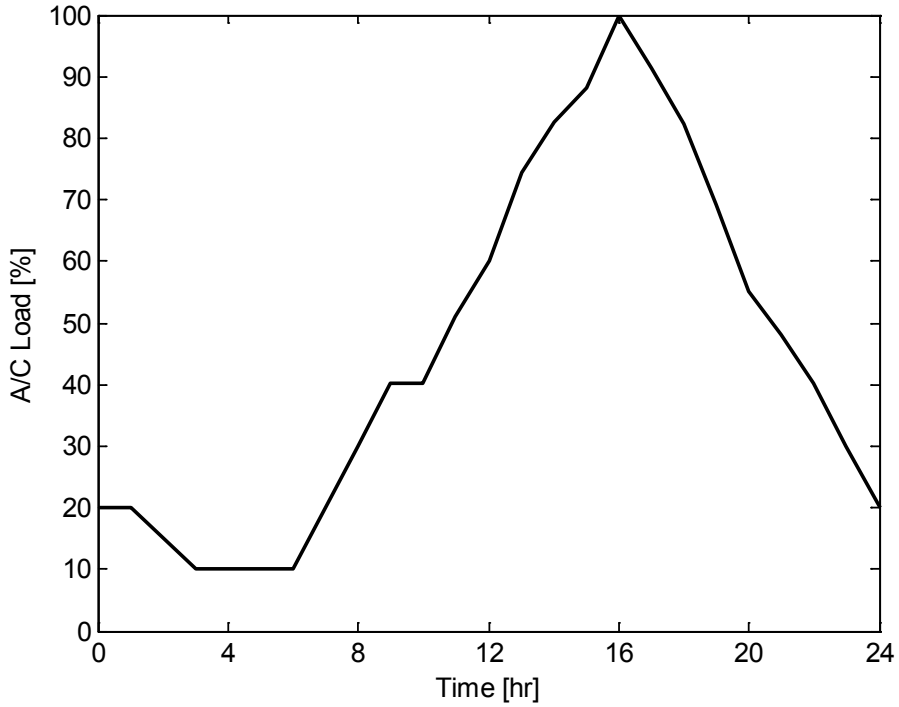

Figure 5-6. Air conditioning hourly load

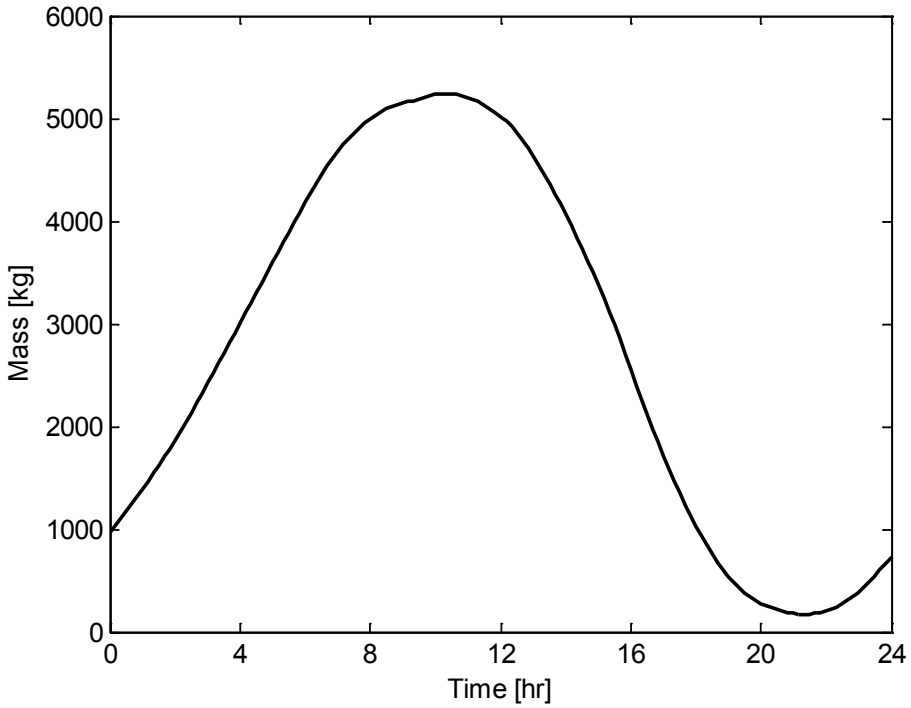

Figure 5-7. Mass change of the stored ice [kg]

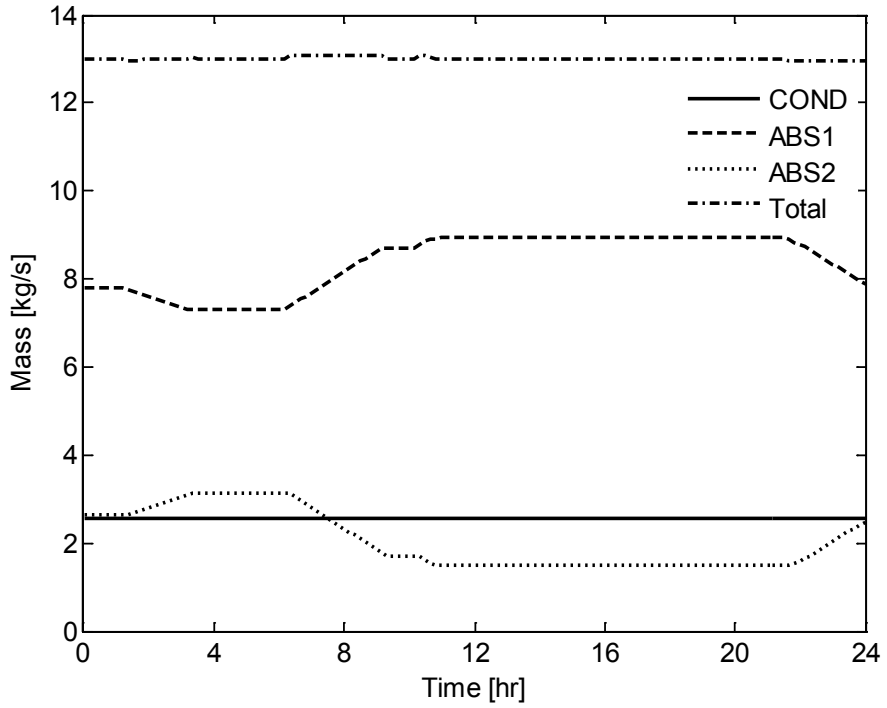

Figure 5-8. Chilled water use [kg/s]

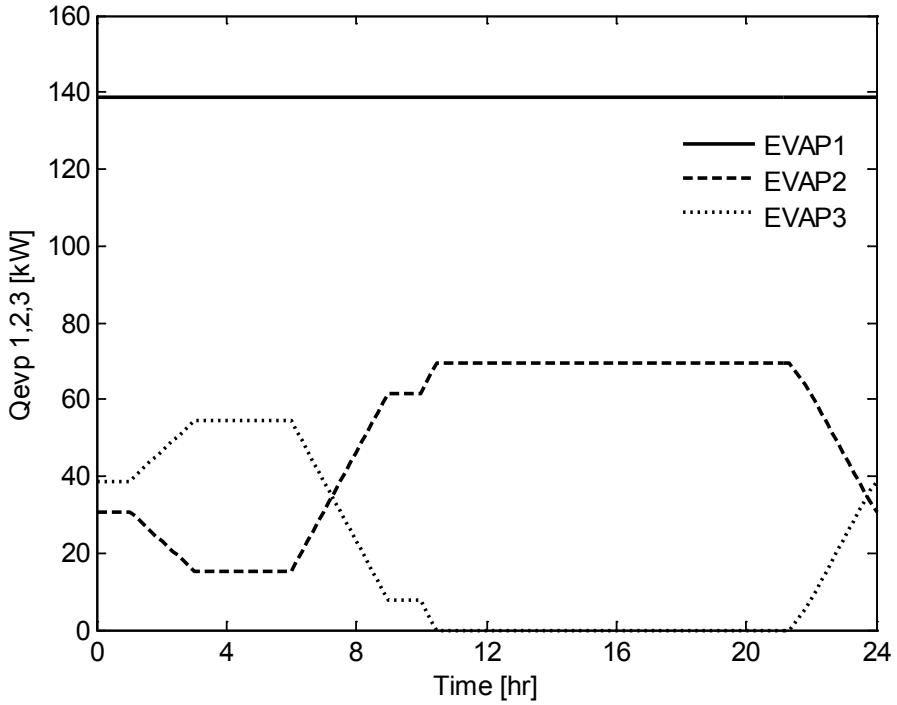

Figure 5-9. Heat gained from surroundings at three evaporators

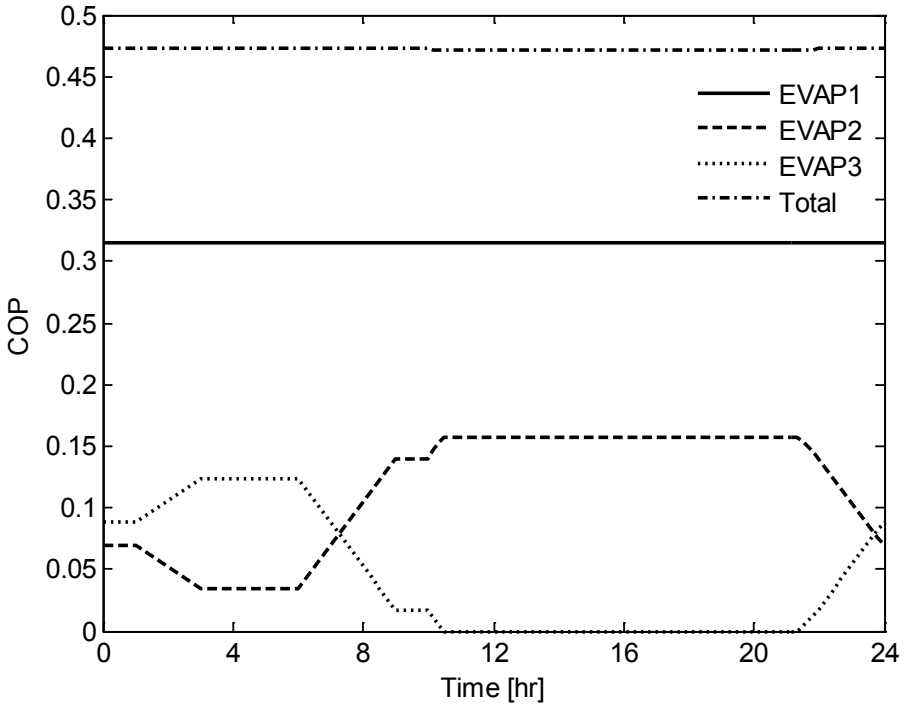

Figure 5-10. Coefficient of performance (COP)

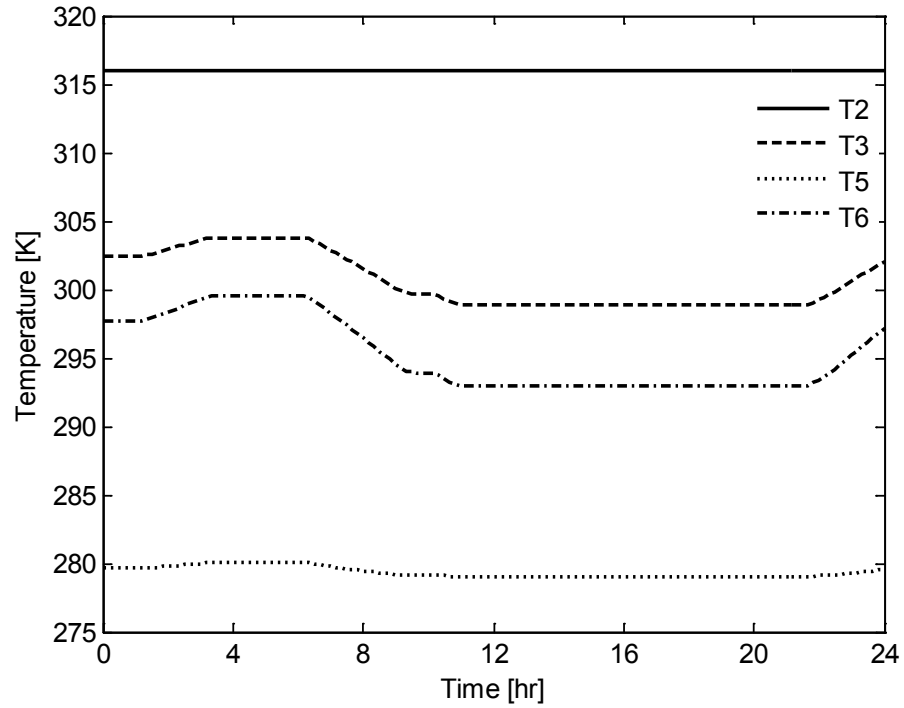

Figure 5-11. Temperature of RHX1

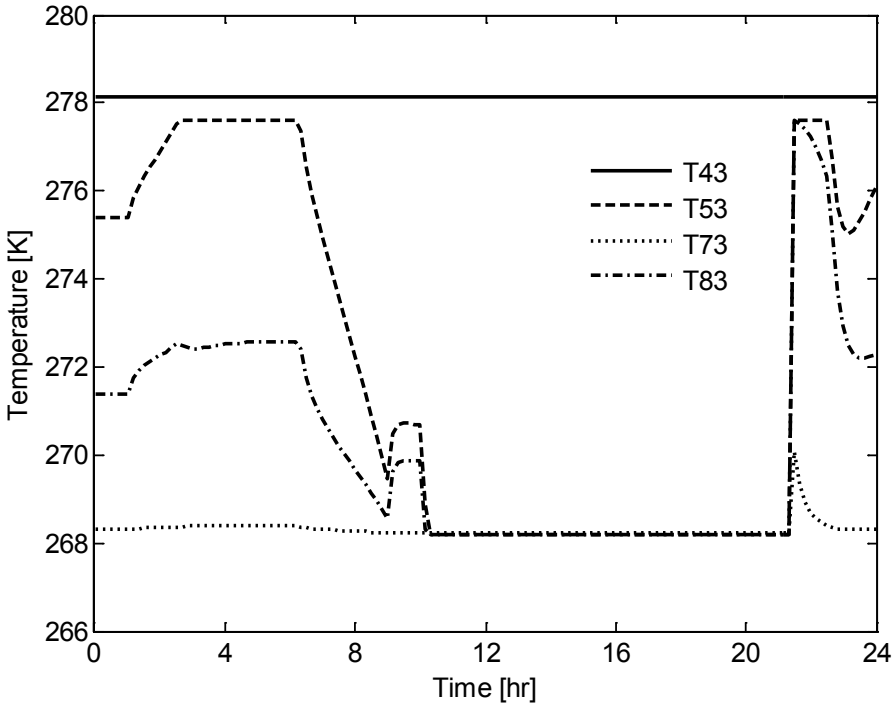

Figure 5-12. Temperature of RHX2

CHAPTER 6
DYNAMIC MODELING FOR TWO-COMPONENT FLUIDS

A dynamic model of a special case, constant ammonia mass fraction, has been presented in the previous chapter. Even though the fluids in the VARS dynamic model are two-component (ammonia-water), they were treated as single-component fluids in Chapter 5. Further, the time-discretized dynamic model has the limitation that it cannot be applied to conventional control theory. In this chapter, a dynamic model is derived to satisfy both approximate two-phase two-component fluid physics and the requirements of the intended control application.

Introduction to Dynamic Heat Exchanger Modeling

There are various heat exchanger mathematical models in the literature. Among them, only two-phase dynamic models are discussed and then adapted to meet the goal of this research.

Moving Boundary Model for Single Component Fluid

To model two-phase flow in heat exchangers such as evaporators and condensers, the conservation equations are expressed as partial differential equations (PDEs). Complex PDEs sometimes cannot be solved analytically and are usually difficult to be transformed into ordinary differential equations (ODEs). For heat exchanger applications, PDEs can be spatially discretized, but the discretized models are of high order and do not lend themselves to further model reduction. Therefore, models of this type are not well suited for control design.

A moving boundary model is one of the better alternatives for two phase flow modeling in heat exchangers. For system level modeling, moving boundary models represent the physics of two-phase flow well and satisfy the simplicity requirement.

Although various mathematical model variants are necessary based on boundary conditions, the resulting models are generally fast enough to provide real-time results compared to discretized models and are robust relative to sudden changes in the boundary conditions.

Based on the standard physical laws of compressible fluid mechanics, mass, momentum, and energy balance equations are derived as shown in Chapter 5 (Equations 5-19 to 5-21). To simplify these equations, several assumptions are applied. The heat exchanger is idealized as a long, thin, horizontal tube; the flow is considered one-dimensional; axial heat conduction is neglected. Compared with the total absolute pressure, the pressure drop along the heat exchanger tube due to momentum change in the refrigerant and viscous friction is typically small and is neglected. Therefore, refrigerant pressure can be assumed uniform along the heat exchanger and the momentum equation is no longer necessary. The resulting set of PDEs is as follows:

Mass balance:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial z} = 0 \tag{6-1}$$

Energy balance:

$$\frac{\partial (\rho h - P)}{\partial t} + \frac{\partial (\rho u h)}{\partial z} = \frac{4}{D_i} \alpha_i (T_w - T_r) \tag{6-2}$$

In the two-phase section, the refrigerant temperature is at its saturated value with no spatial variation. For the evaporator shown schematically in Figure 6-1, the temperature of the refrigerant in the superheated section increases as it travels toward

the evaporator outlet due to the single-phase region. For this situation including temperature variation, we lump the evaporator dynamics into two nodes: the two-phase node and the superheated node.

The key feature of the conventional moving boundary two-phase model is an assumption of a time-invariant mean void fraction. Wedekind et al. [23] showed that the mean void fraction remains relatively invariant in the two-phase region of a heat exchanger during most regimes of operation. This implies that under different inflow conditions, the liquid dry-out point in an evaporator may change its location along the evaporator; however, the distribution of the liquid and vapor remains similar at all times.

To prepare for control algorithm development, it is necessary to create ODEs from the one-dimensional PDEs (6-1) and (6-2) for both control volumes in Figure 6-1. Following the principle of time-invariant mean void fraction, Equations (6-1) and (6-2) can be integrated from $z=0$ to $l_1(t)$ for two-phase refrigerant. (see Appendix C for derivation in detail).

Mass balance for two-phase region:

$$Al_1 \frac{d\overline{\rho_1}}{dt} + A\overline{\rho_1} \frac{dl_1}{dt} = \dot{m}_i - \dot{m}_{int} \tag{6-3}$$

Energy balance for two-phase region:

$$Al_1 \frac{d\overline{\rho_1 h_1}}{dt} + A\overline{\rho_1 h_1} \frac{dl_1}{dt} - Al_1 \frac{dP}{dt} = \dot{m}_i h_i - \dot{m}_{int} h_g + \alpha_{i1} \pi D_i l_1 \left(T_{w1} - T_{r1}\right) \tag{6-4}$$

where A is the cross-sectional area of heat exchanger tube, l_1 is the length of two-phase region, $\dot{m}_i$ and $\dot{m}_{int}$ are the mass flow rate at heat exchanger inlet and interface,

respectively. h_g is the enthalpy at the interface and this value is same as that of saturated vapor of refrigerant. He et al. [21] used absolute speed for the mass flow rate at the interface. However, the interface between two-phase and vapor fluids moves back and forth, so the mass flow rate of the refrigerant at interface, $\dot{m}_{int}$, is always related to the relative speed (Equation (C-4)). The Equations (6-3) and (6-4) are corrected equations from He et al. [21].

The mass and energy balances for superheated vapor refrigerant can be obtained by integration from $z=l_1(t)$ to L (see Appendix C for derivation).

Mass balance for superheated vapor region:

$$Al_2 \frac{d\overline{\rho_2}}{dt} - A\overline{\rho_2}\frac{dl_1}{dt} = \dot{m}_{int} - \dot{m}_o \tag{6-5}$$

Energy balance for superheated vapor region:

$$Al_2 \frac{d\overline{\rho_2 h_2}}{dt} - A\overline{\rho_2 h_2}\frac{dl_1}{dt} - Al_2\frac{dP}{dt} = \dot{m}_{int}h_g - \dot{m}_o h_o + \alpha_{i2}\pi D_i l_2 \left(T_{w2} - T_{r2}\right) \tag{6-6}$$

where $l_2(t)$ is the length of superheated vapor region, $\dot{m}_o$ is the mass flow rate at heat exchanger exit. h_o is refrigerant exit enthalpy.

The energy equations for the tube wall of control volumes 1 and 2 are

$$\left(C_p \rho A\right)_w \frac{dT_{w1}}{dt} = \alpha_{i1}\pi D_i \left(T_{r1} - T_{w1}\right) + \alpha_o \pi D_o \left(T_a - T_{w1}\right) \tag{6-7}$$

$$\left(C_p \rho A\right)_w \frac{dT_{w2}}{dt} + \frac{T_{w1} - T_{w2}}{l_2}\frac{dl_1}{dt} = \alpha_{i2}\pi D_i \left(T_{r2} - T_{w2}\right) + \alpha_o \pi D_o \left(T_a - T_{w1}\right) \tag{6-8}$$

The mean values of density and enthalpy in two-phase region are functions of pressure only. For superheated vapor refrigerant, however, they are functions of pressure and temperature or the density and the temperature are functions of pressure and enthalpy. It is assumed that the bulk enthalpy of superheated vapor region is half of enthalpies at interface and exit. After cancelling out $\dot{m}_{int}$ (Appendix C), we obtain the final form of conservation equations as matrix form:

$$\dot{\mathbf{x}} = \mathbf{D}^{-1}\mathbf{f}\left(\mathbf{x},\mathbf{u}\right) \tag{6-9}$$

where $\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & 0 & 0 \\ d_{21} & d_{22} & 0 & 0 & 0 \\ d_{31} & d_{32} & d_{33} & 0 & 0 \\ 0 & 0 & 0 & d_{44} & 0 \\ d_{51} & 0 & 0 & 0 & d_{55} \end{bmatrix}$, $\mathbf{f}\left(\mathbf{x},\mathbf{u}\right) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} l_1 \\ P \\ T_{r2} \\ T_{w1} \\ T_{w2} \end{bmatrix}$ and $\mathbf{u} = \begin{bmatrix} \dot{m}_i \\ h_i \\ \dot{m}_o \end{bmatrix}$. $\mathbf{x}$ is the vector of state variables and $\mathbf{u}$ is the vector of inputs (The coefficients of matrices $\mathbf{D}$ and $\mathbf{f}$ are indicated in Appendix C).

Two-Component Fluids

In this research, ammonia-water mixtures are used as a working fluid in vapor absorption refrigeration system rather than a pure component as in a simple vapor compression refrigeration system. Therefore, some extensions are necessary to model two-component two-phase flow in a heat exchanger. First, a new state variable, concentration, is necessary. Because almost pure ammonia is used as a refrigerant, ammonia mass fraction is equivalent to a concentration.

Unlike a pure component flow heat exchanger, the temperature decreases for a condenser and increases for an evaporator in the two-phase region, even at uniform

pressure. Figure 6-2 shows the two-phase flow temperature increase in an evaporator for several different pressures. At the same time, the ammonia mass fraction in the vapor and liquid drops because ammonia is more volatile than water. (Figure 6-3 and 6-4)

Densities and enthalpies for saturated vapor and liquid are functions of pressure, temperature, and ammonia mass fraction, so saturation properties are not uniform along the heat exchanger. The most important change is that the mean void fraction is no longer uniform because ammonia concentration is involved as well. When vapor quality is fixed in two-phase flow, the density ratio determines mean void fraction. Density ratio can be considered as a constant in the pressure range of interest. However, it is affected by ammonia concentration.

Two-Phase Model Options

For control-oriented dynamic modeling of a VARS, the conventional moving boundary two-phase model is extended to include two-component fluids. Species balances are added for the conservation equations. Thermodynamic properties in the two-phase region are no longer uniform even at constant pressure along the heat exchanger. To model this complex heat exchanger using lumped parameters, three approaches were considered: 1. Uniform mass fraction model; 2. Thermodynamic equilibrium model; 3. Spatial mean value model.

For the first two models, both vapor and liquid in the two-phase region are considered to be saturated. It is assumed that bulk ammonia mass fractions of saturated vapor and liquid are the same for uniform mass fraction model while temperature is uniform along the entire heat exchanger for the thermodynamic equilibrium model. At steady state, saturated vapor and saturated liquid have different

bulk temperatures when the uniform mass fraction model is applied. For the thermodynamic equilibrium model, however, saturated vapor and liquid have different ammonia mass fractions. Figure 6-5 shows two very limited cases of the flow state on enthalpy versus ammonia mass fraction (h-x) diagram at fixed pressure.

The h-x diagram is based on Bogart's T-P-x tables [12] and Ibrahim's enthalpy relations [15] at 1,000 kPa. The upper and lower curves are saturated vapor and saturated liquid enthalpy lines, respectively. The dotted lines between the two curves indicate isothermal line in the two-phase region. The solid vertical line is based on the uniform mass fraction model and the inclined solid line is based on thermodynamic equilibrium model.

When average enthalpy, h_{avg}, and bulk ammonia mass fraction, x_b, are the same for the two models shown in the Figure 6-5, the saturated vapor temperature, T_{sg}, is always higher than the thermodynamic equilibrium temperature, T_e, and the saturated liquid temperature, T_{sl}, is lower than that for uniform mass fraction model. Ammonia mass fraction in saturated vapor, x_{eg}, is higher than that of saturated liquid, x_{el}, for thermodynamic equilibrium model. Because of those behaviors, the two models show different vapor qualities and void fractions except for the pure component region. This difference makes it difficult to apply the void fraction model to both models.

To avoid this problem, using spatially averaged values of thermodynamic properties in the two-phase region is the most appropriate method of the three. The spatial mean values are determined based on the assumptions, which are: pressure and ammonia mass fraction are uniform along the heat exchanger, and each local infinitesimal control volume is in thermodynamic equilibrium.

With these assumptions, temperature distribution information is available in the two-phase region (Figure 6-2) and densities and enthalpies can be expressed spatially at fixed pressure and ammonia mass fraction. These spatial distributions of thermodynamic properties can be tabulated or can be curvefit as functions of pressure, ammonia concentration, initial conditions and boundary conditions. However, fitted equations may encounter Runge's phenomenon, which is discussed in Chapter 3. Therefore, tables for thermodynamic property data in the two-phase regime are used in this research.

Thermodynamic Models of VARS for Two-Component Fluids

In this research, a spatially mean value model is used with tabulated two-phase thermodynamic property data to model and solve the conservation equations. This approach represents two-component two-phase flow physics relatively well and meets control theory requirements at the same time. With this approach, some modifications are necessary related to conventional moving boundary models [21, 22]. For the system model in this research, pressure is determined from the system model. Therefore, pressure is an input for all heat exchangers in the VARS. Either inlet or outlet mass flow rate is an output.

From the schematic diagram of the VARS shown in Figure 5-1, there are six different heat exchangers; condenser (COND), absorber (ABS), generator (HRVG), evaporator (EVAP), refrigerant heat exchanger (RHX), and solution heat exchanger (SHX). Among them, RHX has similar fluid condition to that of the EVAP and RHX has not been modeled separately.

Condenser

For general refrigeration applications, high pressure superheated vapor is condensed to sub-cooled liquid in condensers for normal operation modes. Therefore, three different phases can exist in condensers, the superheated, two-phase and sub-cooled regions. However, in this research, only two-phase and sub-cooled regions are considered because theoretically only saturated vapor exits from the top of rectifier when there is no pressure drop in connecting ducts between rectifier and condenser, and modeling of the single phase region is straightforward. Because saturated vapor enters the condenser, the full two-phase region always exists in that device. As a secondary fluid (coolant), sub-cooled liquid is assumed, though this choice does not directly enter the analysis. A schematic of a condenser is shown in Figure 6-6 illustrating the two-phase region (on the left) and the sub-cooled region (on the right). General conservation equations are derived next based on two control volumes.

Basic assumptions used for conventional moving boundary models are applied, so the equations (6-1) and (6-2) are used for mass balance and energy balance, respectively. Heat input, which is the right hand side of Equation (6-2), is expressed as heat flux, $\dot{Q}''$, taken here as approximately constant. For a two-component fluid, a species balance for ammonia concentration is needed as an additional conservation equation:

Species balance:

$$\frac{\partial(\rho x)}{\partial t}+\frac{\partial(\rho u x)}{\partial z}=0 \qquad (6\text{-}10)$$

where x is ammonia mass fraction.

To obtain ODEs of the conservation equations, the above equations are integrated along the control volume length. After applying Leibniz' rule, the following six equations are obtained:

Mass balance for refrigerant in control volume 1:

$$A_c l_{c1} \frac{d\overline{\rho_{c1}}}{dt} + A_c \overline{\rho_{c1}} \frac{dl_{c1}}{dt} = \dot{m}_{ci} - \dot{m}_{c12} \tag{6-11}$$

Species balance for refrigerant in control volume 1:

$$A_c l_{c1} \frac{d\overline{\rho_{c1} x_{c1}}}{dt} + A_c \overline{\rho_{c1} x_{c1}} \frac{dl_{c1}}{dt} = \dot{m}_{ci} x_{ci} - \dot{m}_{c12} x_{c12} \tag{6-12}$$

Energy balance for refrigerant in control volume 1:

$$A_c l_{c1} \frac{d\overline{\rho_{c1} h_{c1}}}{dt} + A_c \overline{\rho_{c1} h_{c1}} \frac{dl_{c1}}{dt} - A_c l_{c1} \frac{dP_c}{dt} = \dot{m}_{ci} h_{ci} - \dot{m}_{c12} h_{c12} + (UD)_{c1} l_{c1} \left(\overline{T_{ca1}} - \overline{T_{cr1}} \right) \tag{6-13}$$

Mass balance for refrigerant in control volume 2:

$$A_c l_{c2} \frac{d\overline{\rho_{c2}}}{dt} - A_c \overline{\rho_{c2}} \frac{dl_{c1}}{dt} = \dot{m}_{c12} - \dot{m}_{co} \tag{6-14}$$

Species balance for refrigerant in control volume 2:

$$A_c l_{c2} \frac{d\overline{\rho_{c2} x_{c2}}}{dt} - A_c \overline{\rho_{c2} x_{c2}} \frac{dl_{c1}}{dt} = \dot{m}_{c12} x_{c12} - \dot{m}_{co} x_{co} \tag{6-15}$$

Energy balance for refrigerant in control volume 2:

$$A_c l_{c2} \frac{d\overline{\rho_{c2} h_{c2}}}{dt} - A_c \overline{\rho_{c2} h_{c2}} \frac{dl_{c1}}{dt} - A_c l_{c2} \frac{dP_c}{dt} = \dot{m}_{c12} h_{c12} - \dot{m}_{co} h_{co} + (UD)_{c2} l_{c2} \left(\overline{T_{ca2}} - \overline{T_{cr2}} \right) \quad (6\text{-}16)$$

where $(UD)_{c1}$ and $(UD)_{c2}$ are universal heat transfer coefficient per length for control volume 1 and 2, respectively.

Six base conservation equations (Equations 6-11 to 6-16) are used for the ammonia-water mixture (refrigerant) side in the condenser and additional conservation equations are necessary for the coolant (secondary fluid). It is assumed that the coolant is incompressible and has a constant heat capacity. Therefore, mass balances for two control volumes are trivial because densities are the constant. Then, only the energy balance remains:

Energy balance for coolant in control volume 1:

$$\left(\rho A C_p \right)_{ca} l_{c1} \frac{d\overline{T_{ca1}}}{dt} = \dot{m}_{ca} C_{pca} \left(T_{ca21} - T_{cao} \right) + (UD)_{c1} l_{c1} \left(\overline{T_{cr1}} - \overline{T_{ca1}} \right) \quad (6\text{-}17)$$

Energy balance for coolant in control volume 2;

$$\left(\rho A C_p \right)_{ca} l_{c2} \frac{d\overline{T_{ca2}}}{dt} = \dot{m}_{ca} C_{pca} \left(T_{cai} - T_{ca21} \right) + (UD)_{c2} l_{c2} \left(\overline{T_{cr2}} - \overline{T_{ca2}} \right) \quad (6\text{-}18)$$

For the refrigerant, mass balances can be substituted for species and energy balances and the outlet mass flow rate is determined by the fluid condition in the condenser at the exit. Then we have four remaining equations:

Species balance for refrigerant in control volume 1:

$$A_c l_{c1} \left(\frac{d\overline{\rho_{c1} x_{c1}}}{dt} - x_{c12} \frac{d\overline{\rho_{c1}}}{dt} \right) + A_c \left(\overline{\rho_{c1} x_{c1}} - \overline{\rho_{c1}} x_{c12} \right) \frac{dl_{c1}}{dt} = \dot{m}_{ci} \left(x_{ci} - x_{c12} \right) \quad (6\text{-}19)$$

Energy balance for refrigerant in control volume 1:

$$A_c l_{c1}\left(\frac{d\overline{\rho_{c1}h_{c1}}}{dt}-h_{c12}\frac{d\overline{\rho_{c1}}}{dt}-\frac{dP_c}{dt}\right)+A_c\left(\overline{\rho_{c1}h_{c1}}-\overline{\rho_{c1}}h_{c12}\right)\frac{dl_{c1}}{dt}$$
$$=\dot{m}_{ci}\left(h_{ci}-h_{c12}\right)+\left(UD\right)_{c1}l_{c1}\left(\overline{T_{ca1}}-\overline{T_{cr1}}\right) \tag{6-20}$$

Species balance for refrigerant in control volume 2:

$$A_c l_{c1}\left(x_{c12}-x_{co}\right)\frac{d\overline{\rho_{c1}}}{dt}+A_c l_{c2}\left(\frac{d\overline{\rho_{c2}x_{c2}}}{dt}-x_{co}\frac{d\overline{\rho_{c2}}}{dt}\right)$$
$$+A_c\left\{\overline{\rho_{c1}}\left(x_{c12}-x_{co}\right)-\overline{\rho_{c2}x_{c2}}+\overline{\rho_{c2}}x_{co}\right\}\frac{dl_{c1}}{dt}=\dot{m}_{ci}\left(x_{c12}-x_{co}\right) \tag{6-21}$$

Energy balance for refrigerant in control volume 2:

$$A_c l_{c1}\left(h_{c12}-h_{co}\right)\frac{d\overline{\rho_{c1}}}{dt}+A_c l_{c2}\left(\frac{d\overline{\rho_{c2}h_{c2}}}{dt}-h_{co}\frac{d\overline{\rho_{c2}}}{dt}-\frac{dP_c}{dt}\right)$$
$$+A_c\left\{\overline{\rho_{c1}}\left(h_{c12}-h_{co}\right)-\overline{\rho_{c2}h_{c2}}+\overline{\rho_{c2}}h_{co}\right\}\frac{dl_{c1}}{dt}=\dot{m}_{ci}\left(h_{c12}-h_{co}\right)+\left(UD\right)_{c2}l_{c2}\left(\overline{T_{ca2}}-\overline{T_{cr2}}\right) \tag{6-22}$$

Because ammonia mass fraction is assumed uniform, overbars on x_{c1} and x_{c2} are not necessary and similarly for the bulk temperature of control volume 2, T_{cr2}. Now all three independent variables, pressure, temperature, and concentration are uniform, rather than spatially distributed, in control volume 2, so its density and enthalpy do not need overbars. For the coolant, it is assumed that both pressure and temperature are uniform also, although the only fundamental requirement is that it provides nearly constant heat flux.

At the interface between two-phase and sub-cooled liquid, the ammonia mass fraction is assumed to be the same as two-phase bulk ammonia mass fraction:

$$x_{c12} = x_{c1} \tag{6-23}$$

Then, Equation (6-19) is rewritten as

$$A_c l_{c1} \overline{\rho_{c1}} \frac{dx_{c1}}{dt} = \dot{m}_{ci} \left(x_{ci} - x_{c12} \right) \tag{6-24}$$

Mean values of density and enthalpy in the two-phase region are functions of pressure and ammonia mass fraction:

$$\frac{d\overline{\rho_{c1}}}{dt} = \frac{\partial \overline{\rho_{c1}}}{\partial P_c} \frac{dP_c}{dt} + \frac{\partial \overline{\rho_{c1}}}{\partial x_{c1}} \frac{dx_{c1}}{dt} \tag{6-25}$$

$$\frac{d\overline{\rho_{c1} h_{c1}}}{dt} = \frac{\partial \overline{\rho_{c1} h_{c1}}}{\partial P_c} \frac{dP_c}{dt} + \frac{\partial \overline{\rho_{c1} h_{c1}}}{\partial x_{c1}} \frac{dx_{c1}}{dt} \tag{6-26}$$

Sub-cooled liquid enthalpy and density are functions not only of pressure and ammonia concentration but also of temperature:

$$\frac{d\rho_{c2}}{dt} = \frac{\partial \rho_{c2}}{\partial P_c} \frac{dP_c}{dt} + \frac{\partial \rho_{c2}}{\partial T_{cr2}} \frac{dT_{cr2}}{dt} + \frac{\partial \rho_{c2}}{\partial x_{c2}} \frac{dx_{c2}}{dt} \tag{6-27}$$

$$\frac{d\rho_{c2} h_{c2}}{dt} = \frac{\partial \rho_{c2} h_{c2}}{\partial P_c} \frac{dP_c}{dt} + \frac{\partial \rho_{c2} h_{c2}}{\partial T_{cr2}} \frac{dT_{cr2}}{dt} + \frac{\partial \rho_{c2} h_{c2}}{\partial x_{c2}} \frac{dx_{c2}}{dt} \tag{6-28}$$

Then, the species equation for sub-cooled liquid is

$$
\begin{aligned}
&A_c\left\{\overline{\rho_{c1}}\left(x_{c12}-x_{co}\right)+\rho_{c2}\left(x_{co}-x_{c2}\right)\right\}\frac{dl_{c1}}{dt}+A_c l_{c1}\left(x_{c12}-x_{co}\right)\frac{\partial\overline{\rho_{c1}}}{\partial x_{c1}}\frac{dx_{c1}}{dt}\\
&+A_c l_{c2}\left\{\rho_{c2}+\left(x_{c2}-x_{co}\right)\frac{\partial\rho_{c2}}{\partial x_{c2}}\right\}\frac{dx_{c2}}{dt}+A_c l_{c2}\left(x_{c2}-x_{co}\right)\frac{\partial\rho_{c2}}{\partial T_{cr2}}\frac{dT_{cr2}}{dt}\\
&=\dot{m}_{ci}\left(x_{c12}-x_{co}\right)-\left\{A_c l_{c1}\left(x_{c12}-x_{co}\right)\frac{\partial\overline{\rho_{c1}}}{\partial P_c}+A_c l_{c2}\left(x_{c2}-x_{co}\right)\frac{\partial\rho_{c2}}{\partial P_c}\right\}\frac{dP_c}{dt}
\end{aligned}
\tag{6-29}
$$

Energy equations for both phases are rewritten after substitution of Equations (6-20) to (6-22).

$$
\begin{aligned}
&A_c\left(\overline{\rho_{c1}h_{c1}}-\overline{\rho_{c1}}h_{c12}\right)\frac{dl_{c1}}{dt}+A_c l_{c1}\left(\frac{\partial\overline{\rho_{c1}h_{c1}}}{\partial x_{c1}}-h_{c12}\frac{\partial\overline{\rho_{c1}}}{\partial x_{c1}}\right)\frac{dx_{c1}}{dt}\\
&=\dot{m}_{ci}\left(h_{ci}-h_{c12}\right)+\left(UD\right)_{c1}l_{c1}\left(T_{ca1}-\overline{T_{cr1}}\right)-A_c l_{c1}\left(\frac{\partial\overline{\rho_{c1}h_{c1}}}{\partial P_c}-h_{c12}\frac{\partial\overline{\rho_{c1}}}{\partial P_c}-1\right)\frac{dP_c}{dt}
\end{aligned}
\tag{6-30}
$$

$$
\begin{aligned}
&A_c\left\{\overline{\rho_{c1}}\left(h_{c12}-h_{co}\right)+\rho_{c2}\left(h_{co}-h_{c2}\right)\right\}\frac{dl_{c1}}{dt}+A_c l_{c1}\left(h_{c12}-h_{co}\right)\frac{\partial\overline{\rho_{c1}}}{\partial x_{c1}}\frac{dx_{c1}}{dt}\\
&+A_c l_{c2}\left\{\left(h_{c2}-h_{co}\right)\frac{\partial\rho_{c2}}{\partial x_{c2}}+\rho_{c2}\frac{\partial h_{c2}}{\partial x_{c2}}\right\}\frac{dx_{c2}}{dt}+A_c l_{c2}\left\{\left(h_{c2}-h_{co}\right)\frac{\partial\rho_{c2}}{\partial T_{cr2}}+\rho_{c2}\frac{\partial h_{c2}}{\partial T_{cr2}}\right\}\frac{dT_{cr2}}{dt}\\
&=\dot{m}_{ci}\left(h_{c12}-h_{co}\right)+\left(UD\right)_{c2}l_{c2}\left(T_{ca2}-T_{cr2}\right)\\
&-\left[A_c l_{c1}\left(h_{c12}-h_{co}\right)\frac{\partial\overline{\rho_{c1}}}{\partial P_c}+A_c l_{c2}\left\{\left(h_{c2}-h_{co}\right)\frac{\partial\rho_{c2}}{\partial P_c}+\rho_{c2}\frac{\partial h_{c2}}{\partial P_c}-1\right\}\right]\frac{dP_c}{dt}
\end{aligned}
\tag{6-31}
$$

Equations (6-17, 18, 24, 29, 30 and 31) can be expressed in matrix form:

$$
\mathbf{C}\dot{\mathbf{x}}_{\mathbf{c}}=\mathbf{F}_{\mathbf{c}}\left(\mathbf{x}_{\mathbf{c}},\mathbf{u}_{\mathbf{c}}\right)
\tag{6-32}
$$

where $\mathbf{C}=\begin{bmatrix} 0 & c_{12} & 0 & 0 & 0 & 0 \\ c_{21} & c_{22} & c_{23} & c_{24} & 0 & 0 \\ c_{31} & c_{32} & 0 & 0 & 0 & 0 \\ c_{41} & c_{42} & c_{43} & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{bmatrix}$, $\dot{\mathbf{x}}_\mathbf{c}=\begin{bmatrix} \dot{l}_{c1} \\ \dot{x}_{c1} \\ \dot{x}_{c2} \\ \dot{T}_{cr2} \\ \dot{T}_{ca1} \\ \dot{T}_{ca2} \end{bmatrix}$, $\mathbf{F}_\mathbf{c}=\begin{bmatrix} f_{c1} \\ f_{c2} \\ f_{c3} \\ f_{c4} \\ f_{c5} \\ f_{c6} \end{bmatrix}$, and $\mathbf{u}_\mathbf{c}=\begin{bmatrix} P_c \\ \dot{P}_c \\ x_{ci} \\ h_{ci} \\ \dot{m}_{ci} \\ T_{cai} \\ \dot{m}_{ca} \end{bmatrix}$

$\mathbf{x}_\mathbf{c}$ is the vector of state variables and $\mathbf{u}_\mathbf{c}$ is the vector of inputs for the condenser. The vector $\mathbf{F}_\mathbf{c}$ and the six by six matrix $\mathbf{C}$ are composed of coefficients shown in Appendix C. The derivative of the state variable vector, $\dot{\mathbf{x}}_\mathbf{c}$ is determined as

$$\dot{\mathbf{x}}_\mathbf{c}=\mathbf{C}^{-1}\mathbf{F}_\mathbf{c}\left(\mathbf{x}_\mathbf{c},\mathbf{u}_\mathbf{c}\right) \tag{6-33}$$

Outputs are $\mathbf{y}_\mathbf{c}=\begin{bmatrix} l_{c1} & x_{co} & h_{co} & \dot{m}_{co} & T_{cao} \end{bmatrix}^T$, consisting of the condenser mass flow rate, ammonia mass fraction and enthalpy at the exit of the refrigerant side, outlet temperature at coolant side, and the length of the two-phase region. Outlet mass flow rate is determined from the derivative of state variables and equations (6-11) and (6-14):

$$\begin{aligned} \dot{m}_{co}=\dot{m}_{ci}-A_c\left(\overline{\rho_{c1}}-\rho_{c2}\right)\frac{dl_{c1}}{dt}-A_c l_{c1}\left(\frac{\partial\overline{\rho_{c1}}}{\partial P_c}\frac{dP_c}{dt}+\frac{\partial\overline{\rho_{c1}}}{\partial x_{c1}}\frac{dx_{c1}}{dt}\right) \\ -A_c l_{c2}\left(\frac{\partial\rho_{c2}}{\partial P_c}\frac{dP_c}{dt}+\frac{\partial\rho_{c2}}{\partial x_{c2}}\frac{dx_{c2}}{dt}+\frac{\partial\rho_{c2}}{\partial T_{cr2}}\frac{dT_{cr2}}{dt}\right) \end{aligned} \tag{6-34}$$

For single-phase two-component fluid, exit ammonia mass fraction is same as bulk ammonia mass fraction of control volume 2. The bulk enthalpy is assumed to be the

average of the control volume inlet and outlet enthalpies. Therefore, outlet ammonia mass fraction and enthalpy are

$$x_{co} = x_{c2} \tag{6-35}$$

$$h_{co} = 2h_{c2} - h_{c12} \tag{6-36}$$

Because coolant in the condenser is assumed to have constant density, temperature differences are in proportion to enthalpy differences. So, the bulk temperature of the coolant is also assumed to be the average of the inlet and outlet temperatures.

$$T_{ca1} = \frac{1}{2}\left(T_{ca21} + T_{cao}\right) \tag{6-37}$$

$$T_{ca2} = \frac{1}{2}\left(T_{cai} + T_{ca21}\right) \tag{6-38}$$

$$T_{ca21} = 2T_{ca2} - T_{cai} \tag{6-39}$$

$$T_{cao} = 2T_{ca1} - T_{ca21} \tag{6-40}$$

Absorber

A conventional vapor compression refrigeration system uses a compressor to pressurize superheated refrigerant. Typical compressors need significant power to run the refrigerant system. However, a VARS uses an absorber and an energy efficient pump, which pressurizes liquid. Therefore, the absorber must produce a liquid absorbent-refrigerant fluid before the pump. Superheated or two-phase refrigerant is absorbed in a dilute solution in this device. For an ammonia-water VARS, almost pure ammonia vapor or two-phase refrigerant is absorbed in a weak solution. To maximize

the interface area of vapor and liquid, the weak liquid solution is sprayed into the vapor, or vapor is injected into the liquid solution.

In this research, however, the absorber is treated as an adiabatic mixing step, followed by a simple tube and shell heat exchanger. Two-phase nearly pure ammonia and a weak liquid solution mix before the absorber and their mixture, a strong solution, is cooled by the coolant. A schematic of an absorber is shown in Figure 6-7.

The absorber analysis is very similar to that of the condenser in this research. The only difference is that the inlet condition of the refrigerant side is not saturated vapor but two-phase. Therefore, the conservation equations for the absorber are also similar to those of the condenser (Equations (6-24), (6-29), (6-30), and (6-31)):

Species balance for refrigerant in control volume 1:

$$A_a l_{a1} \overline{\rho_{a1}} \frac{dx_{a1}}{dt} = \dot{m}_{ai} \left(x_{ai} - x_{a12} \right) \tag{6-41}$$

Energy balance for refrigerant in control volume 1:

$$\begin{aligned} & A_a \left(\overline{\rho_{a1} h_{a1}} - \overline{\rho_{a1}} h_{a12} \right) \frac{dl_{a1}}{dt} + A_a l_{a1} \left(\frac{\partial \overline{\rho_{a1} h_{a1}}}{\partial x_{a1}} - h_{a12} \frac{\partial \overline{\rho_{a1}}}{\partial x_{a1}} \right) \frac{dx_{a1}}{dt} \\ & = \dot{m}_{ai} \left(h_{ai} - h_{a12} \right) + \left(UD \right)_{a1} l_{a1} \left(T_{aa1} - \overline{T_{ar1}} \right) - A_a l_{a1} \left(\frac{\partial \overline{\rho_{a1} h_{a1}}}{\partial P_a} - h_{a12} \frac{\partial \overline{\rho_{a1}}}{\partial P_a} - 1 \right) \frac{dP_a}{dt} \end{aligned} \tag{6-42}$$

Species balance for refrigerant in control volume 2:

$$
\begin{aligned}
&A_a\left\{\overline{\rho_{a1}}\left(x_{a12}-x_{ao}\right)+\rho_{a2}\left(x_{ao}-x_{a2}\right)\right\}\frac{dl_{a1}}{dt}+A_a l_{a1}\left(x_{a12}-x_{ao}\right)\frac{\partial\overline{\rho_{a1}}}{\partial x_{a1}}\frac{dx_{a1}}{dt}\\
&+A_a l_{a2}\left\{\rho_{a2}+\left(x_{a2}-x_{ao}\right)\frac{\partial\rho_{a2}}{\partial x_{a2}}\right\}\frac{dx_{a2}}{dt}+A_a l_{a2}\left(x_{a2}-x_{ao}\right)\frac{\partial\rho_{a2}}{\partial T_{ar2}}\frac{dT_{ar2}}{dt}\\
&=\dot{m}_{ai}\left(x_{a12}-x_{ao}\right)-\left\{A_a l_{a1}\left(x_{a12}-x_{ao}\right)\frac{\partial\overline{\rho_{a1}}}{\partial P_a}+A_a l_{a2}\left(x_{a2}-x_{ao}\right)\frac{\partial\rho_{a2}}{\partial P_a}\right\}\frac{dP_a}{dt}
\end{aligned}
\tag{6-43}
$$

Energy balance for refrigerant in control volume 2:

$$
\begin{aligned}
&A_a\left\{\overline{\rho_{a1}}\left(h_{a12}-h_{ao}\right)+\rho_{a2}\left(h_{ao}-h_{a2}\right)\right\}\frac{dl_{a1}}{dt}+A_a l_{a1}\left(h_{a12}-h_{ao}\right)\frac{\partial\overline{\rho_{a1}}}{\partial x_{a1}}\frac{dx_{a1}}{dt}\\
&+A_a l_{a2}\left\{\left(h_{a2}-h_{ao}\right)\frac{\partial\rho_{a2}}{\partial x_{a2}}+\rho_{a2}\frac{\partial h_{a2}}{\partial x_{a2}}\right\}\frac{dx_{a2}}{dt}+A_a l_{a2}\left\{\left(h_{a2}-h_{ao}\right)\frac{\partial\rho_{a2}}{\partial T_{ar2}}+\rho_{a2}\frac{\partial h_{a2}}{\partial T_{ar2}}\right\}\frac{dT_{ar2}}{dt}\\
&=\dot{m}_{ai}\left(h_{a12}-h_{ao}\right)+\left(UD\right)_{a2}l_{a2}\left(T_{aa2}-T_{ar2}\right)\\
&-\left[A_a l_{a1}\left(h_{a12}-h_{ao}\right)\frac{\partial\overline{\rho_{a1}}}{\partial P_a}+A_a l_{a2}\left\{\left(h_{a2}-h_{ao}\right)\frac{\partial\rho_{a2}}{\partial P_a}+\rho_{a2}\frac{\partial h_{a2}}{\partial P_a}-1\right\}\right]\frac{dP_a}{dt}
\end{aligned}
\tag{6-44}
$$

To determine the mean values of the two-phase properties, a reference (imaginary) saturated liquid location needs to be found first. This imaginary moving boundary location is determined by inlet boundary conditions, pressure, and bulk ammonia mass fraction in the two-phase flow. If heat flux is uniform along the heat exchanger length, the specific enthalpy distribution is linear along the length. Based on pressure and bulk ammonia mass fraction, enthalpies for saturated vapor and liquid can be found. Then the specified inlet enthalpy determines the ratio of the dimensionless two-phase region length to that based on imaginary boundary.

The distributions of temperature, density and enthalpy are determined based on pressure and bulk ammonia mass fraction. Their mean values are calculated from properties between the inlet and the moving boundary inside the absorber for the

refrigerant side. For the coolant and sub-cooled refrigerant in the absorber, the analysis is similar to that of the condenser. So, refrigerant mass flow rate is determined similarly:

Energy balance for coolant in control volume 1:

$$\left(\rho AC_p\right)_{aa} l_{a1} \frac{dT_{aa1}}{dt} = \dot{m}_{aa} C_{paa} \left(T_{aa21} - T_{aao}\right) + \left(UD\right)_{a1} l_{a1} \left(\overline{T_{ar1}} - T_{aa1}\right) \tag{6-45}$$

Energy balance for coolant in control volume 2;

$$\left(\rho AC_p\right)_{aa} l_{a2} \frac{dT_{aa2}}{dt} = \dot{m}_{aa} C_{paa} \left(T_{aai} - T_{aa21}\right) + \left(UD\right)_{a2} l_{a2} \left(T_{ar2} - T_{aa2}\right) \tag{6-46}$$

The matrix form of the conservation equations is

$$\mathbf{A}\dot{\mathbf{x}}_\mathbf{a} = \mathbf{F}_\mathbf{a}\left(\mathbf{x}_\mathbf{a}, \mathbf{u}_\mathbf{a}\right) \tag{6-47}$$

where $\mathbf{A} = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix}$, $\dot{\mathbf{x}}_\mathbf{a} = \begin{bmatrix} \dot{l}_{a1} \\ \dot{x}_{a1} \\ \dot{x}_{a2} \\ \dot{T}_{ar2} \\ \dot{T}_{aa1} \\ \dot{T}_{aa2} \end{bmatrix}$, $\mathbf{F}_\mathbf{a} = \begin{bmatrix} f_{a1} \\ f_{a2} \\ f_{a3} \\ f_{a4} \\ f_{a5} \\ f_{a6} \end{bmatrix}$, and $\mathbf{u}_\mathbf{a} = \begin{bmatrix} P_a \\ \dot{P}_a \\ x_{ai} \\ h_{ai} \\ \dot{m}_{ai} \\ T_{aai} \\ \dot{m}_{aa} \end{bmatrix}$

$\mathbf{x}_\mathbf{a}$ is the vector of state variables and $\mathbf{u}_\mathbf{a}$ is the vector of inputs for the absorber. The vector $\mathbf{F}_\mathbf{a}$ and the six by six matrix $\mathbf{A}$ are composed of coefficients shown in Appendix C. The differentiation of state variables, $\dot{\mathbf{x}}_\alpha$ yields

$$\dot{\mathbf{x}}_\mathbf{a} = \mathbf{A}^{-1}\mathbf{F}_\mathbf{a}\left(\mathbf{x}_\mathbf{a}, \mathbf{u}_\mathbf{a}\right) \tag{6-48}$$

Outputs are $\mathbf{y_a} = \begin{bmatrix} l_{a1} & x_{ao} & h_{ao} & \dot{m}_{ao} & T_{aao} \end{bmatrix}^T$, considering of the absorber mass flow rate, ammonia mass fraction and enthalpy at the exit of refrigerant side, outlet temperature at coolant side, and the length of two-phase region. The outlet mass flow rate is again determined by derivatives of state variables and mass balances:

$$\dot{m}_{ao} = \dot{m}_{ai} - A_a\left(\overline{\rho_{a1}} - \rho_{a2}\right)\frac{dl_{a1}}{dt} - A_a l_{a1}\left(\frac{\partial\overline{\rho_{a1}}}{\partial P_a}\frac{dP_a}{dt} + \frac{\partial\overline{\rho_{a1}}}{\partial x_{a1}}\frac{dx_{a1}}{dt}\right) - A_a l_{a2}\left(\frac{\partial\rho_{a2}}{\partial P_a}\frac{dP_a}{dt} + \frac{\partial\rho_{a2}}{\partial x_{a2}}\frac{dx_{a2}}{dt} + \frac{\partial\rho_{a2}}{\partial T_{ar2}}\frac{dT_{ar2}}{dt}\right) \tag{6-49}$$

For single-phase two-component fluid, exit ammonia mass fraction is the same as the bulk ammonia mass fraction of control volume 2. The bulk enthalpy is assumed to be the average of the control volume inlet and outlet enthalpies. Therefore, outlet ammonia mass fraction and enthalpy are

$$x_{ao} = x_{a2} \tag{6-50}$$

$$h_{ao} = 2h_{a2} - h_{a12} \tag{6-51}$$

Because the coolant in the absorber is assumed to have constant density and heat capacity, temperature differences are in proportion to enthalpy differences. So, the bulk temperature of coolant is the average of the inlet and outlet temperatures.

$$T_{aa21} = 2T_{aa2} - T_{aai} \tag{6-52}$$

$$T_{aao} = 2T_{aa1} - T_{aa21} \tag{6-53}$$

Generator

A VARS may use various heat sources as an energy input, such as power cycle waste heat as in this study. This heat generates refrigerant vapor from a two-component liquid in a (so called) generator to provide the high-quality energy input needed for a VARS. Unlike the first two heat exchangers, sub-cooled liquid enters and two-phase fluid exits; in the actual system, the phases would be subsequently separated. Figure 6-8 shows a schematic of a generator heat exchange process.

The equations for the generator are also very similar to those of the first two heat exchangers. For the generator, mean values are used for control volume 2 instead of control volume 1. The resulting equations are:

Species balance for refrigerant in control volume 1:

$$\begin{aligned} & A_g \rho_{g1} \left(x_{g1} - x_{g12} \right) \frac{dl_{g1}}{dt} + A_g l_{g1} \left(x_{g1} - x_{g12} \right) \frac{\partial \rho_{g1}}{\partial T_{gr1}} \frac{dT_{gr1}}{dt} \\ & + A_g l_{g1} \left\{ \rho_{g1} + \left(x_{g1} - x_{g12} \right) \frac{\partial \rho_{g1}}{\partial x_{g1}} \right\} \frac{dx_{g1}}{dt} \\ & = \dot{m}_{gi} \left(x_{gi} - x_{g12} \right) - A_g l_{g1} \left(x_{g1} - x_{g12} \right) \frac{\partial \rho_{g1}}{\partial P_g} \frac{dP_g}{dt} \end{aligned} \tag{6-54}$$

Species balance for refrigerant in control volume 2:

$$\begin{aligned} & A_g \rho_{g1} \left(x_{g12} - x_{go} \right) \frac{dl_{g1}}{dt} + A_g l_{g1} \left(x_{g12} - x_{go} \right) \frac{\partial \rho_{g1}}{\partial T_{gr1}} \frac{dT_{gr1}}{dt} \\ & + A_g l_{g1} \left(x_{g12} - x_{g\ o} \right) \frac{\partial \rho_{g1}}{\partial x_{g1}} \frac{dx_{g1}}{dt} + A_g l_{g2} \overline{\rho}_{g2} \frac{dx_{g2}}{dt} \\ & = \dot{m}_{gi} \left(x_{g12} - x_{go} \right) - A_g l_{g1} \left(x_{g12} - x_{go} \right) \frac{\partial \rho_{g1}}{\partial P_a} \frac{dP_a}{dt} \end{aligned} \tag{6-55}$$

Energy balance for refrigerant in control volume 1:

$$\begin{aligned} & A_g \rho_{g1} \left(h_{g1} - h_{g12}\right) \frac{dl_{g1}}{dt} + A_g l_{g1} \left\{ \left(h_{g1} - h_{g12}\right) \frac{\partial \rho_{g1}}{\partial T_{gr1}} + \rho_{g1} \frac{\partial h_{g1}}{\partial T_{gr1}} \right\} \frac{dT_{gr1}}{dt} \\ & + A_g l_{g1} \left\{ \left(h_{g1} - h_{g12}\right) \frac{\partial \rho_{g1}}{\partial x_{g1}} + \rho_{g1} \frac{\partial h_{g1}}{\partial x_{g1}} \right\} \frac{dx_{g1}}{dt} \\ = \dot{m}_{gi} \left(h_{gi} - h_{g12}\right) + \left(UD\right)_{g1} l_{g1} \left(T_{ga1} - T_{gr1}\right) - A_g l_{g1} & \left\{ \left(h_{g1} - h_{g12}\right) \frac{\partial \rho_{g1}}{\partial P_g} + \rho_{g1} \frac{\partial h_{g1}}{\partial P_g} - 1 \right\} \frac{dP_g}{dt} \end{aligned} \tag{6-56}$$

Energy balance for refrigerant in control volume 2:

$$\begin{aligned} & A_g \left\{ \rho_{g1} \left(h_{g12} - h_{go}\right) + \left(\overline{\rho_{g2} h_{g2}} - \overline{\rho_{g2}} h_{go}\right) \right\} \frac{dl_{g1}}{dt} + A_g l_{g1} \left(h_{g12} - h_{go}\right) \frac{\partial \rho_{g1}}{\partial T_{gr1}} \frac{dT_{gr1}}{dt} \\ & + A_g l_{g1} \left(h_{g12} - h_{go}\right) \frac{\partial \rho_{g1}}{\partial x_{g1}} \frac{dx_{g1}}{dt} + A_g l_{g2} \left(\frac{\partial \overline{\rho_{g2} h_{g2}}}{\partial x_{g2}} - h_{go} \frac{\partial \overline{\rho_{g2}}}{\partial x_{g2}} \right) \frac{dx_{g2}}{dt} \\ & = \dot{m}_{gi} \left(h_{g12} - h_{go}\right) + \left(UD\right)_{g2} l_{g2} \left(T_{ga2} - \overline{T_{gr2}}\right) \\ & - \left\{ A_g l_{g1} \left(h_{g12} - h_g\right) \frac{\partial \rho_{g1}}{\partial P_g} - A_g l_{g2} \left(\frac{\partial \overline{\rho_{g2} h_{g2}}}{\partial P_g} - h_{go} \frac{\partial \overline{\rho_{g2}}}{\partial P_g} - 1 \right) \right\} \frac{dP_g}{dt} \end{aligned} \tag{6-57}$$

Energy balance for coolant control volume 1:

$$\left(\rho A C_p\right)_{ga} l_{g1} \frac{dT_{ga1}}{dt} = \dot{m}_{ga} C_{pga} \left(T_{ga21} - T_{gao}\right) + \left(UD\right)_{g1} l_{g1} \left(T_{gr1} - T_{ga1}\right) \tag{6-58}$$

Energy balance for coolant control volume 2:

$$\left(\rho A C_p\right)_{ga} l_{g2} \frac{dT_{ga2}}{dt} = \dot{m}_{ga} C_{pga} \left(T_{gai} - T_{ga21}\right) + \left(UD\right)_{g2} l_{g2} \left(\overline{T_{gr2}} - T_{ga2}\right) \tag{6-59}$$

The matrix form of the conservation equations is

$$\mathbf{G}\dot{\mathbf{x}}_\mathbf{g} = \mathbf{F}_\mathbf{g}\left(\mathbf{x}_\mathbf{g}, \mathbf{u}_\mathbf{g}\right) \tag{6-60}$$

where $\mathbf{G} = \begin{bmatrix} 0 & 0 & g_{13} & 0 & 0 & 0 \\ g_{21} & g_{22} & g_{23} & g_{24} & 0 & 0 \\ g_{31} & g_{32} & g_{33} & 0 & 0 & 0 \\ g_{41} & g_{42} & g_{43} & g_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & g_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & g_{66} \end{bmatrix}$, $\dot{\mathbf{x}}_\mathbf{g} = \begin{bmatrix} \dot{l}_{g1} \\ \dot{T}_{gr1} \\ \dot{x}_{g1} \\ \dot{x}_{g2} \\ \dot{T}_{ga1} \\ \dot{T}_{ga2} \end{bmatrix}$, $\mathbf{F}_\mathbf{g} = \begin{bmatrix} f_{g1} \\ f_{g2} \\ f_{g3} \\ f_{g4} \\ f_{g5} \\ f_{g6} \end{bmatrix}$, and $\mathbf{u}_\mathbf{g} = \begin{bmatrix} P_g \\ \dot{P}_g \\ x_{gi} \\ h_{gi} \\ \dot{m}_{gi} \\ T_{gai} \\ \dot{m}_{ga} \end{bmatrix}$

$\mathbf{x}_\mathbf{g}$ is the vector of state variables and $\mathbf{u}_\mathbf{g}$ is the vector of inputs for the generator. The vector $\mathbf{F}_\mathbf{g}$ and the 6 by 6 matrix $\mathbf{G}$ are composed of coefficients shown at Appendix C. The differentiation of state variables, $\dot{\mathbf{x}}_\mathbf{g}$ yields

$$\dot{\mathbf{x}}_\mathbf{g} = \mathbf{G}^{-1}\mathbf{F}_\mathbf{g}\left(\mathbf{x}_\mathbf{g}, \mathbf{u}_\mathbf{g}\right) \tag{6-61}$$

Outputs are $\mathbf{y}_\mathbf{g} = \begin{bmatrix} l_{g1} & x_{gol} & x_{gog} & h_{gol} & h_{gog} & \dot{m}_{gol} & \dot{m}_{gog} & T_{aao} \end{bmatrix}^T$, consisting of the generator mass flow rates, ammonia mass fractions and enthalpies in both saturated vapor and liquid at the exit of refrigerant side, outlet temperature at coolant side, and the length of single-phase region. To find the refrigerant outputs, the state of the refrigerant at the exit must be known. The saturated vapor position is after the outlet, as shown in Figure 6-10. There are two different ways to determine the exit. One is to find the refrigerant mean temperature, and the other is to use the imaginary moving boundary position directly. For a generator, the first method is used, whereas the second one is used for an evaporator.

Finding the refrigerant mean temperature is similar to the method used in absorber. For a given instant in time, the two-phase mean temperature is calculated from heat source thermodynamic properties and the universal heat transfer coefficient:

$$\overline{T_{gr2}} = T_{ga2} + \left(T_{ga2} - T_{gai}\right)\frac{2\dot{m}_{ga}C_{pga}}{\left(UD\right)_{g2}\, l_{g2}} \tag{6-62}$$

Based on the average refrigerant temperature in control volume 2, the imaginary saturated vapor location, L_g^*, can be found. Once L_g^* is calculated, the profiles of enthalpies and ammonia mass fractions in both saturated vapor and liquid, as well as vapor quality (or dryness), $d_g(z)$, in two-phase region are determined, with the assumptions that heat flux along the generator is constant, and thermodynamic equilibrium exists within each infinitesimal control volume.

Outlet total mass flow rate is determined from derivatives of state variables and mass balances:

$$\begin{aligned}\dot{m}_{go} = \dot{m}_{gi} - A_g\left(\rho_{g1} - \overline{\rho_{g2}}\right)\frac{dl_{g1}}{dt} - A_g l_{g2}\left(\frac{\partial\overline{\rho_{g2}}}{\partial P_g}\frac{dP_g}{dt} + \frac{\partial\overline{\rho_{g2}}}{\partial x_{g2}}\frac{dx_{g2}}{dt}\right) \\ - A_g l_{g1}\left(\frac{\partial\rho_{g1}}{\partial P_g}\frac{dP_g}{dt} + \frac{\partial\rho_{g1}}{\partial x_{g1}}\frac{dx_{g1}}{dt} + \frac{\partial\rho_{g1}}{\partial T_{gr1}}\frac{dT_{gr1}}{dt}\right)\end{aligned} \tag{6-63}$$

From the total mass flow rate and quality distribution, vapor and liquid mass flow rates are determined at the exit:

$$\dot{m}_{gog} = d_g\dot{m}_{go} \tag{6-64}$$

$$\dot{m}_{gol} = \left(1 - d_g\right)\dot{m}_{go} \tag{6-65}$$

Because the fluid heat source in the generator is assumed to have constant density and heat capacity, temperature differences are in proportion to enthalpy differences. So, the bulk temperature of the coolant is the average of the inlet and outlet temperatures.

$$T_{ga21} = 2T_{ga2} - T_{gai} \tag{6-66}$$

$$T_{gao} = 2T_{ga1} - T_{ga21} \tag{6-67}$$

Evaporator

For most refrigeration systems using single component refrigerant, two-phase, but almost saturated liquid or liquid is fully vaporized in evaporator to maximize cooling effect and to protect compressor from liquid droplets after evaporator. For an ammonia-water VARS, however, almost pure ammonia refrigerant always contains small amount of water because rectifier cannot make pure ammonia. Saturation temperature is not constant any more even with very small amount of water as shown in Figure 6-2.

At very high ammonia mass fraction, 0.998, saturation temperature can be treated as a constant for some region, but temperature rises up dramatically near saturated vapor area. To use lowest possible evaporator temperature, refrigerant is not fully vaporized intentionally in an evaporator for a VARS. Therefore, fluid conditions of inlet and outlet of evaporator are two-phase during normal evaporator operation. A schematic of an evaporator is shown in Figure 6-9.

For the evaporator, imaginary moving boundary position is directly used to get inlet and outlet conditions of the evaporator. Consider two control volumes, one is from z=0 to L_e^* and the other is from z=L_e to L_e^*. L_e^* is the location of imaginary moving boundary.

Note that control volume 1 includes control volume 2. Conservations equations for the evaporator are determined as follows (see Appendix C for detailed derivations).

Mass balance for refrigerant:

$$A_e\left(\overline{\rho_{e1}}-\overline{\rho_{e2}}\right)\frac{dL_e^*}{dt}+A_eL_e^*\frac{d\overline{\rho_{e1}}}{dt}+A_e\left(L_e-L_e^*\right)\frac{d\overline{\rho_{e2}}}{dt}=\dot{m}_{ei}-\dot{m}_{eo} \tag{6-68}$$

Species balance for refrigerant:

$$A_e\left\{L_e^*\overline{\rho_{e1}}+\left(L_e-L_e^*\right)\overline{\rho_{e2}}\right\}\frac{dx_e}{dt}=\dot{m}_{ei}\left(x_{ei}-x_{eo}\right) \tag{6-69}$$

Energy balance for refrigerant:

$$\begin{aligned}&A_e\left(\overline{\rho_{e1}h_{e1}}-\overline{\rho_{e1}}h_{eo}+\overline{\rho_{e2}}h_{eo}-\overline{\rho_{e2}h_{e2}}\right)\frac{dL_e^*}{dt}\\&+\left\{A_eL_e^*\left(\frac{\partial\overline{\rho_{e1}h_{e1}}}{\partial x_e}-h_{eo}\frac{\partial\overline{\rho_{e1}}}{\partial x_e}\right)+A_e\left(L_e-L_e^*\right)\left(\frac{\partial\overline{\rho_{e2}h_{e2}}}{\partial x_e}-h_{eo}\frac{\partial\overline{\rho_{e2}}}{\partial x_e}\right)\right\}\frac{dx_e}{dt}\\&=\dot{m}_{ei}\left(h_{ei}-h_{eo}\right)+\left(UA\right)_e\left(T_{ea}-\overline{T_{er}}\right)\\&-\left\{A_eL_e^*\left(\frac{\partial\overline{\rho_{e1}h_{e1}}}{\partial P_e}-h_{eo}\frac{\partial\overline{\rho_{e1}}}{\partial P_e}-1\right)+A_e\left(L_e-L_e^*\right)\left(\frac{\partial\overline{\rho_{e2}h_{e2}}}{\partial P_e}-h_{eo}\frac{\partial\overline{\rho_{e2}}}{\partial P_e}-1\right)\right\}\frac{dP_e}{dt}\end{aligned} \tag{6-70}$$

Energy balance for the secondary fluid:

$$\left(\rho AC_p\right)_{ea}L_e\frac{dT_{ea}}{dt}=\dot{m}_{ea}C_{pea}\left(T_{eai}-T_{eao}\right)+\left(UA\right)_e\left(\overline{T_{er}}-T_{ea}\right) \tag{6-71}$$

The matrix form of the conservation equations for the evaporator is

$$\mathbf{E}\dot{\mathbf{x}}_\mathbf{e}=\mathbf{F}_\mathbf{e}\left(\mathbf{x}_\mathbf{e},\mathbf{u}_\mathbf{e}\right) \tag{6-72}$$

where $\mathbf{E}=\begin{bmatrix} 0 & e_{12} & 0 \\ e_{21} & e_{22} & 0 \\ 0 & 0 & e_{33} \end{bmatrix}$, $\dot{\mathbf{x}}_\mathbf{e}=\begin{bmatrix} \dot{L}_e^* \\ \dot{x}_e \\ \dot{T}_{ea} \end{bmatrix}$, $\mathbf{F}_\mathbf{e}=\begin{bmatrix} f_{e1} \\ f_{e2} \\ f_{e3} \end{bmatrix}$, and $\mathbf{u}_\mathbf{e}=\begin{bmatrix} P_e \\ \dot{P}_e \\ x_{ei} \\ h_{ei} \\ \dot{m}_{ei} \\ T_{eai} \\ \dot{m}_{ea} \end{bmatrix}$

$\mathbf{x}_\mathbf{e}$ is the vector of state variables and $\mathbf{u}_\mathbf{e}$ is the vector of inputs for the evaporator. The vector $\mathbf{F}_\mathbf{e}$ and the 3 by 3 matrix $\mathbf{E}$ are composed of coefficients shown at Appendix C. The differentiation of state variables, $\dot{\mathbf{x}}_\mathbf{e}$ yields

$$\dot{\mathbf{x}}_\mathbf{e}=\mathbf{E}^{-1}\mathbf{F}_\mathbf{e}\left(\mathbf{x}_\mathbf{e},\mathbf{u}_\mathbf{e}\right) \tag{6-73}$$

Outputs are $\mathbf{y}_\mathbf{e}=\begin{bmatrix} L_e^* & x_{eo} & h_{eo} & \dot{m}_{eo} & T_{eao} \end{bmatrix}^T$, consisting of the evaporator mass flow rate, ammonia mass fraction, and enthalpy at the exit of refrigerant side, outlet temperature at coolant side and the imaginary boundary location. Outlet mass flow rate is determined by derivatives of state variables and equations (6-68):

$$\begin{aligned} \dot{m}_{eo}=\dot{m}_{ei}-A_e\left(\overline{\rho_{e1}}-\overline{\rho_{e2}}\right)\frac{dL_e^*}{dt}-\left\{A_eL_e^*\frac{\partial\overline{\rho_{e1}}}{\partial x_e}+A_e\left(L_e-L_e^*\right)\frac{\partial\overline{\rho_{e2}}}{\partial x_e}\right\}\frac{dx_e}{dt} \\ -\left\{A_eL_e^*\frac{\partial\overline{\rho_{e1}}}{\partial P_e}+A_e\left(L_e-L_e^*\right)\frac{\partial\overline{\rho_{e2}}}{\partial P_e}\right\}\frac{dP_e}{dt} \end{aligned} \tag{6-74}$$

Because the secondary fluid in evaporator is assumed to have constant density, temperature differences are in proportion to enthalpy differences. So, the bulk temperature of the secondary fluid is the average of the inlet and outlet temperatures.

$$T_{eao} = 2T_{ea} - T_{eai} \tag{6-75}$$

Solution Heat Exchanger

The fluid from the bottom of rectifier is hot enough to warm up the fluid from pump to generator in SHX. Therefore, heat energy required to run a VARS from generator can be reduced. The fluids through SHX are always sub-cooled liquid without phase change in the heat exchanger. Because there is no moving interface in SHX, only two control volumes are remained to be modeled. A schematic of SHX is shown Figure 6-10. Even though the fluid is sub-cooled liquid, species balance is necessary because the variance of ammonia mass fraction cannot be ignored.

Mass balance for colder fluid:

$$A_{sc}L_s\frac{d\rho_{sc}}{dt} = \dot{m}_{sci} - \dot{m}_{sco} \tag{6-76}$$

Species balance for colder fluid:

$$\begin{aligned} &A_{sc}L_s\left(x_{sc}-x_{sco}\right)\frac{\partial\rho_{sc}}{\partial T_{sc}}\frac{dT_{sc}}{dt} + A_{sc}L_s\left\{\left(x_{sc}-x_{sco}\right)\frac{\partial\rho_{sc}}{\partial x_{sc}} + \rho_{sc}\right\}\frac{dx_{sc}}{dt} \\ &= \dot{m}_{sci}\left(x_{sci}-x_{sco}\right) - A_{sc}L_s\left(x_{sc}-x_{sco}\right)\frac{\partial\rho_{sc}}{\partial P_s}\frac{dP_s}{dt} \end{aligned} \tag{6-77}$$

Energy balance for colder fluid:

$$\begin{aligned} &A_{sc}L_s\left\{\left(h_{sc}-h_{sco}\right)\frac{\partial\rho_{sc}}{\partial T_{sc}} + \rho_{sc}\frac{\partial h_{sc}}{\partial T_{sc}}\right\}\frac{dT_{sc}}{dt} + A_{sc}L_s\left\{\left(h_{sc}-h_{sco}\right)\frac{\partial\rho_{sc}}{\partial x_{sc}} + \rho_{sc}\frac{\partial h_{sc}}{\partial x_{sc}}\right\}\frac{dx_{sc}}{dt} \\ &= \dot{m}_{sci}\left(h_{sci}-h_{sco}\right) + \left(UA\right)_s\left(T_{sh}-T_{sc}\right) - A_{sc}L_s\left\{\left(h_{sc}-h_{sco}\right)\frac{\partial\rho_{sc}}{\partial P_s} + \rho_{sc}\frac{\partial h_{sc}}{\partial P_s} - 1\right\}\frac{dP_s}{dt} \end{aligned} \tag{6-78}$$

Mass balance for hotter fluid:

$$A_{sh}L_s\frac{d\rho_{sh}}{dt}=\dot{m}_{shi}-\dot{m}_{sho} \tag{6-79}$$

Species balance for hotter fluid:

$$\begin{aligned}&A_{sh}L_s\left(x_{sh}-x_{shi}\right)\frac{\partial\rho_{sh}}{\partial T_{sh}}\frac{dT_{sh}}{dt}+A_{sh}L_s\left\{\left(x_{sh}-x_{shi}\right)\frac{\partial\rho_{sh}}{\partial x_{sh}}+\rho_{sh}\right\}\frac{dx_{sh}}{dt}\\&=\dot{m}_{sho}\left(x_{shi}-x_{sho}\right)-A_{sh}L_s\left(x_{sh}-x_{shi}\right)\frac{\partial\rho_{sh}}{\partial P_s}\frac{dP_s}{dt}\end{aligned} \tag{6-80}$$

Energy balance for hotter fluid:

$$\begin{aligned}&A_{sh}L_s\left\{\left(h_{sh}-h_{schi}\right)\frac{\partial\rho_{sh}}{\partial T_{sh}}+\rho_{sh}\frac{\partial h_{sh}}{\partial T_{sh}}\right\}\frac{dT_{sh}}{dt}+A_{sh}L_s\left\{\left(h_{sh}-h_{shi}\right)\frac{\partial\rho_{sh}}{\partial x_{sh}}+\rho_{sh}\frac{\partial h_{sh}}{\partial x_{sh}}\right\}\frac{dx_{sh}}{dt}\\&=\dot{m}_{sho}\left(h_{shi}-h_{sho}\right)+\left(UA\right)_s\left(T_{sc}-T_{sh}\right)-A_{sh}L_s\left\{\left(h_{sh}-h_{shi}\right)\frac{\partial\rho_{sh}}{\partial P_s}+\rho_{sh}\frac{\partial h_{sh}}{\partial P_s}-1\right\}\frac{dP_s}{dt}\end{aligned} \tag{6-81}$$

The matrix form of the conservation equations is

$$\mathbf{S}\dot{\mathbf{x}}_\mathbf{s}=\mathbf{F}_\mathbf{s} \tag{6-82}$$

where $\mathbf{S}=\begin{bmatrix}s_{11}&s_{12}&0&0\\s_{21}&s_{22}&0&0\\0&0&s_{33}&s_{34}\\0&0&s_{43}&s_{44}\end{bmatrix}$, $\dot{\mathbf{x}}_\mathbf{s}=\begin{bmatrix}\dot{T}_{sc}\\\dot{x}_{sc}\\\dot{T}_{sh}\\\dot{x}_{sh}\end{bmatrix}$, $\mathbf{F}_\mathbf{s}=\begin{bmatrix}f_{s1}\\f_{s2}\\f_{s3}\\f_{s4}\end{bmatrix}$

The differentiation of state variables, $\dot{\mathbf{x}}_\mathbf{s}$ are determined as:

$$\dot{\mathbf{x}}_\mathbf{s}=\mathbf{S}^{-1}\mathbf{F}_\mathbf{s} \tag{6-83}$$

Outputs of SHX are mass flow rate, ammonia mass fraction and enthalpy at the exit of colder refrigerant side and inlet mass flow rate and outlet ammonia mass fraction and enthalpy at hotter side. Outlet mass flow rate of colder refrigerant is determined by derivatives of state variables and mass balance:

$$\dot{m}_{sco} = \dot{m}_{sci} - A_{sc} L_s \left(\frac{\partial \rho_{sc}}{\partial P_s} \frac{dP_s}{dt} + \frac{\partial \rho_{sc}}{\partial T_{sc}} \frac{dT_{sc}}{dt} + \frac{\partial \rho_{sc}}{\partial x_{sc}} \frac{dx_{sc}}{dt} \right) \tag{6-84}$$

$$x_{sco} = x_{sc} \tag{6-85}$$

$$h_{sco} = 2h_{sc} - h_{sci} \tag{6-86}$$

Inlet mass flow rate of hotter refrigerant is determined by derivatives of state variables and mass balance:

$$\dot{m}_{xhi} = \dot{m}_{so} + A_{sh} L_s \left(\frac{\partial \rho_{sh}}{\partial P_s} \frac{dP_s}{dt} + \frac{\partial \rho_{sh}}{\partial T_{sh}} \frac{dT_{sh}}{dt} + \frac{\partial \rho_{sh}}{\partial x_{sh}} \frac{dx_{sh}}{dt} \right) \tag{6-87}$$

$$x_{sho} = x_{sh} \tag{6-88}$$

$$h_{sho} = 2h_{sh} - h_{shi} \tag{6-89}$$

Results and Discussion

Computer codes implementing the model for the heat exchangers as described above are now used to calculate exit conditions of those components, based on step inputs. Ordinary differential equations for each component constitute the governing equations to be solved simultaneously with constitutive relations. They were solved numerically by Gear's method, which uses backward differentiation and is used when a problem is stiff. The values of temperature (or enthalpy), mass flow rate, and ammonia mass fraction of each component were input as a function of time. The time increment

in numerical integration was automatically adjusted to keep the largest acceptable solver error to be 1×10^{-5} during each time step.

For general expressions of outputs, non-dimensional variables were used for time and output variables. Non-dimensional time and variables are expressed as

$$t_j^* = \frac{t}{\tau_j} \tag{6-90}$$

$$x^* = \frac{x - x_{final}}{x_{initial} - x_{final}} \tag{6-91}$$

where characteristic time, $\tau_j = \frac{m_j}{\dot{m}_{j,i}}$, m_j and $\dot{m}_{j,i}$ are total mass in the control volume and inlet mass flow rate of the refrigerant side in the heat exchanger, j. x is a output variable, $x_{initial}$ and x_{final} are initial and final values of x.

Condenser

Figures 6-11 to 6-13 show non-dimensional response to three step inputs. Inlet ammonia mass fraction, inlet mass flow rate, and secondary fluid inlet mass flow rate are input perturbations; ammonia mass fraction, enthalpy, and secoundary fluid temperature at the exit and the interface position are shown as outputs. Inputs, initial conditions, and geometric parameters of the condenser are given in Table 6-1.

When inlet ammonia mass fraction is suddenly increased, saturated vapor enthalpy is decreased. As a result, the two-phase region is shorter, and the sub-cooled liquid is further cooled so that the exit enthalpy is decreased. While outputs of ammonia mass fraction and secondary fluid temperature at the exit transition to the final values monotonically, the interface position and enthalpy at the exit show undershoots (Figure

6-11). However, if a higher flow rate of saturated vapor is coupled with constant coolant flow (Figure 6-12), the two-phase/single-phase interface position is close to the exit, therefore, exit enthalpy is increased. On the other hand, higher flow rate of coolant (higher heat flux) reduces the two-phase region (Figure 6-13). When mass flow rates are perturbed, the time constants are shorter than for the case when inlet ammonia mass fraction is changed.

Absorber

Figures 6-14 to 6-17 show non-dimensional response to four step inputs for the absorber. Inlet ammonia mass fraction, enthalpy and mass flow rate and secondary fluid mass flow rate are input perturbations; ammonia mass fraction and enthalpy at the exit and the interface position are shown as outputs. Inputs, initial conditions, and geometric parameters of the absorber are given in Table 6-2.

Unlike the condenser, absorber inlet enthalpy is an independent variable. Therefore, inlet ammonia mass fraction and pressure change do not affect inlet enthalpy directly because the absorber inlet condition is two-phase as shown in Figure 6-7. The effect of an increase of inlet ammonia mass fraction and secondary fluid mass flow rate is an increased cooling effect to the ammonia-water mixture. Therefore, the length of the two-phase region is shorter as exit enthalpy is decreased. We can see the opposite behavior when inlet enthalpy and mass flow rate are increased.

There is unexpected inaccuracy in Figure 6-14. Exit temperature of secondary flow is expected to decrease, but the temperature increases in early stages. This artifact is because of the assumption that the control volume temperature is the average of inlet and outlet temperature for the secondary fluid. Single phase fluids in moving boundary models traditionally have the potential for this non-physical effect [22]. As shown at

condenser, the response time is longer for the perturbation of inlet ammonia mass fraction for absorber.

Generator

Figures 6-18 and 6-19 show non-dimensional response to two step inputs for the generator. Inlet ammonia mass fraction and inlet mass flow rate are the input perturbations, while exit enthalpies and ammonia mass fractions in both vapor and liquid are the outputs shown, which are important input variables for the rectifier. Inputs, initial conditions, and geometric parameters of the generator are given in Table 6-3.

For the generator, ammonia-water mixtures are heated rather than cooled as in the condenser and absorber. Therefore, non-dimensional enthalpies for both vapor and liquid tend to increase. If inlet ammonia mass fraction and mass flow rate are increased, vapor and liquid enthalpies will be decreased while the ammonia mass fraction at the exit will be increased. Although the final results are as expected, there is an unexpected jump in the opposite direction when ammonia mass fraction is suddenly increased, as in Figure 6-18. The reason is the same as for the absorber. Because the enthalpy of single-phase flow is assumed to be the average of inlet and outlet enthalpies, if initial enthalpy is suddenly changed, the outlet value is immediately affected. This means that the model is inaccurate early in step changer transients, as He et al. [21] mentioned in his research.

Evaporator

Figures 6-20 to 6-22 show non-dimensional response to three step inputs for the evaporator. Inlet ammonia mass fraction, enthalpy, and mass flow rate are input perturbations; exit ammonia mass fraction, enthalpy, imaginary boundary position, and

secondary fluid temperature are shown as outputs. Inputs, initial conditions, and geometric parameters of the evaporator are given in Table 6-4.

The refrigerant in the evaporator is heated by the hot secondary fluid. Therefore, lower inlet enthalpy increases the cooling effect. For almost saturated vapor, when ammonia mass fraction is decreased, the enthalpy is also decreased, unlike the case for nearly saturated ammonia liquid. Both cases increase the cooling effect of the evaporator and the results decreased length of the imaginary moving boundary and lower exit enthalpies.

The cases for perturbed enthalpy and mass flow rate show changed show relatively shorter response time than that of ammonia mass flow change, just like the above-mentioned two heat exchangers.

Solution Heat Exchanger

Figures 6-23 and 6-24 show non-dimensional response to two step inputs for the solution heat exchanger. Because both fluids are ammonia-water mixtures and are liquid phase, only the colder fluid results are considered here. Inlet ammonia mass fraction and mass flow rate are input perturbations; exit ammonia mass fractions and enthalpies for both fluids are shown as outputs. Inputs, initial conditions, and geometric parameters of the SHX are given in Table 6-5.

For sub-cooled liquid of ammonia-water mixtures, the mixture enthalpy is a function of not only pressure and ammonia mass fraction but also temperature. When only inlet ammonia mass fraction is increased at the specified fixed pressure and inlet enthalpy, inlet temperature can be increased or decreased because mixture temperature is not monotonic on an enthalpy-composition plane. For the case represented in Table 6-5, colder flow temperature is increased. When mass flow rate of

the colder flow is increased, the overall SHX is cooled more, resulting in lower enthalpies for both fluids.

Summary

A dynamic modeling method appropriate to heat exchangers for control applications has been developed in this chapter. To satisfy both two-phase flow physics and the control design requirements, the conventional moving boundary model has been adapted. This model has been used in the literature to model single-component two-phase fluid heat exchangers for control applications. The conventional moving boundary model has been extended in this work to include species balance for two-component two-phase fluids. Among three modeling options considered, the spatial mean value model was employed, along with tabulated steady-state property data for the two-phase region.

Use of tabulated data significantly reduces the number of equations and numerical complexity of determining equilibrium state. Also, the data can be easily applied to imaginary moving boundaries, which are reference states outside of the physical heat exchangers. Because thermodynamic properties at the boundaries are known, two-phase inlet and outlet conditions are easily obtained.

However, this newly-developed moving boundary model was modified from the widespread conventional moving boundary model, so inherent problems of the conventional model still exist.

Table 6-1. Parameters of the condenser

Parameters	Values
Inlet ammonia mass fraction, x_{ci}	0.9980
Condenser pressure, P_c [kPa]	1791.9
Inlet mass flow rate, $\dot{m}_{ci}$ [kg/s]	0.0618
Secondary flow mass flow rate, $\dot{m}_{ca}$ [kg/s]	0.6442
Condenser length, L_c [m]	1.3335
Condenser tube area, A_c [m^2]	0.0041
Universal heat transfer coefficient, $(UD)_{c1}$ [kW/m·K]	3.0970
Universal heat transfer coefficient, $(UD)_{c2}$ [kW/m·K]	1.9100
Step input: Inlet ammonia mass fraction, x_{cif}	0.9990
Step input: Inlet mass flow rate, $\dot{m}_{cif}$ [kg/s]	0.0680
Step input: Secondary flow mass flow rate, $\dot{m}_{caf}$ [kg/s]	0.7087

Table 6-2. Parameters of the absorber

Parameters	Values
Inlet ammonia mass fraction, x_{ai}	0.6200
Absorber pressure, P_a [kPa]	833.96
Inlet mass flow rate, $\dot{m}_{ai}$ [kg/s]	0.2166
Inlet enthalpy, h_{ai} [kJ/kg]	330.69
Secondary flow mass flow rate, $\dot{m}_{aa}$ [kg/s]	1.6584
Absorber length, L_a [m]	1.3208
Absorber tube area, A_a [m^2]	0.0079
Universal heat transfer coefficient, $(UD)_{a1}$ [kW/m·K]	3.0900
Universal heat transfer coefficient, $(UD)_{a2}$ [kW/m·K]	4.2690
Step input: Inlet ammonia mass fraction, x_{aif}	0.6262
Step input: Inlet enthalpy, h_{aif} [kJ/kg]	333.99
Step input: Inlet mass flow rate, $\dot{m}_{aif}$ [kg/s]	0.2188
Step input: Secondary flow mass flow rate, $\dot{m}_{aaf}$ [kg/s]	1.6749

Table 6-3. Parameters of the generator

Parameters	Values
Inlet ammonia mass fraction, x_{gi}	0.6200
Generator pressure, P_g [kPa]	1791.9
Inlet mass flow rate, $\dot{m}_{gi}$ [kg/s]	0.1860
Secondary flow mass flow rate, $\dot{m}_{ga}$ [kg/s]	1.1529
Generator length, L_g [m]	0.6096
Generator tube area, A_g [m^2]	0.0411
Universal heat transfer coefficient, $(UD)_{g1}$ [kW/m·K]	0.9657
Universal heat transfer coefficient, $(UD)_{g2}$ [kW/m·K]	0.8539
Step input: Inlet ammonia mass fraction, x_{gif}	0.6262
Step input: Inlet mass flow rate, $\dot{m}_{gif}$ [kg/s]	0.1879

Table 6-4. Parameters of the evaporator

Parameters	Values
Inlet ammonia mass fraction, x_{ei}	0.9980
Evaporator pressure, P_e [kPa]	833.96
Inlet mass flow rate, $\dot{m}_{ei}$ [kg/s]	0.0618
Inlet enthalpy, h_{ei} [kJ/kg]	125.27
Secondary flow mass flow rate, $\dot{m}_{ea}$ [kg/s]	1.1529
Evaporator length, L_e [m]	0.9144
Evaporator tube area, A_e [m^2]	0.0592
Universal heat transfer coefficient, $(UA)_e$ [kW/K]	2.7840
Step input: Inlet ammonia mass fraction, x_{eif}	0.9990
Step input: Inlet enthalpy, h_{eif} [kJ/kg]	137.80
Step input: Inlet mass flow rate, $\dot{m}_{eif}$ [kg/s]	0.0680

Table 6-5. Parameters of SHX

Parameters	Values
Inlet ammonia mass fraction for colder side, x_{sci}	0.6200
Inlet ammonia mass fraction for hotter side, x_{shi}	0.4690
SHX pressure, P_s [kPa]	1791.9
Inlet mass flow rate for colder side, $\dot{m}_{sci}$ [kg/s]	0.1860
Outlet mass flow rate for hotter side, $\dot{m}_{sho}$ [kg/s]	0.1548
SHX length, L_s [m]	2.0320
SHX tube area, A_s [m^2]	0.0027
Universal heat transfer coefficient, $(UA)_s$ [kW/m·K]	1.7958
Step input: Inlet ammonia mass fraction, x_{scif}	0.6820
Step input: Inlet mass flow rate, $\dot{m}_{scif}$ [kg/s]	0.2046

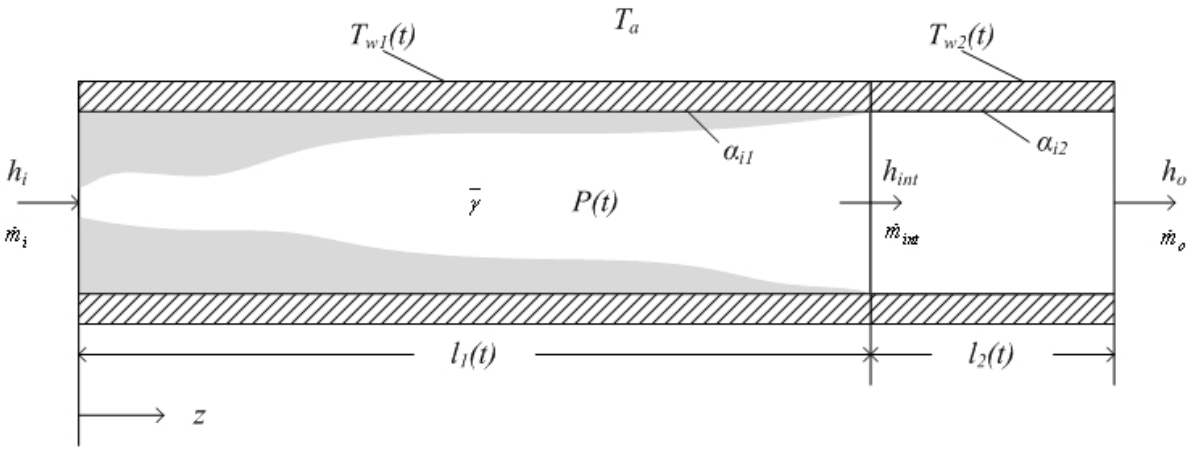

Figure 6-1. A schematic of an evaporator model for single component flow

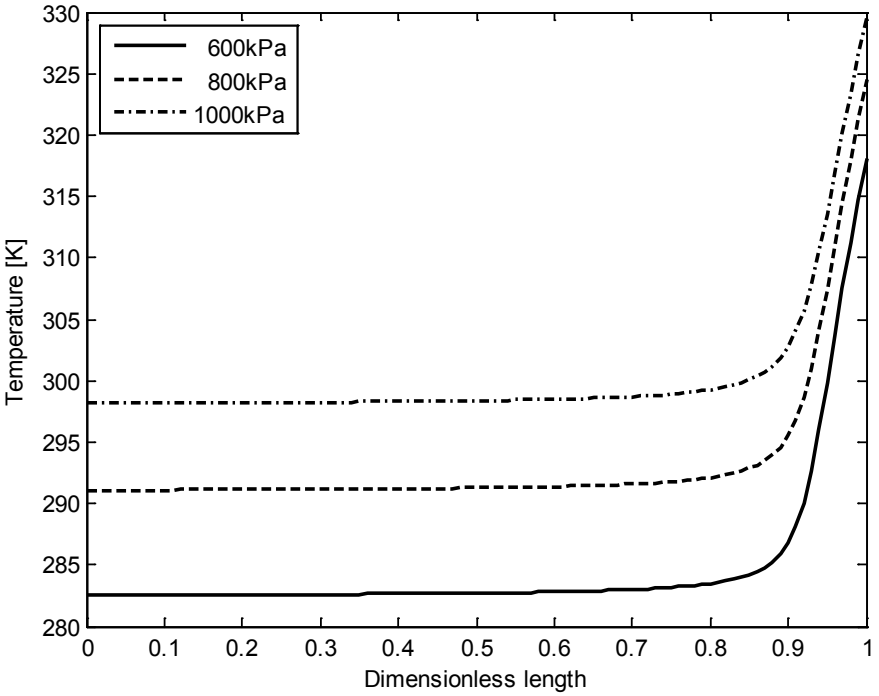

Figure 6-2. Two-phase flow temperature distribution in an evaporator

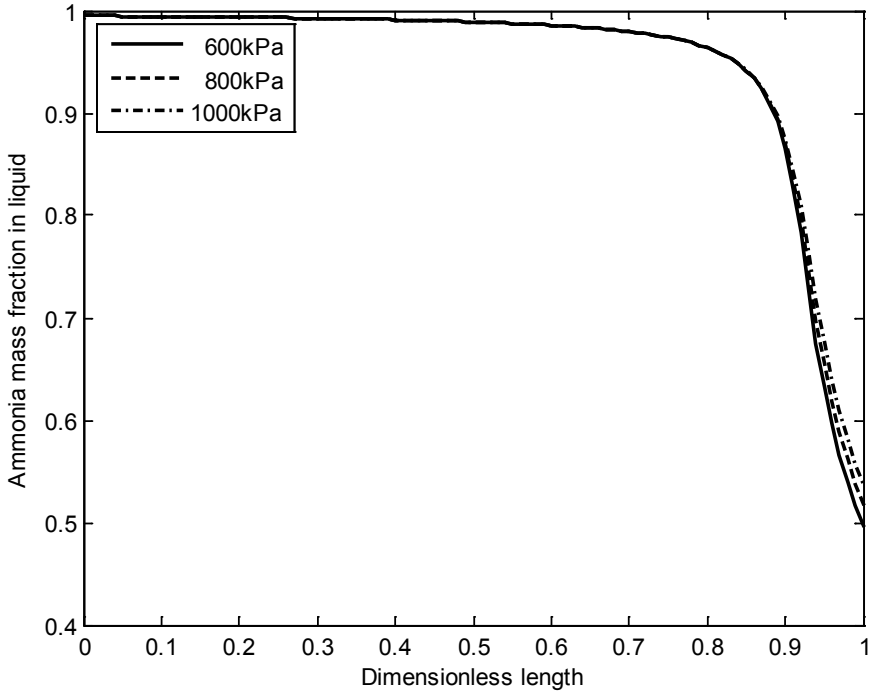

Figure 6-3. Saturated vapor ammonia mass fraction in an evaporator

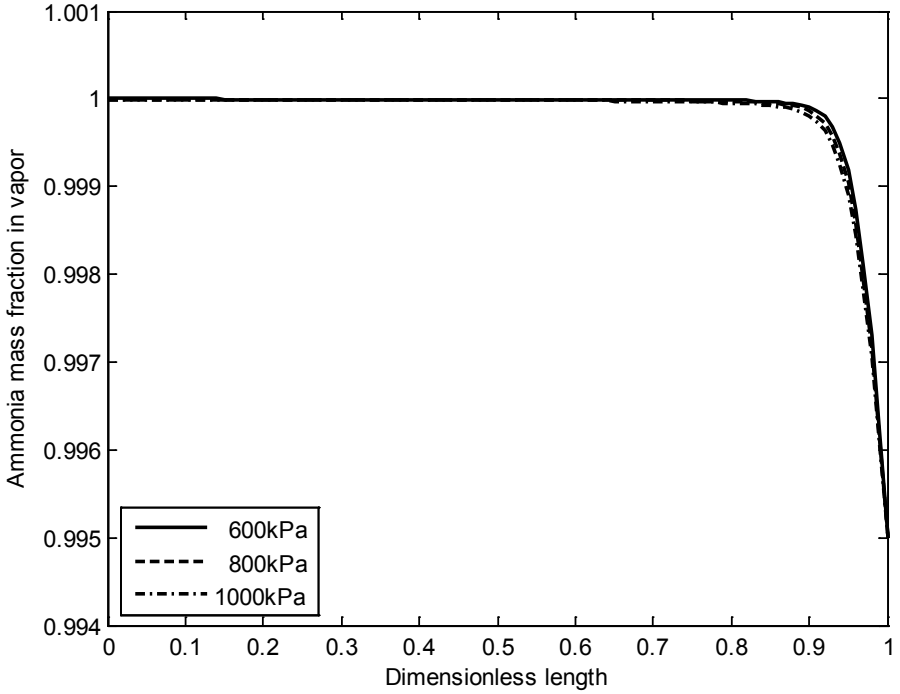

Figure 6-4. Saturated liquid ammonia mass fraction in an evaporator

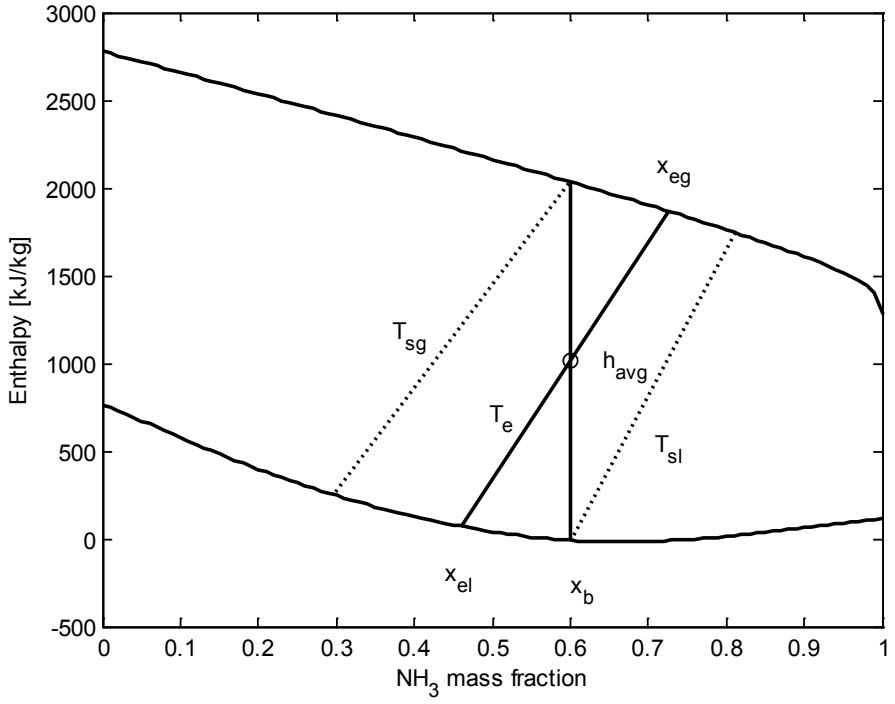

Figure 6-5. Enthalpy-Ammonia mass fraction (h-x) diagram

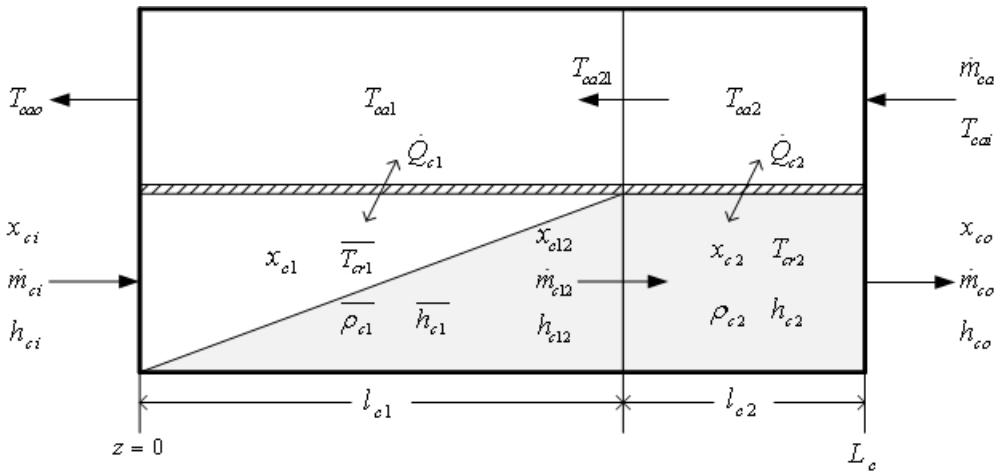

Figure 6-6. A schematic of a condenser

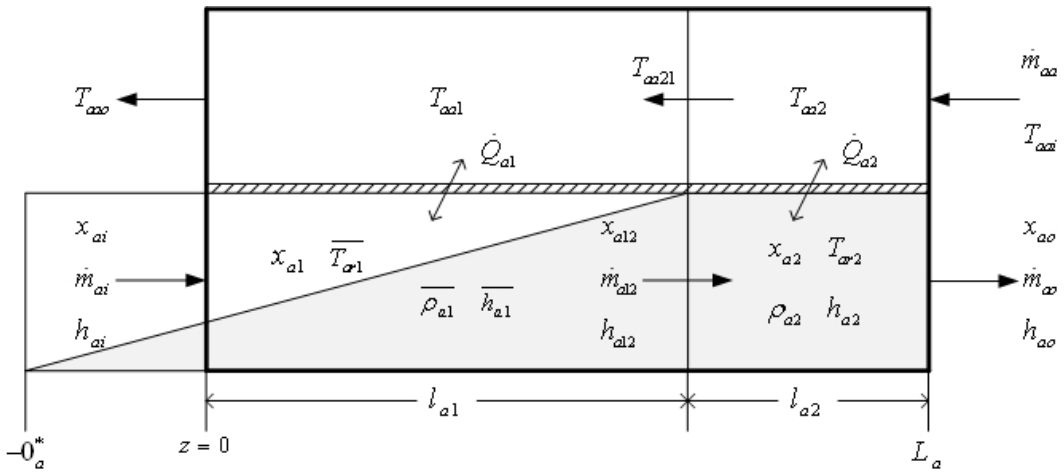

Figure 6-7. A schematic of an absorber

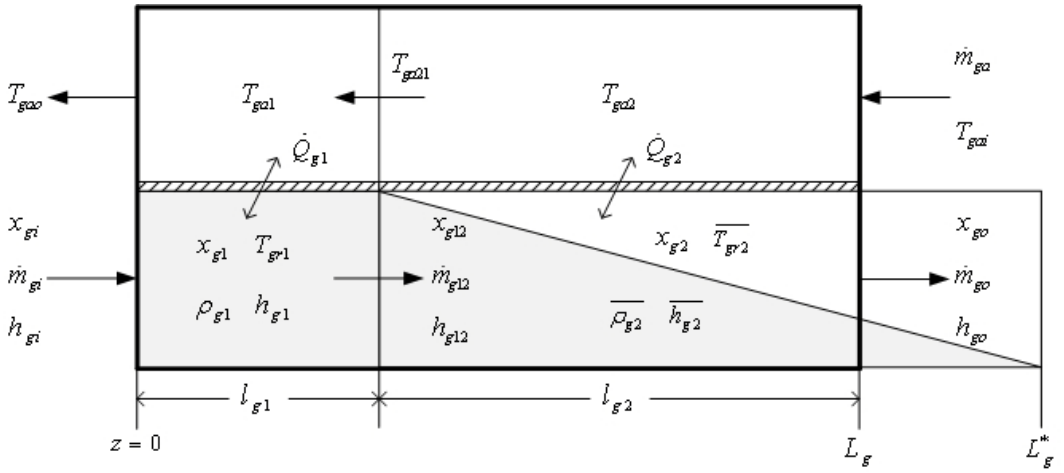

Figure 6-8. A schematic of a generator

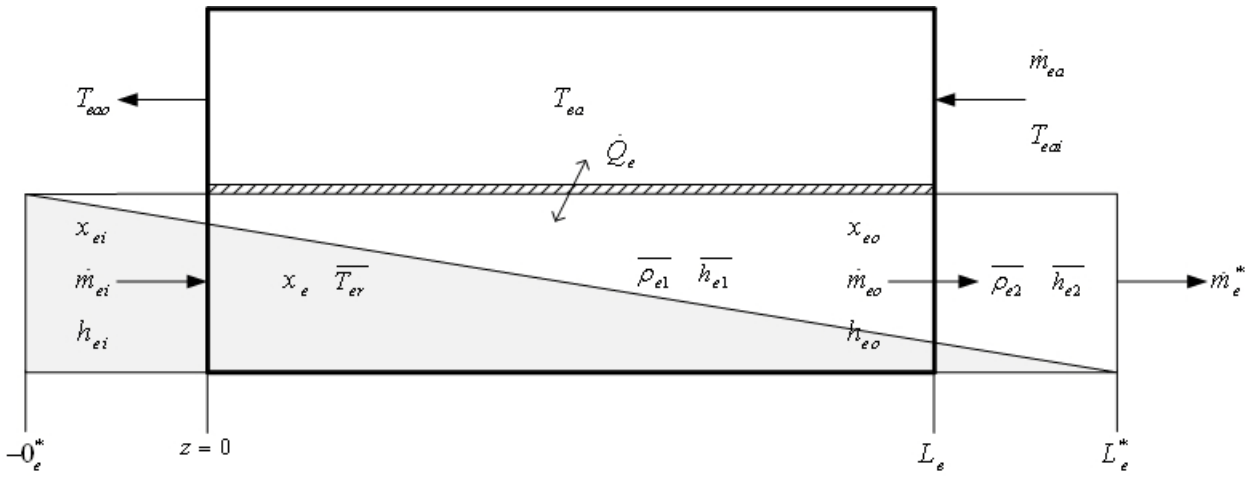

Figure 6-9. A schematic of an evaporator

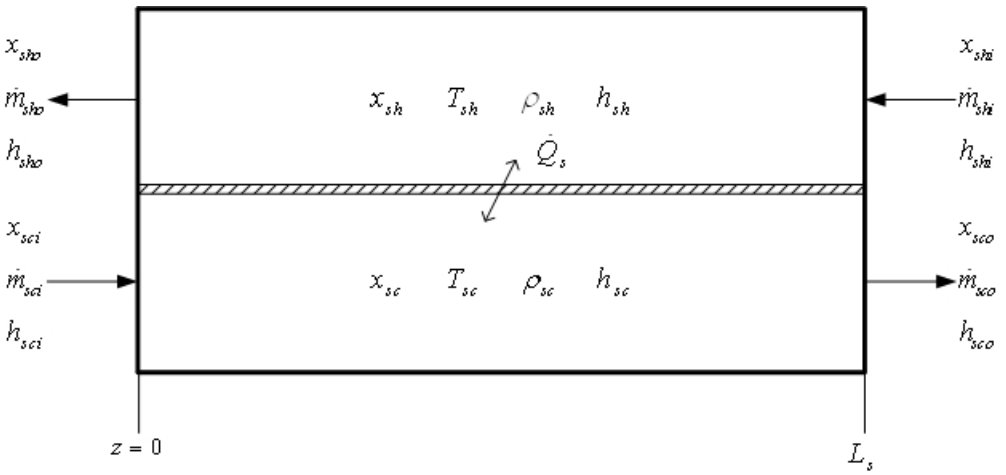

Figure 6-10. A schematic of SHX

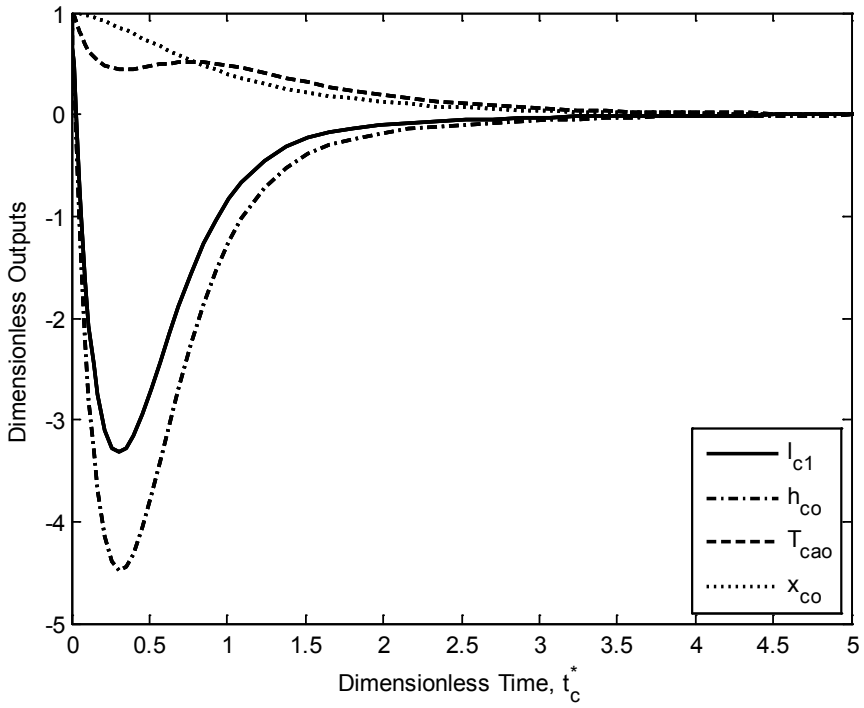

Figure 6-11. Inlet ammonia mass fraction step increase for the condenser

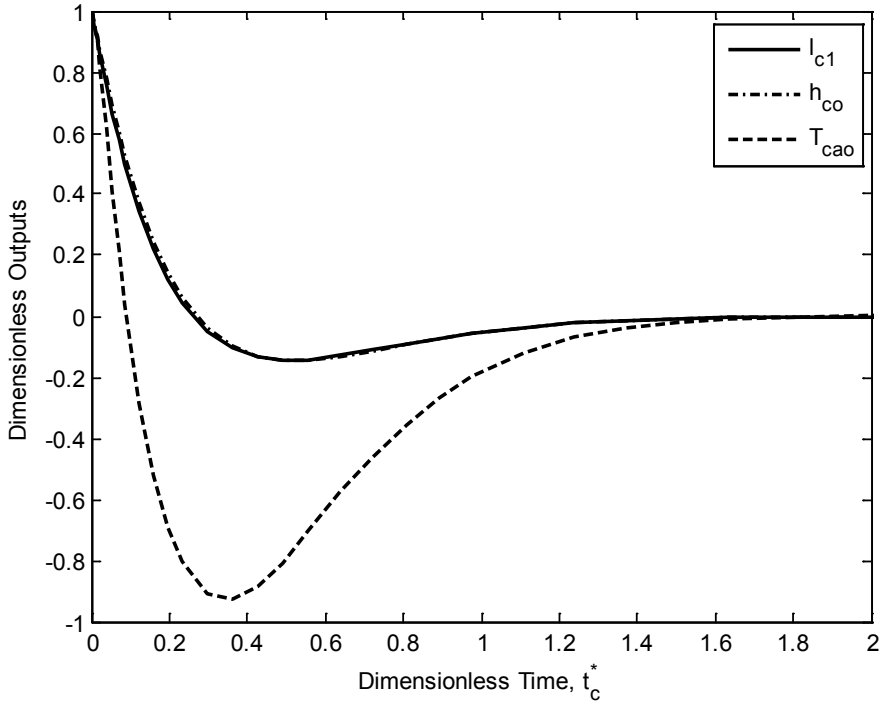

Figure 6-12. Inlet mass flow rate step increase for the condenser

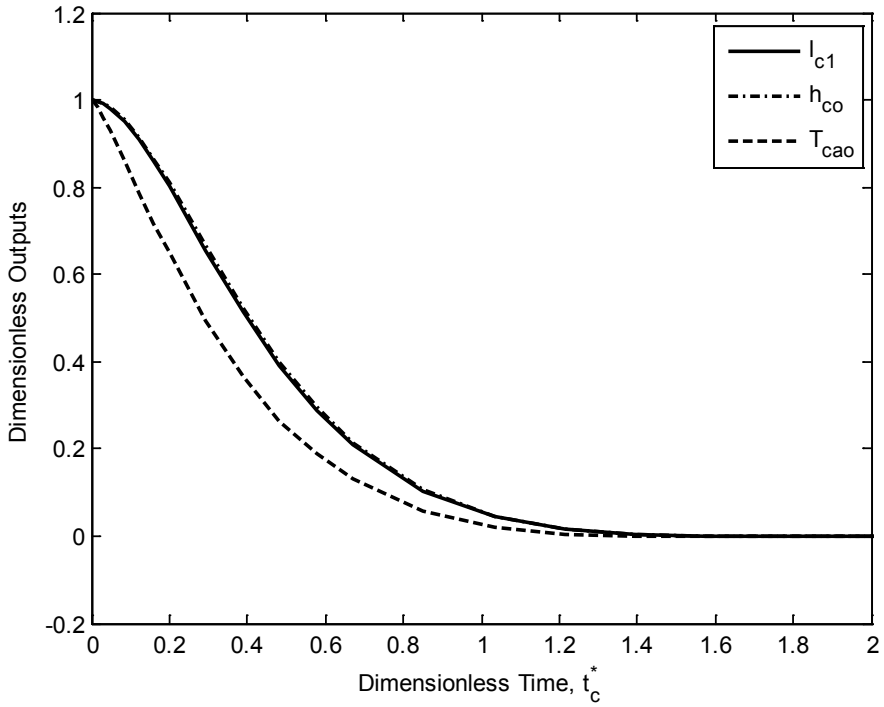

Figure 6-13. Secondary fluid inlet mass flow rate step increase for the condenser

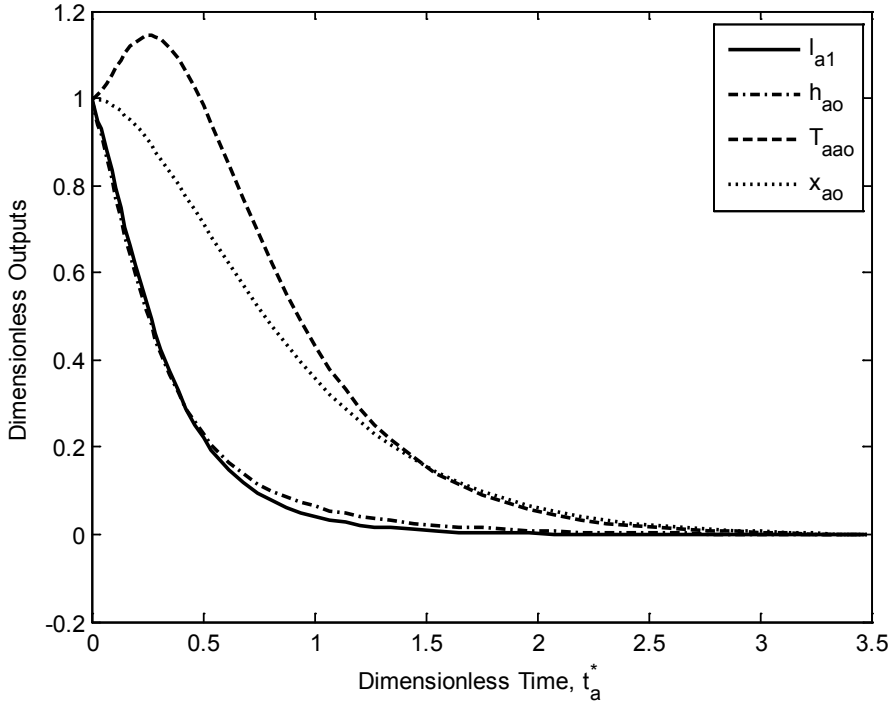

Figure 6-14. Inlet ammonia mass fraction step increase for the absorber

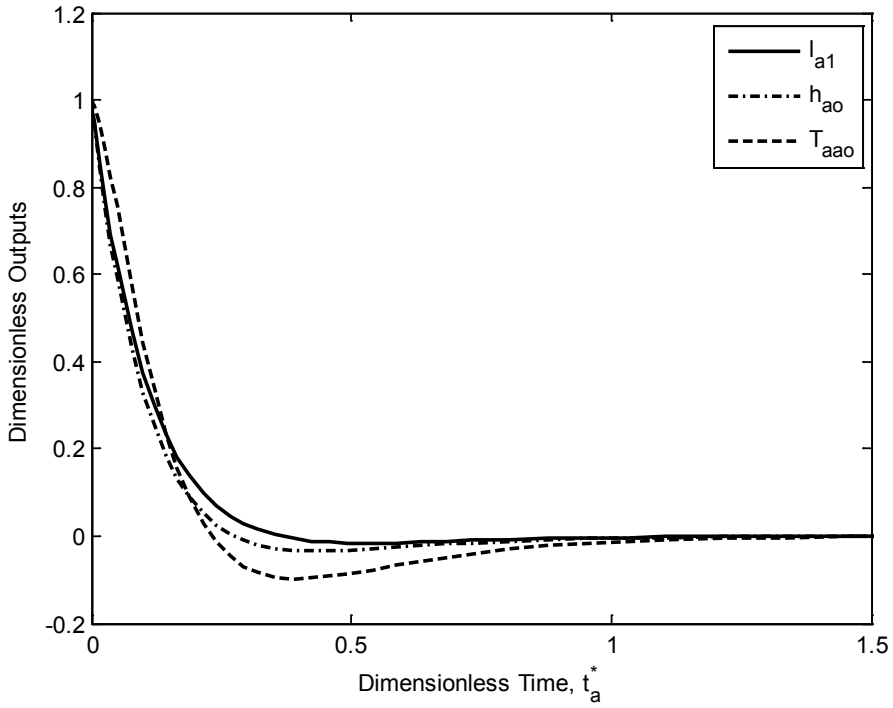

Figure 6-15. Inlet enthalpy step increase for the absorber

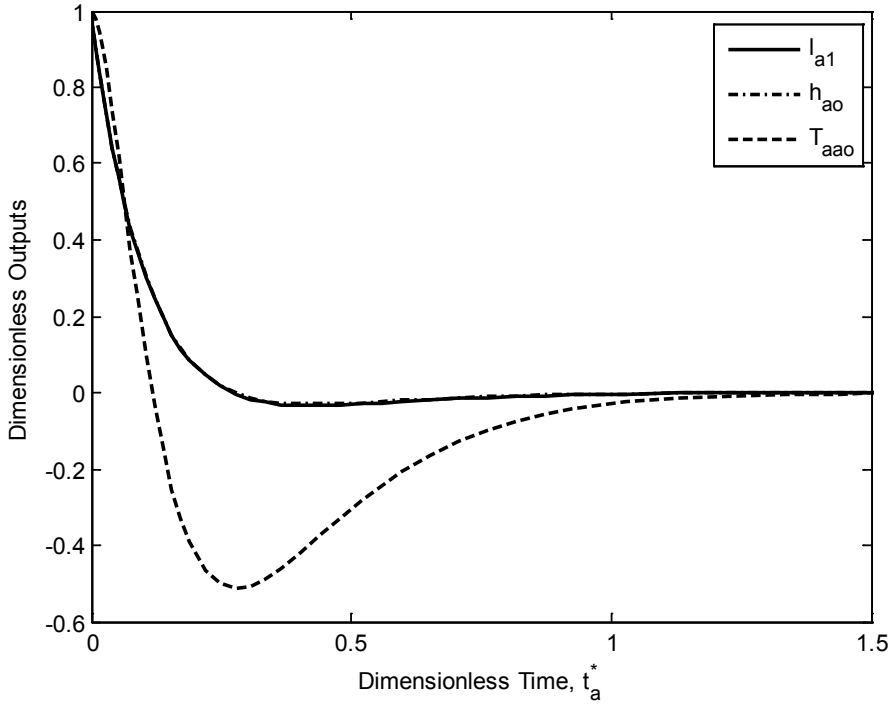

Figure 6-16. Inlet mass flow rate step increase for the absorber

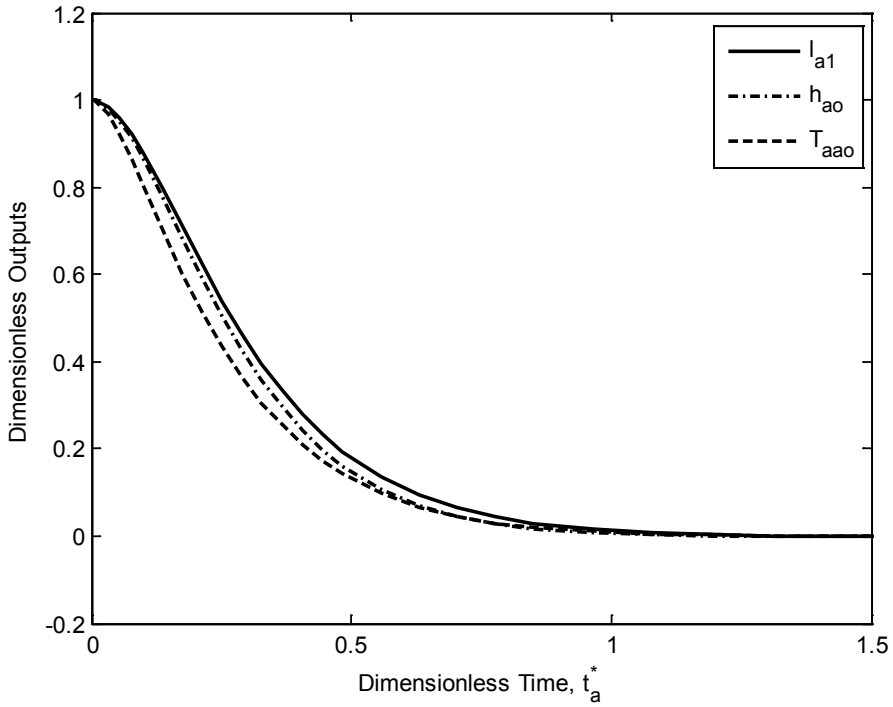

Figure 6-17. Secondary fluid inlet mass flow rate step increase for the absorber

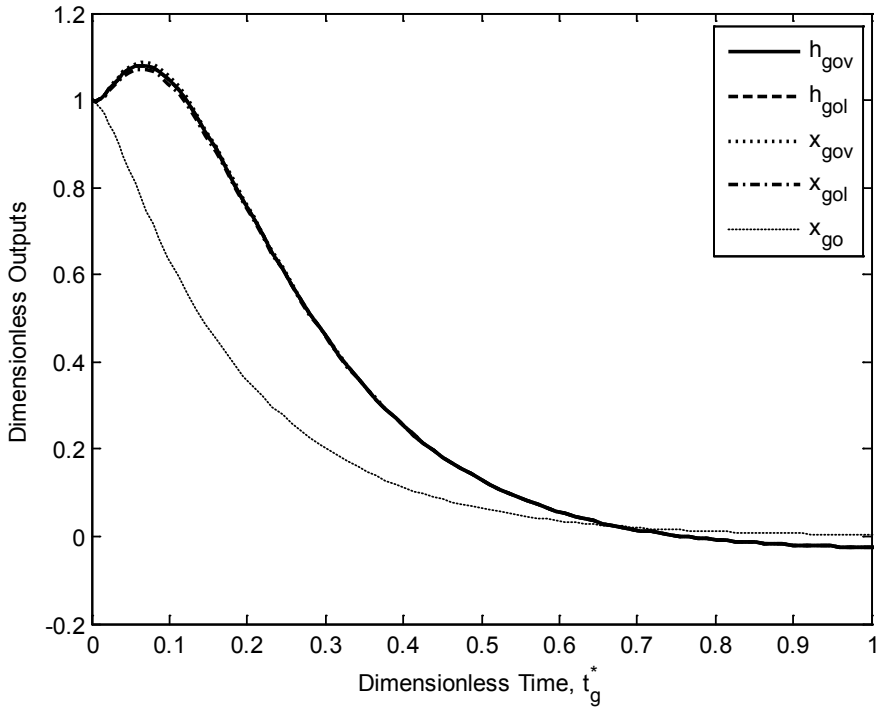

Figure 6-18. Inlet ammonia mass fraction step increase for the generator

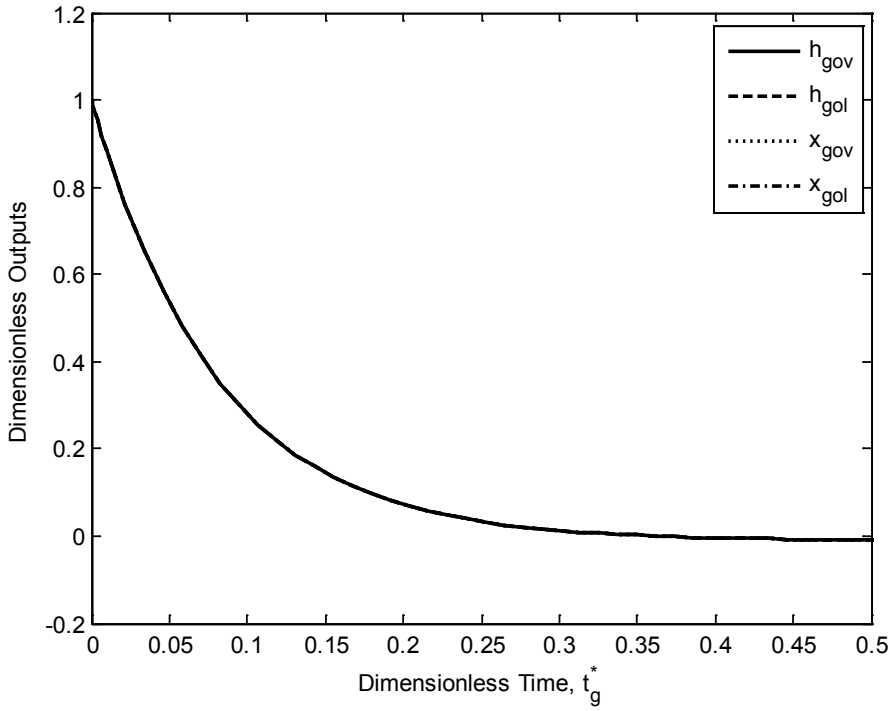

Figure 6-19. Inlet mass flow rate step increase for the generator

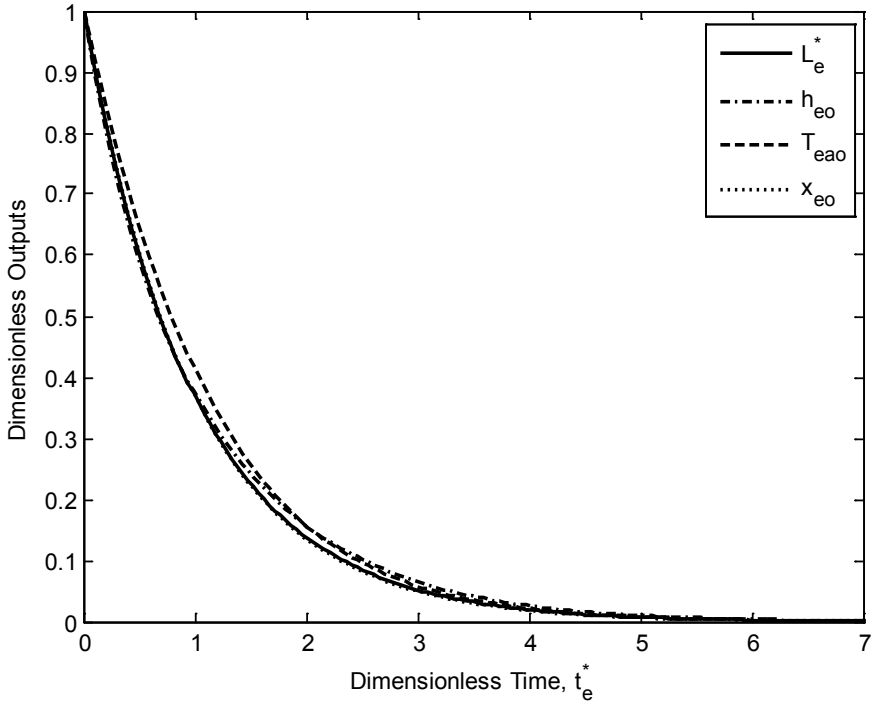

Figure 6-20. Inlet ammonia mass fraction step increase for the evaporator

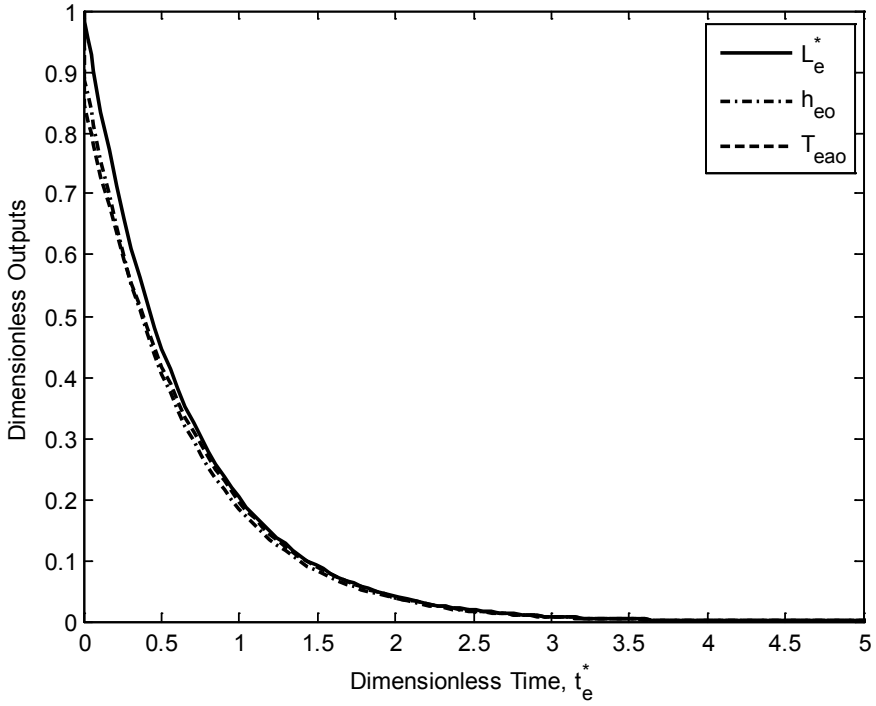

Figure 6-21. Inlet enthalpy step increase for the evaporator

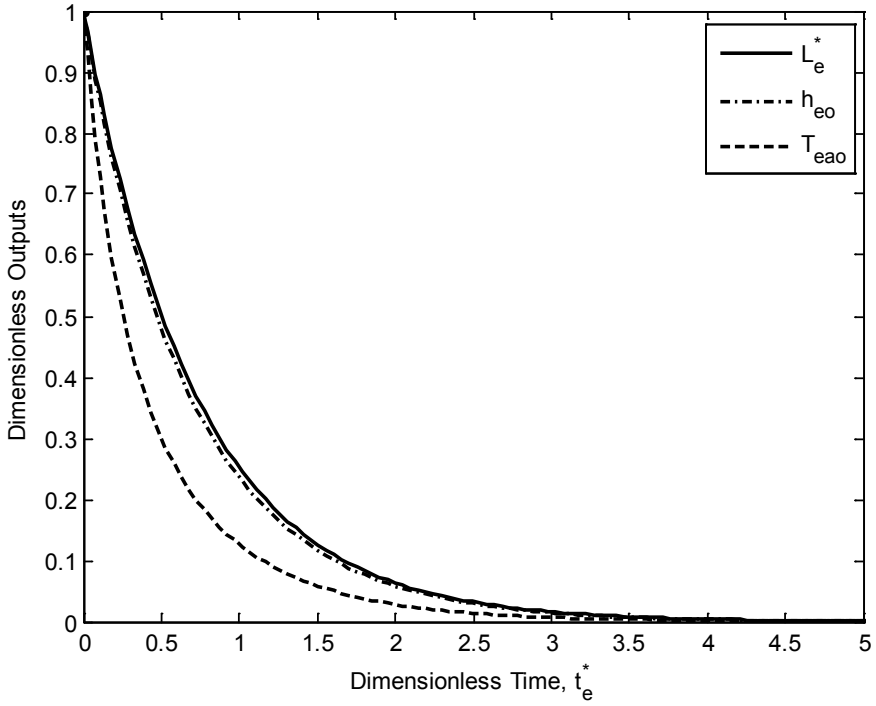

Figure 6-22. Inlet mass flow rate step increase for the evaporator

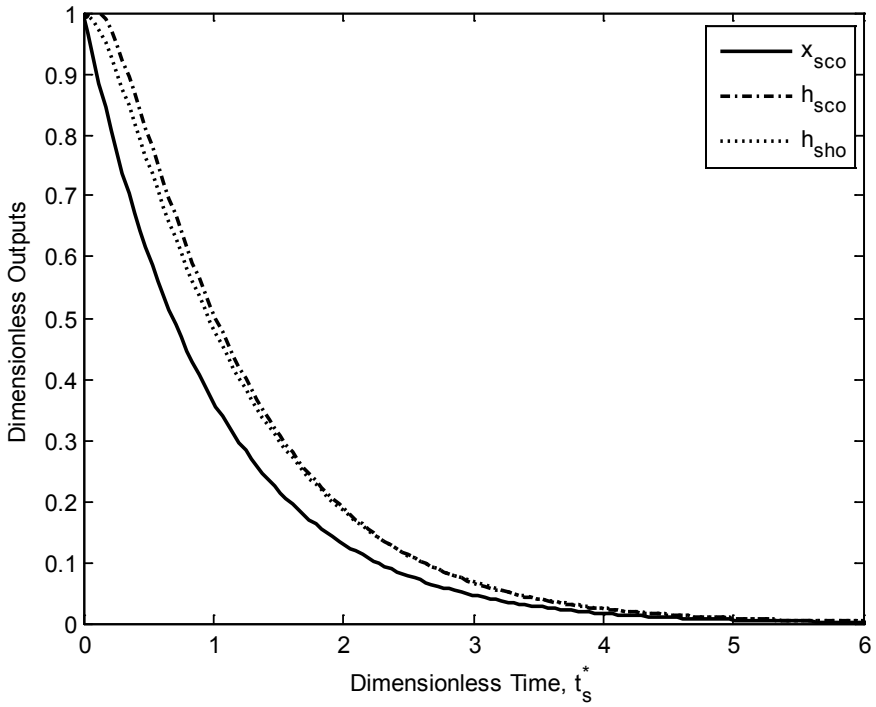

Figure 6-23. Colder side inlet ammonia mass fraction step increase for SHX

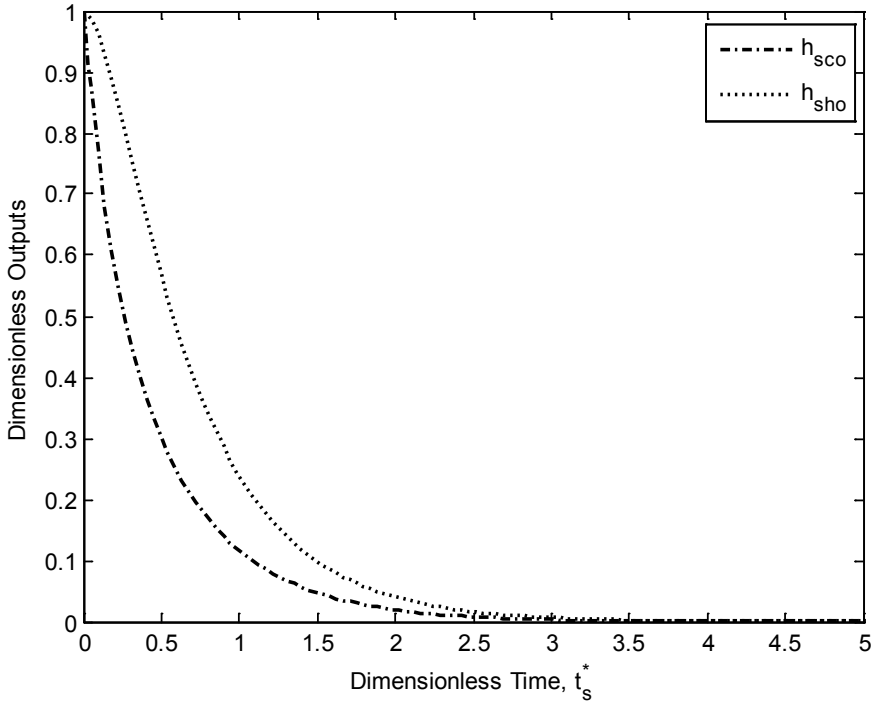

Figure 6-24. Colder side inlet mass flow rate step increase for SHX

CHAPTER 7
CONCLUSIONS

The PoWER system has been described, including its system attributes and dynamic challenges, as motivation for this work towards development of physics-based controls. In the current research, we focused primarily on the steady-state and dynamic modeling of the VARS appropriate for development of overall system controls algorithms. Furthermore, for accurate modeling, thermodynamic properties of ammonia-water mixtures have been studied and critiqued, and a recommended approach has been implemented for overcoming the limitations of current curvefit-based models.

Thermodynamic Properties of Ammonia-Water Mixtures

Several commonly used P-T-x relations have been studied to avoid time-consuming iterative numerical calculation. Two high-order fitted relationships and tabulated data with interpolation have been compared with experimental data. High-order fitted P-T-x equations are:

- easy to program and their execution time is very short.
- They show very good agreement with experimental data.
- Local oscillatory patterns (Runge's phenomenon) exist, near the pure component limits.
- Bubble and dew point temperature lines do not meet at the pure component limits.

These problems can lead to instability in iterative system solutions, especially in systems with nearly pure ammonia such as the VARS. They may be most useful for rapid calculations in some steady state analyses. However, alternative methods should be considered for programmed iterative calculations. Tabulated data are used in this research with good results. The use of tabulated data and interpolation method show:

- Single values for both bubble and dew point temperature at pure component regime.
- Elimination of Runge's phenomenon.
- Only slightly slower execution time than fitted equations.

Given the memory capabilities of modern computing platforms, it is possible to store and access very large tables with very little cost in execution time. However, to achieve this objective, accurate experimental data over the range on interest in the calculation are necessary.

Preliminary Design of VARS

A conceptual VARS model with an ice-making capability has been designed and analyzed. From this research, it is observed that waste heat from the HPRTE can be utilized effectively with a VARS having three evaporators. Based on a typical daily load demand for a building, cooling capacity has been calculated and distributed to air conditioning and to an ice maker for thermal storage. Ice produced primarily at night can be used as a refrigeration source to supplement the direct air conditioning from the VARS during the heat of the day. Initial modeling of the VARS used the Second Law efficiency, which produced the following conclusions:

- When the generator capacity is increased, the gas turbine subsystem evaporator can be downsized, as additional heat from generator is better used in air conditioning and ice-making.
- The ice can be produced when air conditioning load is low and it can be used to chill water to cool the PoWER system, thereby increasing efficiency, or for additional air conditioning.
- The energy storage in daily ice production is ideally approximately twice the total daily air conditioning capacity.
- This additional cooling capacity could be used to air condition a second building of similar size to the design building, or used in a load leveling manner to downsize the PoWER system.

Considering the combination of attributes listed above, it can be concluded that the PoWER system seems to be a promising candidate for distributed power generation, especially in hot climates where the summer load-leveling benefits are important.

A discretized dynamic VARS model has been developed under the assumption of uniform ammonia mass fraction, though initially applied to a quasi-steady problem. Heat exchangers have been spatially and temporally discretized and the governing differential equations thereby converted to algebraic form. From this research, a detailed multi-stage VARS model has been developed and the results have been compared with the Second Law analysis model. Comparing these two steady-state analyses, the following conclusions may be reached:

- The Second Law analysis is more convenient approach and is effective for preliminary design.
- The analysis is useful for comparing this concept to competing options and for conceptual design when the details of the VARS have necessaryily not yet been specified.

Dynamic Modeling of Heat Exchangers

To model two-component, two-phase, dynamic heat exchangers, the conventional moving boundary model has been extended. Some modifications are needed for two-component fluids to allow two component fluids, removing the single-component limitation of the conventional model. It has been clearly shown that the moving boundary model satisfies both two-phase flow physics and the requirements of control design.

A dynamic modeling method of heat exchangers for control applications has been discussed, as presented in the literature. To satisfy both two-phase flow physics and the

control design requirements, the conventional moving boundary model has been found to have the following characteristics:

- It is a one-dimensional two-phase dynamic heat exchanger model.
- The interface between two-phase and single-phase moves in response to time-varying boundary conditions.
- Void fraction is always assumed constant in the two-phase region.
- The model is capable of only single-component flow and it is limited to the condition that the moving boundary is inside of the heat exchanger.

To extend the conventional moving boundary model for an ammonia-water VARS, species balances are added to the governing equation set. From the three modeling options considered, the spatial mean value model is employed and steady-state calculation data, functions of pressure and ammonia mass fraction, are used in tabulated form for the two-phase region. A reference, or imaginary, moving boundary, which may fall upstream or downstream of the heat exchanger, has been successfully introduced as a means of handling moving boundary problems with two-phase flow at the inlet or exit.

- Tabulated data was found to significantly reduce complexity of the conservation equation set.

However, this newly-developed moving boundary model was modified from the widespread conventional moving boundary model, so inherent problems of the conventional model still exist.

Future Work

The ultimate goal of this research is to enable development of advanced control of the PoWER system. For system control, linear dynamic models of the system and more

accurate thermodynamic properties of the working fluid are necessary. To improve the model, following research should be done:

- Thermodynamic consistency test with existing and newly-found thermodynamic properties of ammonia-water mixtures.
- Non-equilibrium modeling for two-phase ammonia-water mixtures during transient state.
- Improved moving boundary models for single-phase flow to reduce or eliminate the immediate time response at the single-phase exit.
- Complete cycle modeling for a VARS including rectifier, storage, pump, and expansion valves.

Once the above-mentioned research is finished, the nonlinear dynamic model for a VARS can be linearized to develop a robust control algorithm.

APPENDIX A
COMPUTER CODE FOR THEMODYNAMIC PROPERTIES OF AMMONIA-WATER MIXTURES

The results shown in Chapter 3 are based on following computer codes. The necessary files to obtain the results are shown at Table A-1 with descriptions.

Table A-1. Matlab files for thermodynamic properties of ammonia-water mixtures

File name	Descriptions
bogart_si.mat	Bogart P-T-x tables in SI unit The file consists of 5 tables • P_bog: 24 pressure ranges • Tb_bog: bubble point temperature table • Td_bog: dew point temperature table • x_bog: ammonia mass fraction for Tb_bog • y_bog: ammonia mass fraction for Td_bog
Tb_Bogart.m	Bubble point temperature from pressure and liquid ammonia mass fraction
Td_Bogart.m	Dew point temperature from pressure and vapor ammonia mass fraction
y_Bogart.m	Vapor ammonia mass fraction from pressure and liquid ammonia mass fraction
liqprop_ibrahim.m	Specific enthalpy, entropy, and volume from temperature, pressure, and liquid ammonia mass fraction
vapprop_ibrahim.m	Specific enthalpy, entropy, and volume from temperature, pressure, and vapor ammonia mass fraction

The files, Tb_Bogart.m, Td_Bogart.m, and y_Bogart.m use Bogart tables with two-dimensional linear interpolation method for P-T-x of ammonia-water mixtures. For specific enthalpy, entropy, and volume, Ibrahim's equations [15] are used for liqprop_ibrahim.m and vapprop_ibrahim.m.

Bubble and dew point temperatures are fundtions of pressure and ammonia concentration. Scalar values of pressure and ammonia mass fraction are used as inputs for both Tb_Bogart.m and Td_Bogart.m. Consider P, x, and y as pressure and ammonia mass fraction in saturated liquid and vapor. Their temperatures are easily obtained by

typing Tb_Bogart(P,x) or Td_Bogart(P,y) on Matlab command window if P, x, and y are memorized in Matlab Workspace. Pressure is in kPa and concentration is in kg/kg and resulting temperature is in Kelvin. To get ammonia mass fraction in saturated vapor, type y_Bogart(P,x).

Ibrahim's relations use T (temperature), P, and x or y as inputs. To use the m files, liqprop_ibrahim.m and vapprop_ibrahim.m, typing order for inputs is important. Temperature is always the first followed by P and x. Output order is specific enthalpy, entropy, and volume. For example, if saturated vapor enthalpy is interested, simply type vapprop_ibrahim(Td_Bogart(P,y),P,y) on the command window.

Matlab code for each file is shown below:

- Tb_Bogart.m

```
% Tb_Bogart.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Tb flag1] = Tb_Bogart(P,x)

load bogart_si.mat

if P > max(P_bog)
    disp('pressure is too high, the max pressure is 2068.4271kPa = 300psia')
    Tb = [];
    flag1 = 0;  % To stop simulation
elseif P < min(P_bog)
    disp('pressure is too low, the min pressure is 41.368542kPa = 6psia')
    Tb = [];
    flag1 = 0;
elseif x > 1
    disp('Ammonia concentration cannot be higher than 1')
    Tb = [];
    flag1 = 0;
elseif x < 0
    disp('Ammonia concentration cannot be lower than 0')
    Tb = [];
    flag1 = 0;
else
    Tb = interp2(x_bog,P_bog,Tb_bog,x,P,'linear');
    flag1 = 1;
end
% End of Tb_Bogart.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- Td_Bogart.m

```
% Td_Bogart.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Td flag1]= Td_Bogart(P,y)

load bogart_si.mat

if P > max(P_bog)
    disp('pressure is too high, the max pressure is 2068.4271kPa = 300psia')
    Td = [];
    flag1 = 0;  % To stop simulation
elseif P < min(P_bog)
    disp('pressure is too low, the min pressure is 41.368542kPa = 6psia')
    Td = [];
    flag1 = 0;
elseif y > 1
    disp('Ammonia concentration cannot be higher than 1')
    Td = [];
    flag1 = 0;
elseif y < 0
    disp('Ammonia concentration cannot be lower than 0')
    Td = [];
    flag1 = 0;
else
    Td = interp2(y_bog,P_bog,Td_bog,y,P,'linear');
    flag1 = 1;
end
% End of Td_Bogart.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- y_Bogart.m

```
% y_Bogart.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [y flag1] = y_Bogart(P,x)

load bogart_si.mat

if P > max(P_bog)
    disp('pressure is too high, the max pressure is 2068.4271kPa = 300psia')
    y = [];
    flag1 = 0;  % To stop simulation
elseif P < min(P_bog)
    disp('pressure is too low, the min pressure is 41.368542kPa = 6psia')
    y = [];
    flag1 = 0;
elseif x > 1
    disp('Ammonia concentration cannot be higher than 1')
    y = [];
    flag1 = 0;
elseif x < 0
    disp('Ammonia concentration cannot be lower than 0')
    y = [];
    flag1 = 0;
else
    y = interp2(x_bog,P_bog,y_bog,x,P,'linear');
```

```
end
% End of y_Bogart.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- liqprop_ibrahim.m

```
% liqprop_ibrahim.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [hl sl vl] = liqprop_ibrahim(T,P,x)

% Ibrahim thermodynamic property of the liquid mixture
%
% Input:    T[K],           Temperature (scalar)
%           P[kPa],         Pressure (scalar)
%           x[kg/kg],       NH3 mass fraction (scalar btw 0 & 1)
% Output:   hl[kJ/kg],      Specific enthalpy
%           sl[kJ/kg/K],    Specific entropy
%           vl[m3/kg],      Specific volume
%
% Ver. 0, 20080205
% Ver. 1, 20080221 - enthalpy eliminated (to use Patek's enthalpy)
% Ver. 2, 20090604 - enthalpy re-use, x in row vector
% Ver, 3, 20090615 - x and results in a scalar (no need to be a vector)
% made by Choon Jae Ryu: choonjae.ryu@gmail.com
% Energy and Gas Dynamics Lab.
% Mechanical and Aerospace Engineering, University of Florida
%
% Data and equations from O.M. Ibrahim, S.A. Klein,
% "Thermodynamic Properties of Ammonia-Water Mixtures",
% ASHRAE Transactions: Symposia, Vol.99, pp.1495-1502, 1993

cmol = 18.015268*x./(0.985008*x+17.03026);   %[kmol/kmol] mole fraction
mwmix = cmol*17.03026 + (1-cmol)*18.015268; %[kg/kmol] mix. mole. weight
pbar = P/100; %[bar]

tb = 100;   %[K] reference temperature
pb = 10;    %[bar] reference pressure
R = 8.314;  %[kJ/kmol/K] gas constant

coliq = [3.971423e-2    2.748796e-2;
         -1.790557e-5   -1.016665e-5;
         -1.308905e-2   -4.452025e-3;
         3.752836e-3    8.389246e-4;
         1.634519e1     1.214557e1;
         -6.508119      -1.898065;
         1.448937       2.911966e-1;
         4.878573       21.821141;
         1.644773       5.733498;
         3.2252         5.0705;
         2              3          ];

exc = [-41.733398   0.02414     6.702285    -0.011475 ...
       63.608967    -62.490768  1.761064    0.008626 ...
       0.387983     -0.004772   -4.648107   0.836376 ...
       -3.553627    0.000904    24.361723   -20.736547];
```

```
pr = pbar/pb; %dimensionless pressure
tr = T/tb;    %dimensionless temperature

hal = -R * tb * (-coliq(8,1) + coliq(5,1) * (coliq(10,1) - tr) ...
                 + coliq(6,1) / 2 * (coliq(10,1)^2 - tr^2) ...
                 + coliq(7,1) / 3 * (coliq(10,1)^3 - tr^3) ...
                 + (coliq(4,1) * tr^2 - coliq(1,1)) * (pr - coliq(11,1)) ...
                 - coliq(2,1) / 2 * (pr^2 - coliq(11,1)^2));
                 % [kJ/kmole] NH3 molar enthalpy

sal = R * (coliq(9,1) + coliq(5,1) * log(tr / coliq(10,1)) ...
           + coliq(6,1) * (tr - coliq(10,1)) ...
           + coliq(7,1) / 2 * (tr^2 - coliq(10,1)^2) ...
           + (coliq(11,1) - pr) * (coliq(3,1) + 2 * coliq(4,1) * tr));
           % [kJ/kmole/K] NH3 molar entropy

val = R * tb / pb / 100 * (coliq(1,1) + coliq(2,1) * pr ...
                           + coliq(3,1) * tr + coliq(4,1) * tr^2);
                           % [m3/kmole], NH3 molar volume

hwl = -R * tb * (-coliq(8,2) + coliq(5,2) * (coliq(10,2) - tr) ...
                 + coliq(6,2) / 2 * (coliq(10,2)^2 - tr^2) ...
                 + coliq(7,2) / 3 * (coliq(10,2)^3 - tr^3) ...
                 + (coliq(4,2) * tr^2 - coliq(1,2)) * (pr - coliq(11,2)) ...
                 - coliq(2,2) / 2 * (pr^2 - coliq(11,2)^2));
                 % [kJ/kmole] H2O molar enthalpy

swl = R * (coliq(9,2) + coliq(5,2) * log(tr / coliq(10,2)) ...
           + coliq(6,2) * (tr - coliq(10,2)) ...
           + coliq(7,2) / 2 * (tr^2 - coliq(10,2)^2) ...
           + (coliq(11,2) - pr) * (coliq(3,2) + 2 * coliq(4,2) * tr));
           % [kJ/kmole/K] H2O molar entropy

vwl = R * tb / pb / 100 * (coliq(1,2) + coliq(2,2) * pr ...
                           + coliq(3,2) * tr + coliq(4,2) * tr^2);
                           % [m3/kmole], H2O molar volume

% Excess properties for liquid only
if (cmol == 0) | (cmol == 1)
    he = 0; se = 0; ve = 0; slmix = 0;
else
    he = R * tb * (1 - cmol) .* (exc(1) + exc(2) * pr + 2 * exc(5) / tr ...
        + 3 * exc(6) / tr^2 + (2 * cmol - 1) .* (exc(7) + exc(8) * pr ...
        + 2 * exc(11) / tr + 3 * exc(12) / tr^2) + (2 * cmol - 1).^2 ...
        * (exc(13) + exc(14) * pr + 2 * exc(15) / tr + 3 * exc(16) / tr^2));

    se = -R * (1 - cmol) .* (exc(3) + exc(4) * pr - exc(5) / tr^2 ...
        - 2 * exc(6) / tr^3 + (2 * cmol - 1) .* (exc(9) + exc(10) * pr ...
        - exc(11) / tr^2 - 2 * exc(12)/tr^3 + (2 * cmol - 1).^2 ...
        * (-exc(15) / tr^2 - 2 * exc(16) / tr^3)));

    ve = R * tb / pb / 100 * (1 - cmol) .* (exc(2) + exc(4) * tr ...
        + (2 * cmol - 1) .* (exc(8) + exc(10) * tr) ...
        + exc(14) * (2 * cmol - 1).^2);
```

```
    slmix = -R * (cmol .* log(cmol) + (1 - cmol) .* log(1 - cmol));
end

hml = cmol * hal + (1 - cmol) * hwl + cmol .* he;  %[kJ/kmol] mix mol
enthalpy
sml = cmol * sal + (1 - cmol) * swl + cmol .* se + slmix;
vml = cmol * val + (1 - cmol) * vwl + cmol .* ve;

hl = hml ./ mwmix; %[kJ/kg] mixture enthalpy
sl = sml ./ mwmix; %[kJ/kg/K]
vl = vml ./ mwmix; %[m3/kg]
% End of liqprop_ibrahim.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- vapprop_ibrahim.m

```
% vapprop_ibrahim.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [hv sv vv] = vapprop_ibrahim(T,P,y)

% Ibrahim thermodynamic property of the vapor mixture
%
% Input:    T[K],           Temperature (scalar)
%           P[kPa],         Pressure (scalar)
%           y[kg/kg],       NH3 mass fraction (scalar between 0 & 1)
% Output:   hv[kJ/kg],      Specific enthalpy
%           sv[kJ/kg/K],    Specific entropy
%           vv[m3/kg],      Specific volume
%
% Ver. 0, 20080205
% Ver. 1, 20080221 - enthalpy eliminated (to use Patek's enthalpy)
% Ver. 2, 20090605 - enthalpy re-use, y in row vector
% Ver, 3, 20090615 - y and results in a scalar (no need to be a vector)
% made by Choon Jae Ryu: choonjae.ryu@gmail.com
% Energy and Gas Dynamics Lab.
% Mechanical and Aerospace Engineering, University of Florida
%
% Data and equations from O.M. Ibrahim, S.A. Klein,
% "Thermodynamic Properties of Ammonia-Water Mixtures",
% ASHRAE Transactions: Symposia, Vol.99, pp.1495-1502, 1993

cmol = 18.015268*y ./ (0.985008*y+17.03026);   %[kmol/kmol] mole fraction
mwmix = cmol*17.03026 + (1-cmol)*18.015268; %[kg/kmol] mix mole weight
pbar=P/100; %[bar]

tb = 100;   %[K] reference temperature
pb = 10;    %[bar] reference pressure
R = 8.314;  %[kJ/kmol/K] gas constant

covap = [-1.049377e-2   2.136131e-2;
         -8.288224      -3.169291e1;
         -6.647257e2    -4.634611e4;
         -3.045352e3    0;
         3.673647       4.019170;
         9.989629e-2    -5.175550e-2;
         3.617622e-2    1.951939e-2;
         26.468879      60.965058;
```

```
         8.339026       13.453430;
         3.2252         5.0705;
         2              3           ];

pr = pbar/pb; %dimensionless pressure
tr = T/tb;    %dimensionless temperature

hav = -R * tb * (-covap(8,1) + covap(5,1) * (covap(10,1) - tr) ...
      + covap(6,1) / 2 * (covap(10,1)^2 - tr^2) ...
      + covap(7,1) / 3 * (covap(10,1)^3 - tr^3) ...
      - covap(1,1) * (pr - covap(11,1)) ...
      + 4 * covap(2,1) * (covap(11,1) / covap(10,1)^3 - pr / tr^3) ...
      + 12 * covap(3,1) * (covap(11,1) / covap(10,1)^11 - pr / tr^11) ...
      + 4 * covap(4,1) * (covap(11,1)^3 / covap(10,1)^11 - pr^3 / tr^11));
      %[kJ/kmol] NH3 enthalpy

sav = -R * (-covap(9,1) + 11 * covap(3,1) ...
      * (covap(11,1) / covap(10,1)^12 - pr / tr^12) ...
      + 11 / 3 * covap(4,1) * (covap(11,1)^3 / covap(10,1)^12 - pr^3 /
tr^12) ...
      + 3 * covap(2,1) * (covap(11,1) / covap(10,1)^4 - pr / tr^4) ...
      + covap(6,1) * (covap(10,1) - tr) + covap(7,1) / 2 ...
      * (covap(10,1)^2 - tr^2) + log(pr / covap(11,1)) ...
      - covap(5,1) * log(tr / covap(10,1))); %[KJ/kmol/K]

vav = R * tb / pb / 100 * (tr / pr + covap(1,1) ...
      + covap(2,1) / tr^3 + covap(3,1) / tr^11 ...
      + covap(4,1) * pr^2 / tr^11); %[m3/kmol]

hwv = -R * tb * (-covap(8,2) + covap(5,2) * (covap(10,2) - tr) ...
      + covap(6,2) / 2 * (covap(10,2)^2 - tr^2) ...
      + covap(7,2) / 3 * (covap(10,2)^3 - tr^3) ...
      - covap(1,2) * (pr - covap(11,2)) ...
      + 4 * covap(2,2) * (covap(11,2) / covap(10,2)^3 - pr / tr^3) ...
      + 12 * covap(3,2) * (covap(11,2) / covap(10,2)^11 - pr / tr^11) ...
      + 4 * covap(4,2) * (covap(11,2)^3 / covap(10,2)^11 - pr^3 / tr^11));
      %[kJ/kmol] H2O enthalpy

swv = -R * (-covap(9,2) + 11 * covap(3,2) ...
      * (covap(11,2) / covap(10,2)^12 - pr / tr^12) ...
      + 11 / 3 * covap(4,2) * (covap(11,2)^3 / covap(10,2)^12 - pr^3 /
tr^12) ...
      + 3 * covap(2,2) * (covap(11,2) / covap(10,2)^4 - pr / tr^4) ...
      + covap(6,2) * (covap(10,2) - tr) + covap(7,2) / 2 ...
      * (covap(10,2)^2 - tr^2) + log(pr / covap(11,2)) ...
      - covap(5,2) * log(tr / covap(10,2))); %[KJ/kmol/K]

vwv = R * tb / pb / 100 * (tr / pr + covap(1,2) ...
      + covap(2,2) / tr^3 + covap(3,2) / tr^11 ...
      + covap(4,2) * pr^2 / tr^11); %[m3/kmol]

if (cmol == 0) | (cmol == 1)
    svmix = 0;
else
    svmix = -R * (cmol .* log(cmol) + (1 - cmol) .* log(1 - cmol));
```

```
end

hmv = cmol*hav + (1-cmol)*hwv;  %[kJ/kmol] mixture molecular enthalpy
smv = cmol*sav + (1-cmol)*swv + svmix; %[kJ/kmol/K] mix mol entropy
vmv = cmol*vav + (1-cmol)*vwv;  %[m3/kmol] mix mol specific volume

hv = hmv ./ mwmix; %[kJ/kg] gas mixture enthalpy
sv = smv ./ mwmix; %[kJ/kg/K] gas mixture entropy
vv = vmv ./ mwmix; %[m3/kg] gas mixture specific volume
% End of vapprop_ibrahim.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B
COMPUTER CODE FOR DYNAMIC MODELING WITH CONSTANT AMMONIA MASS FRACTION APPROACH

For dynamic modeling of VARS with constant ammonia mass fraction approach, the softwares, Simulink and Matlab, are used. To run the simulation, three files are needed as shown in Table B-1. Before running main file, AC_input_md.mdl, vars_es08_03.m needs to be run for system parameters. Main input for the Simulink file is air conditioning load. For the results in Chapter 5, ac_load.mat has been used for air conditioning load. All files are developed using Matlab version 2007. It is confirmed that these files do not work on Matlab version 2009 because 'embedded Matlab function' box do not allow an additional function to be placed in Matlab code.

Table B-1. Matlab and Simulink files for dynamic modeling with constant ammonia mass fraction approach.

File name	Descriptions
ac_load.mat	Air conditioning load for two days with time
vars_es08_03.m	System parameters to run AC_input_md.mdl
AC_input_md.mdl	Main code for dynamic simulation

- vars_es08_03.m

```
% vars_es08_03.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Steady state modeling of VARS for ES08 (Energy Sustainability 2008)
%   Triad ThermoCharger
%   Temperature, and Pressure data are from Aug. 10th 2008
%   Data for each point are average data between 3309.3091, and 3948.228
%   seconds, and solution pump pressure is 260psi = 1792.64kPa, which is
%   relatively higher pressure than normal. (Useally, under 250psi)
%
%   Version 0, Feb/09/2008
%
%   made by Choon Jae Ryu (choonjae.ryu@gmail.com, choonjae.ryu@ufl.edu)
%   Energy and Gas Dynamics Lab. (Prof. William E. Lear)
%   Mechanical and Aerospace Engineering, University of Florida

%   time gap, design value
dt_ds = 60000;          %[sec], 100 minutes to quasi-steady state
dt = 600;               %[sec], 10 minutes
dp_test = 0.98;         % 2% of pressure drop based on test data
```

```
%   Pressures
Pmax = 1792.63682;       %[kPa], 260psi design value
Pmdl = 521;              %[kPa], to make 5oC at Evap1, and Evap2
Pmin = 358;              %[kPa], to make -5oC at Evap3, not available
%during maximum A/C (Evap2) operation mode

%   Chilled water supply temperature
Tcws = 290;              %[K], design value
Tcw8 = 310;              %[K], COND heated coolant water temp.
Tcwabs = 300;            %[K], ABS1, ABS2 heated coolant water temp.

%   HRVG: Heat Recover Vapor Generator
%   Important: This block is the basis of the whole model
%   Input
Qhrvg_gp = 440;          %[kW] = 15.99[TR] from gas path to solution
qhrvg = 665;             %[kJ/kg], design value
m_sol = Qhrvg_gp/qhrvg; %[kg/s], solution mass, design value

%   RECT: Rectifier
%   Input
%   Note: the state of the point 1 must always be saturated vapor
Pds1 = 1723.6893;        %[kPa]
cds1 = 0.998;            %[kg/kg]
m1 = 0.2854*m_sol;       %[kg/s], this value is always fixed.

%   EVAP1: Evaporator 1, linked with Gas Turbine System
%   Input
qevap1 = 1100;           %[kJ/kg], design value

%   EVAP2: Evaporator 2, Air Conditioner
%   Input
qevap2 = 1100;           %[kJ/kg], design value
m2max = m1/3;            %Revp = 0.5
Qev2_max = qevap2*m2max;%[kW], design value

%   EVAP3: Evaporator 3, Ice Maker
%   Input
qevap3 = 1110;           %[kJ/kg], design value

%   Pump (temporary), In this model, there is no pump. Only fixed data are
%   used at state point 15.
Peff = 0.5;              %Pump efficiency
%   It should be determined to use the power of Pump, or not for net energy
%   balance.

%   SHX: Solution heat exchanger
T19 = 365.07;            %[K], design value

%   ABS1: Absorber 1
T14 = 314.96;            %[K], design value

%   ABS2: Absorber 2
T141 = 303.2563;         %[K], design value
% End of vars_es08_03.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- AC_input_md.mdl

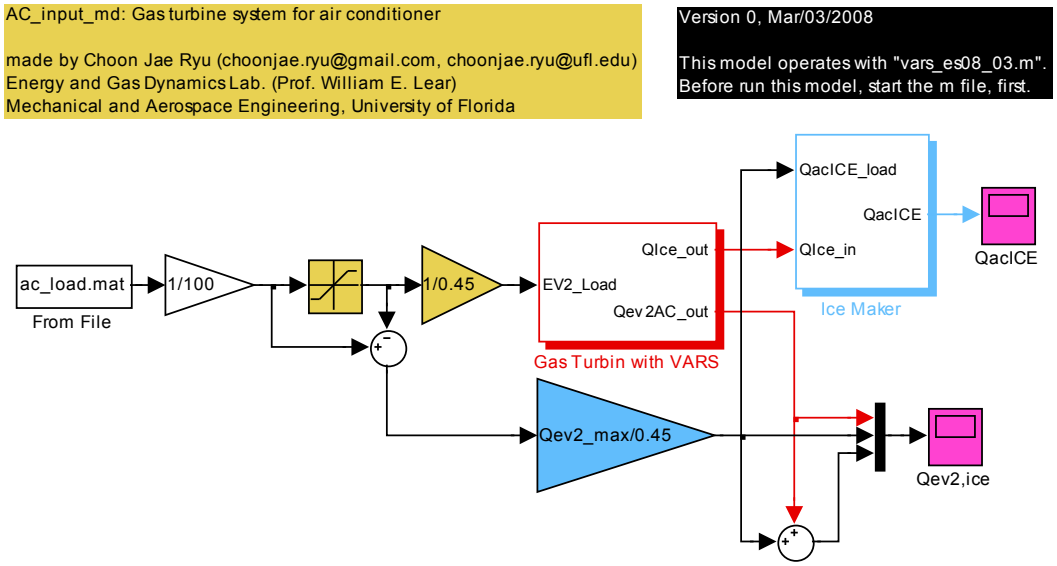

Figure B-1. Simulink model for Chapter 5.

APPENDIX C
GOVERNING EQUATIONS AND COMPUTER CODE FOR MOVING BOUNDARY MODEL

Governing Equations for Moving Boundary Model

Governing equations for each component in a vapor absorption refrigeration system are derived here in detail. Coefficients in matrix form of the conservation equations are also indicated. Leibniz-s rule is often used for differentiation of integrals:

$$\int_{z_2(t)}^{z_1(t)} \frac{\partial f(z,t)}{\partial t} dz = \frac{d}{dt}\int_{z_2(t)}^{z_1(t)} f(z,t)\,dz - f\left(z_2(t),t\right)\frac{dz_2(t)}{dt} + f\left(z_1(t),t\right)\frac{dz_1(t)}{dt} \tag{C-1}$$

Derivations and Coefficients of Single-Component Evaporator Model

For mass balance of control volume 1, integrate Equation (6-1):

$$A\int_0^{l_1} \frac{\partial \rho}{\partial t} dz + \int_0^{l_1} \frac{\partial(\rho A u)}{\partial z} dz = 0 \tag{C-2}$$

where A is the cross sectional area of the evaporator and a constant.

Applying Leibniz's rule to the first term of Equation (C-2) gives:

$$A\left\{\frac{d}{dt}\left(\int_0^{l_1} \rho_1 dz\right) - \rho_{int}\frac{dl_1}{dt}\right\} + (\rho A u)_{int} - (\rho A u)_i = 0 \tag{C-3}$$

This equation is rewritten

$$A\frac{d}{dt}\left(\overline{\rho_1} l_1\right) = (\rho A u)_i - \left\{\rho A\left(u - \frac{dl_1}{dt}\right)\right\}_{int} \tag{C-4}$$

where $\overline{\rho_1} = \dfrac{\int_0^{l_1} \rho_1 dz}{\int_0^{l_1} dz} = \dfrac{\int_0^{l_1} \rho_1 dz}{l_1}$

Because the interface between two-phase and vapor fluids moves back and forth, the mass flow rate of the refrigerant at interface, $\dot{m}_{int}$, is related to the relative speed:

$$\dot{m}_{int} = \left\{ \rho A \left(u - \frac{dl_1}{dt} \right) \right\}_{int} \tag{C-5}$$

Therefore, the mass balance for control volume 1 is

$$A l_1 \frac{d\overline{\rho_1}}{dt} + A\overline{\rho_1} \frac{dl_1}{dt} = \dot{m}_i - \dot{m}_{int} \tag{C-6}$$

Same approach is used for energy balance of control volume 1. Integrate Equation (6-2):

$$A\int_0^{l_1} \frac{\partial(\rho h - P)}{\partial t} dz + \int_0^{l_1} \frac{\partial(\rho A u h)}{\partial z} dz = \int_0^{l_1} \left\{ \alpha_i \pi D_i \left(T_w - T_r \right) \right\} dz \tag{C-7}$$

Apply Leibniz's rule to the first term of Equation (C-7):

$$A\left\{ \frac{d}{dt}\left(\int_0^{l_1} \rho_1 h_1 dz - \int_0^{l_1} P dz \right) - \left(\rho_{int} h_{int} - P \right) \frac{dl_1}{dt} \right\} + \left(\rho A u \right)_{int} h_{int} - \left(\rho A u \right)_i h_i = \alpha_{i1} \pi D_i l_1 \left(T_{w1} - T_{r1} \right) \tag{C-8}$$

This equation is rewritten

$$A\frac{d}{dt}\left\{ \left(\overline{\rho_1 h_1} - P \right) l_1 \right\} + AP\frac{dl_1}{dt} = \left(\rho A u \right)_i h_i - \left\{ \rho A \left(u - \frac{dl_1}{dt} \right) \right\}_{int} h_{int} + \alpha_{i1} \pi D_i l_1 \left(T_{w1} - T_{r1} \right) \tag{C-9}$$

where $\overline{\rho_1 h_1} = \dfrac{\int_0^{l_1} \rho_1 h_1 dz}{\int_0^{l_1} dz} = \dfrac{\int_0^{l_1} \rho_1 h_1 dz}{l_1}$

From Equation (C-5), energy balance for control volume 1:

$$Al_1\frac{d\overline{\rho_1 h_1}}{dt}+A\overline{\rho_1 h_1}\frac{dl_1}{dt}-Al_1\frac{dP}{dt}=\dot{m}_i h_i-\dot{m}_{int}h_{int}+\alpha_{i1}\pi D_i l_1\left(T_{w1}-T_{r1}\right) \tag{C-10}$$

Note that $\overline{\rho_1 h_1}\neq\overline{\rho_1}\cdot\overline{h_1}$ even though they can be close to each other. For control volume 2, mass balance and energy balance after integrating $z=l_1(t)$ to L:

$$A\int_{l_1}^{L}\frac{\partial\rho}{\partial t}dz+\int_{l_1}^{L}\frac{\partial\left(\rho Au\right)}{\partial z}dz=0 \tag{C-11}$$

$$A\int_{l_1}^{L}\frac{\partial\left(\rho h-P\right)}{\partial t}dz+\int_{l_1}^{L}\frac{\partial\left(\rho Auh\right)}{\partial z}dz=\int_{l_1}^{L}\left\{\alpha_i\pi D_i\left(T_w-T_r\right)\right\}dz \tag{C-12}$$

Applying Leibniz's rule to Equations (C-11) and (C-12) gives:

$$A\left\{\frac{d}{dt}\left(\int_{l_1}^{L}\rho_2 dz\right)+\rho_{int}\frac{dl_1}{dt}\right\}+\left(\rho Au\right)_o-\left(\rho Au\right)_{int}=0 \tag{C-13}$$

$$A\left\{\frac{d}{dt}\left(\int_{l_1}^{L}\rho_2 h_2 dz-\int_{l_1}^{L}Pdz\right)+\left(\rho_{int}h_{int}-P\right)\frac{dl_1}{dt}\right\}+\left(\rho Au\right)_o h_o-\left(\rho Au\right)_{int}h_{int}=\alpha_{i2}\pi D_i l_2\left(T_{w2}-T_{r2}\right) \tag{C-14}$$

where $L=l_1+l_2$.

These equations are rewritten after applying Equation (C-5)

$$A\frac{d}{dt}\left(\overline{\rho_2}l_2\right)=\left\{\rho A\left(u-\frac{dl_1}{dt}\right)\right\}_{int}-\left(\rho Au\right)_o \tag{C-15}$$

$$A\frac{d}{dt}\left\{\left(\overline{\rho_2 h_2}-P\right)l_2\right\}+AP\frac{dl_1}{dt}=\left\{\rho A\left(u-\frac{dl_1}{dt}\right)\right\}_{int}h_{int}-\left(\rho Au\right)_o h_o+\alpha_{i2}\pi D_i l_2\left(T_{w2}-T_{r2}\right) \tag{C-16}$$

where $\overline{\rho_2} = \frac{\int_{l_1}^{L} \rho_2 dz}{\int_{l_1}^{L} dz} = \frac{\int_{l_1}^{L} \rho_2 dz}{l_2}$ and $\overline{\rho_2 h_2} = \frac{\int_{l_1}^{L} \rho_2 h_2 dz}{\int_{l_1}^{L} dz} = \frac{\int_{l_1}^{L} \rho_2 h_2 dz}{l_2}$

Therefore, mass balance and energy balance for control volume 2 are:

$$Al_2 \frac{d\overline{\rho_2}}{dt} - A\overline{\rho_2} \frac{dl_1}{dt} = \dot{m}_{int} - \dot{m}_o \tag{C-17}$$

$$Al_2 \frac{d\overline{\rho_2 h_2}}{dt} - A\overline{\rho_2 h_2} \frac{dl_1}{dt} - Al_2 \frac{dP}{dt} = \dot{m}_{int} h_{int} - \dot{m}_o h_o + \alpha_{i2} \pi D_i l_2 \left(T_{w2} - T_{r2}\right) \tag{C-18}$$

Energy equations for the tube walls can be derived similarly. It is assumed that the interface wall temperature, T_{w12}, is same as the control volume 1 wall temperature, T_{w1}. Then, the result equations are:

$$\left(C_p \rho A\right)_w \frac{dT_{w1}}{dt} = \alpha_{i1} \pi D_i \left(T_{r1} - T_{w1}\right) + \alpha_o \pi D_o \left(T_a - T_{w1}\right) \tag{C-19}$$

$$\left(C_p \rho A\right)_w \frac{dT_{w2}}{dt} + \frac{T_{w1} - T_{w2}}{l_2} \frac{dl_1}{dt} = \alpha_{i2} \pi D_i \left(T_{r2} - T_{w2}\right) + \alpha_o \pi D_o \left(T_a - T_{w1}\right) \tag{C-20}$$

To cancel out interface mass flow rate, $\dot{m}_{int}$, Equations (C-6) and (C-17) are combined. Equation (C-6) is substitute to Equation (C-10) and Equation (C-17) is substitute to Equation (C-18):

$$Al_1 \frac{d\overline{\rho_1}}{dt} + Al_2 \frac{d\overline{\rho_2}}{dt} + A\left(\overline{\rho_1} - \overline{\rho_2}\right) \frac{dl_1}{dt} = \dot{m}_i - \dot{m}_o \tag{C-21}$$

$$Al_1 \left(\frac{d\overline{\rho_1 h_1}}{dt} - h_{int} \frac{d\overline{\rho_1}}{dt} - \frac{dP}{dt} \right) + A\left(\overline{\rho_1 h_1} - \overline{\rho_1} h_{int}\right) \frac{dl_1}{dt} = \dot{m}_i \left(h_i - h_{int}\right) + \alpha_{i1} \pi D_i l_1 \left(T_{w1} - T_{r1}\right) \tag{C-22}$$

$$Al_2 \left(\frac{d\overline{\rho_2 h_2}}{dt} - h_{int} \frac{d\overline{\rho_2}}{dt} - \frac{dP}{dt} \right) - A\left(\overline{\rho_2 h_2} - \overline{\rho_2} h_{int}\right) \frac{dl_1}{dt} = \dot{m}_o \left(h_{int} - h_o\right) + \alpha_{i2} \pi D_i l_2 \left(T_{w2} - T_{r2}\right) \tag{C-23}$$

Density and enthalpy are functions of only pressure in two-phase region and are functions of pressure and temperature in vapor region for single-component evaporator.

$$\frac{d\overline{\rho_1}}{dt} = \frac{\partial\overline{\rho_1}}{\partial P}\frac{dP}{dt} \tag{C-24}$$

$$\frac{d\overline{\rho_1 h_1}}{dt} = \frac{\partial\overline{\rho_1 h_1}}{\partial P}\frac{dP}{dt} \tag{C-25}$$

$$\frac{d\overline{\rho_2}}{dt} = \frac{\partial\overline{\rho_2}}{\partial P}\frac{dP}{dt} + \frac{\partial\overline{\rho_2}}{\partial T_{r2}}\frac{dT_{r2}}{dt} \tag{C-26}$$

$$\frac{d\overline{\rho_2 h_2}}{dt} = \frac{\partial\overline{\rho_2 h_2}}{\partial P}\frac{dP}{dt} + \frac{\partial\overline{\rho_2 h_2}}{\partial T_{r2}}\frac{dT_{r2}}{dt} \tag{C-27}$$

Above equations are substitute to Equations (C-21) to (C-23):

$$A\left(\overline{\rho_1} - \overline{\rho_2}\right)\frac{dl_1}{dt} + A\left(l_1\frac{d\overline{\rho_1}}{dP} + l_2\frac{\partial\overline{\rho_2}}{\partial P}\right)\frac{dP}{dt} + Al_2\frac{\partial\overline{\rho_2}}{\partial T_{r2}}\frac{dT_{r2}}{dt} = \dot{m}_i - \dot{m}_o \tag{C-28}$$

$$A\left(\overline{\rho_1 h_1} - \overline{\rho_1}h_{int}\right)\frac{dl_1}{dt} + Al_1\left(\frac{\partial\overline{\rho_1 h_1}}{\partial P} - h_{int}\frac{\partial\overline{\rho_1}}{\partial P} - 1\right)\frac{dP}{dt} = \dot{m}_i\left(h_i - h_{int}\right) + \alpha_{i1}\pi D_i l_1\left(T_{w1} - T_{r1}\right) \tag{C-29}$$

$$\begin{aligned} -A\left(\overline{\rho_2 h_2} - \overline{\rho_2}h_{int}\right)\frac{dl_1}{dt} + Al_2\left(\frac{\partial\overline{\rho_2 h_2}}{\partial P}\frac{dP}{dt} - h_{int}\frac{\partial\overline{\rho_2}}{\partial P} - 1\right)\frac{dP}{dt} + Al_2\left(\frac{\partial\overline{\rho_2 h_2}}{\partial T_{r2}} - h_{int}\frac{\partial\overline{\rho_2}}{\partial T_{r2}}\right)\frac{dT_{r2}}{dt} \\ = \dot{m}_o\left(h_{int} - h_o\right) + \alpha_{i2}\pi D_i l_2\left(T_{w2} - T_{r2}\right) \end{aligned} \tag{C-30}$$

Total conservation equations for a single-component evaporator are expressed as matrix form as shown at Equation (6-13). Their coefficients are:

$$d_{11} = A\left(\overline{\rho_1} - \overline{\rho_2}\right)$$

$$d_{12} = A\left(l_1\frac{d\overline{\rho_1}}{dP} + l_2\frac{\partial\overline{\rho_2}}{\partial P}\right)$$

$$d_{13} = Al_2 \frac{\partial \overline{\rho_2}}{\partial T_{r2}}$$

$$d_{21} = A\left(\overline{\rho_1 h_1} - \overline{\rho_1} h_{int}\right)$$

$$d_{22} = Al_1\left(\frac{\partial \overline{\rho_1 h_1}}{\partial P} - h_{int}\frac{\partial \overline{\rho_1}}{\partial P} - 1\right)$$

$$d_{31} = -A\left(\overline{\rho_2 h_2} - \overline{\rho_2} h_{int}\right)$$

$$d_{32} = Al_2\left(\frac{\partial \overline{\rho_2 h_2}}{\partial P}\frac{dP}{dt} - h_{int}\frac{\partial \overline{\rho_2}}{\partial P} - 1\right)$$

$$d_{33} = Al_2\left(\frac{\partial \overline{\rho_2 h_2}}{\partial T_{r2}} - h_{int}\frac{\partial \overline{\rho_2}}{\partial T_{r2}}\right)$$

$$d_{44} = \left(C_p \rho A\right)_w$$

$$d_{51} = \frac{T_{w1} - T_{w2}}{l_2}$$

$$d_{55} = \left(C_p \rho A\right)_w$$

$$f_1 = \dot{m}_i - \dot{m}_o$$

$$f_2 = \dot{m}_i\left(h_i - h_{int}\right) + \alpha_{i1}\pi D_i l_1\left(T_{w1} - T_{r1}\right)$$

$$f_3 = \dot{m}_o\left(h_{int} - h_o\right) + \alpha_{i2}\pi D_i l_2\left(T_{w2} - T_{r2}\right)$$

$$f_4 = \alpha_{i1}\pi D_i\left(T_{r1} - T_{w1}\right) + \alpha_o \pi D_o\left(T_a - T_{w1}\right)$$

$$f_5 = \alpha_{i2}\pi D_i\left(T_{r2} - T_{w2}\right) + \alpha_o \pi D_o\left(T_a - T_{w1}\right)$$

Coefficients of Two-Component Condenser Model

The coefficients of matrix $\mathbf{C}$ and vector $\mathbf{F_c}$ in Equation (6-32) are shown below:

$$c_{12} = A_c l_{c1} \overline{\rho_{c1}}$$

$$c_{21} = A_c \left\{ \overline{\rho_{c1}} \left(x_{c12} - x_{co} \right) + \rho_{c2} \left(x_{co} - x_{c2} \right) \right\}$$

$$c_{22} = A_c l_{c1} \left(x_{c12} - x_{co} \right) \frac{\partial \overline{\rho_{c1}}}{\partial x_{c1}}$$

$$c_{23} = A_c l_{c2} \left\{ \rho_{c2} + \left(x_{c2} - x_{co} \right) \frac{\partial \rho_{c2}}{\partial x_{c2}} \right\}$$

$$c_{24} = A_c l_{c2} \left(x_{c2} - x_{co} \right) \frac{\partial \rho_{c2}}{\partial T_{cr2}}$$

$$c_{31} = A_c \left(\overline{\rho_{c1} h_{c1}} - \overline{\rho_{c1}} h_{c12} \right)$$

$$c_{32} = A_c l_{c1} \left(\frac{\partial \overline{\rho_{c1} h_{c1}}}{\partial x_{c1}} - h_{c12} \frac{\partial \overline{\rho_{c1}}}{\partial x_{c1}} \right)$$

$$c_{41} = A_c \left\{ \overline{\rho_{c1}} \left(h_{c12} - h_{co} \right) + \rho_{c2} \left(h_{co} - h_{c2} \right) \right\}$$

$$c_{42} = A_c l_{c1} \left(h_{c12} - h_{co} \right) \frac{\partial \overline{\rho_{c1}}}{\partial x_{c1}}$$

$$c_{43} = A_c l_{c2} \left\{ \left(h_{c2} - h_{co} \right) \frac{\partial \rho_{c2}}{\partial x_{c2}} + \rho_{c2} \frac{\partial h_{c2}}{\partial x_{c2}} \right\}$$

$$c_{44} = A_c l_{c2} \left\{ \left(h_{c2} - h_{co} \right) \frac{\partial \rho_{c2}}{\partial T_{cr2}} + \rho_{c2} \frac{\partial h_{c2}}{\partial T_{cr2}} \right\}$$

$$c_{55} = \left(\rho A C_p \right)_{ca} l_{c1}$$

$$c_{66} = \left(\rho A C_p \right)_{ca} l_{c2}$$

$$f_{c1} = \dot{m}_{ci} \left(x_{ci} - x_{c12} \right)$$

$$f_{c2} = \dot{m}_{ci} \left(x_{c12} - x_{co} \right) - \left\{ A_c l_{c1} \left(x_{c12} - x_{co} \right) \frac{\partial \overline{\rho_{c1}}}{\partial P_c} + A_c l_{c2} \left(x_{c2} - x_{co} \right) \frac{\partial \rho_{c2}}{\partial P_c} \right\} \frac{dP_c}{dt}$$

$$f_{c3} = \dot{m}_{ci}\left(h_{ci} - h_{c12}\right) + \left(UD\right)_{c1} l_{c1}\left(T_{ca1} - \overline{T_{cr1}}\right) - A_c l_{c1}\left(\frac{\partial \overline{\rho_{c1} h_{c1}}}{\partial P_c} - h_{c12}\frac{\partial \overline{\rho_{c1}}}{\partial P_c} - 1\right)\frac{dP_c}{dt}$$

$$f_{c4} = \dot{m}_{ci}\left(h_{c12} - h_{co}\right) + \left(UD\right)_{c2} l_{c2}\left(T_{ca2} - T_{cr2}\right)$$
$$-\left[A_c l_{c1}\left(h_{c12} - h_{co}\right)\frac{\partial \overline{\rho_{c1}}}{\partial P_c} + A_c l_{c2}\left\{\left(h_{c2} - h_{co}\right)\frac{\partial \rho_{c2}}{\partial P_c} + \rho_{c2}\frac{\partial h_{c2}}{\partial P_c} - 1\right\}\right]\frac{dP_c}{dt}$$

$$f_{c5} = \dot{m}_{ca} C_{pca}\left(T_{ca21} - T_{cao}\right) + \left(UD\right)_{c1} l_{c1}\left(\overline{T_{cr1}} - T_{ca1}\right)$$

$$f_{c6} = \dot{m}_{ca} C_{pca}\left(T_{cai} - T_{ca21}\right) + \left(UD\right)_{c2} l_{c2}\left(T_{cr2} - T_{ca2}\right)$$

Coefficients of Two-Component Absorber Model

The coefficients of matrix $\mathbf{A}$ and vector $\mathbf{F_a}$ in Equation (6-47) are shown below:

$$a_{12} = A_a l_{a1}\overline{\rho_{a1}}$$

$$a_{21} = A_a\left\{\overline{\rho_{a1}}\left(x_{a12} - x_{ao}\right) + \rho_{a2}\left(x_{ao} - x_{a2}\right)\right\}$$

$$a_{22} = A_a l_{a1}\left(x_{a12} - x_{ao}\right)\frac{\partial \overline{\rho_{a1}}}{\partial x_{a1}}$$

$$a_{23} = A_a l_{a2}\left\{\rho_{a2} + \left(x_{a2} - x_{ao}\right)\frac{\partial \rho_{a2}}{\partial x_{a2}}\right\}$$

$$a_{24} = A_a l_{a2}\left(x_{a2} - x_{ao}\right)\frac{\partial \rho_{a2}}{\partial T_{ar2}}$$

$$a_{31} = A_a\left(\overline{\rho_{a1} h_{a1}} - \overline{\rho_{a1}} h_{a12}\right)$$

$$a_{32} = A_a l_{a1}\left(\frac{\partial \overline{\rho_{a1} h_{a1}}}{\partial x_{a1}} - h_{a12}\frac{\partial \overline{\rho_{a1}}}{\partial x_{a1}}\right)$$

$$a_{41} = A_a\left\{\overline{\rho_{a1}}\left(h_{a12} - h_{ao}\right) + \rho_{a2}\left(h_{ao} - h_{a2}\right)\right\}$$

$$a_{42} = A_a l_{a1}\left(h_{a12} - h_{ao}\right)\frac{\partial \overline{\rho_{a1}}}{\partial x_{a1}}$$

$$a_{43} = A_a l_{a2} \left\{ \left(h_{a2} - h_{ao} \right) \frac{\partial \rho_{a2}}{\partial x_{a2}} + \rho_{a2} \frac{\partial h_{a2}}{\partial x_{a2}} \right\}$$

$$a_{44} = A_a l_{a2} \left\{ \left(h_{a2} - h_{ao} \right) \frac{\partial \rho_{a2}}{\partial T_{ar2}} + \rho_{a2} \frac{\partial h_{a2}}{\partial T_{ar2}} \right\}$$

$$a_{55} = \left(\rho A C_p \right)_{aa} l_{a1}$$

$$a_{66} = \left(\rho A C_p \right)_{aa} l_{a2}$$

$$f_{a1} = \dot{m}_{ai} \left(x_{ai} - x_{a12} \right)$$

$$f_{a2} = \dot{m}_{ai} \left(x_{a12} - x_{ao} \right) - \left\{ A_a l_{a1} \left(x_{a12} - x_{ao} \right) \frac{\partial \overline{\rho_{a1}}}{\partial P_a} + A_a l_{a2} \left(x_{a2} - x_{ao} \right) \frac{\partial \rho_{a2}}{\partial P_a} \right\} \frac{dP_a}{dt}$$

$$f_{a3} = \dot{m}_{ai} \left(h_{ai} - h_{a12} \right) + \left(UD \right)_{a1} l_{a1} \left(T_{aa1} - \overline{T_{ar1}} \right) - A_a l_{a1} \left(\frac{\partial \overline{\rho_{a1} h_{a1}}}{\partial P_a} - h_{a12} \frac{\partial \overline{\rho_{a1}}}{\partial P_a} - 1 \right) \frac{dP_a}{dt}$$

$$f_{a4} = \dot{m}_{ai} \left(h_{a12} - h_{ao} \right) + \left(UD \right)_{a2} l_{a2} \left(T_{aa2} - T_{ar2} \right) - \left[A_a l_{a1} \left(h_{a12} - h_{ao} \right) \frac{\partial \overline{\rho_{a1}}}{\partial P_a} + A_a l_{a2} \left\{ \left(h_{a2} - h_{ao} \right) \frac{\partial \rho_{a2}}{\partial P_a} + \rho_{a2} \frac{\partial h_{a2}}{\partial P_a} - 1 \right\} \right] \frac{dP_a}{dt}$$

$$f_{a5} = \dot{m}_{aa} C_{paa} \left(T_{aa21} - T_{aao} \right) + \left(UD \right)_{a1} l_{a1} \left(\overline{T_{ar1}} - T_{aa1} \right)$$

$$f_{a6} = \dot{m}_{aa} C_{paa} \left(T_{aai} - T_{aa21} \right) + \left(UD \right)_{a2} l_{a2} \left(T_{ar2} - T_{aa2} \right)$$

Coefficients of Two-Component Generator Model

The coefficients of matrix $\mathbf{G}$ and vector $\mathbf{F_g}$ in Equation (6-60) are shown below:

$$g_{13} = A_g l_{g1} \rho_{g1}$$

$$g_{21} = A_g \rho_{g1} \left(x_{g12} - x_{go} \right)$$

$$g_{22} = A_g l_{g1} \left(x_{g12} - x_{go} \right) \frac{\partial \rho_{g1}}{\partial T_{gr1}}$$

$$g_{23} = A_g l_{g1} \left(x_{g12} - x_{go} \right) \frac{\partial \rho_{g1}}{\partial x_{g1}}$$

$$g_{24} = A_g l_{g2} \overline{\rho_{g2}}$$

$$g_{31} = A_g \rho_{g1} \left(h_{g1} - h_{g12} \right)$$

$$g_{32} = A_g l_{g1} \left\{ \left(h_{g1} - h_{g12} \right) \frac{\partial \rho_{g1}}{\partial T_{gr1}} + \rho_{g1} \frac{\partial h_{g1}}{\partial T_{gr1}} \right\}$$

$$g_{33} = A_g l_{g1} \left\{ \left(h_{g1} - h_{g12} \right) \frac{\partial \rho_{g1}}{\partial x_{g1}} + \rho_{g1} \frac{\partial h_{g1}}{\partial x_{g1}} \right\}$$

$$g_{41} = A_g \left\{ \rho_{g1} \left(h_{g12} - h_{go} \right) - \left(\overline{\rho_{g2} h_{g2}} - \overline{\rho_{g2}} h_{go} \right) \right\}$$

$$g_{42} = A_g l_{g1} \left(h_{g12} - h_{go} \right) \frac{\partial \rho_{g1}}{\partial T_{gr1}}$$

$$g_{43} = A_g l_{g1} \left(h_{g12} - h_{go} \right) \frac{\partial \rho_{g1}}{\partial x_{g1}}$$

$$g_{44} = A_g l_{g2} \left(\frac{\partial \overline{\rho_{g2} h_{g2}}}{\partial x_{g2}} - h_{go} \frac{\partial \overline{\rho_{g2}}}{\partial x_{g2}} \right)$$

$$g_{55} = \left(\rho A C_p \right)_{ga} l_{g1}$$

$$g_{66} = \left(\rho A C_p \right)_{ga} l_{g2}$$

$$f_{g1} = \dot{m}_{gi} \left(x_{gi} - x_{g12} \right)$$

$$f_{g2} = \dot{m}_{gi} \left(x_{g12} - x_{go} \right) - A_g l_{g1} \left(x_{g12} - x_{go} \right) \frac{\partial \rho_{g1}}{\partial P_g} \frac{dP_g}{dt}$$

$$f_{g3} = \dot{m}_{gi} \left(h_{gi} - h_{g12} \right) + \left(UD \right)_{g1} l_{g1} \left(T_{ga1} - T_{gr1} \right) - A_g l_{g1} \left\{ \left(h_{g1} - h_{g12} \right) \frac{\partial \rho_{g1}}{\partial P_g} + \rho_{g1} \frac{\partial h_{g1}}{\partial P_g} - 1 \right\} \frac{dP_g}{dt}$$

$$f_{g4} = \dot{m}_{gi}\left(h_{g12} - h_{go}\right) + \left(UD\right)_{g2} l_{g2}\left(T_{ga2} - \overline{T_{gr2}}\right)$$
$$-\left\{A_g l_{g1}\left(h_{g12} - h_{go}\right)\frac{\partial \rho_{g1}}{\partial P_g} + A_g l_{g2}\left(\frac{\partial \overline{\rho_{g2} h_{g2}}}{\partial P_g} - h_{go}\frac{\partial \overline{\rho_{g2}}}{\partial P_g} - 1\right)\right\}\frac{dP_g}{dt}$$

$$f_{g5} = \dot{m}_{ga} C_{pga}\left(T_{ga21} - T_{gao}\right) + \left(UD\right)_{g1} l_{g1}\left(T_{gr1} - T_{ga1}\right)$$

$$f_{g6} = \dot{m}_{ga} C_{pga}\left(T_{gai} - T_{ga21}\right) + \left(UD\right)_{g2} l_{g2}\left(\overline{T_{gr2}} - T_{ga2}\right)$$

Derivations and Coefficients of Two-Component Evaporator Model

For control volume 1, mass balance that Leibniz's rule is applied from (6-1):

$$A_e\left\{\frac{d}{dt}\left(\int_0^{L_e^*} \rho_{e1} dz\right) - \rho_e^* \frac{dL_e^*}{dt}\right\} = \left(\rho A u\right)_{ei} - \left(\rho A u\right)_e^* \tag{C-31}$$

Left hand side is rewritten using mean density:

$$A_e\left\{\frac{d}{dt}\left(\overline{\rho_{e1}} L_e^*\right) - \rho_e^* \frac{dL_e^*}{dt}\right\} = \left(\rho A u\right)_{ei} - \left(\rho A u\right)_e^* \tag{C-32}$$

And we can obtain

$$A_e \overline{\rho_{e1}} \frac{dL_e^*}{dt} + A_e L_e^* \frac{d\overline{\rho_{e1}}}{dt} = \dot{m}_{ei} - \dot{m}_e^* \tag{C-33}$$

where $\dot{m}_{ei} = \left(\rho A u\right)_{ei}$ and $\dot{m}_e^* = \left(\rho A u\right)_e^* - \rho_e^* A_e \frac{dL_e^*}{dt}$

Just like control volume 1, mass balance for control volume 2 is

$$A_e \overline{\rho_{e2}} \frac{dL_e^*}{dt} + A_e\left(L_e^* - L_e\right)\frac{d\overline{\rho_{e2}}}{dt} = \dot{m}_{eo} - \dot{m}_e^* \tag{C-34}$$

We are interested in the fluid inside the evaporator, so, subtract Equation (C-34) from (C-33) and the mass balance for refrigerant:

$$A_e\left(\overline{\rho_{e1}}-\overline{\rho_{e2}}\right)\frac{dL_e^*}{dt}+A_eL_e^*\frac{d\overline{\rho_{e1}}}{dt}+A_e\left(L_e-L_e^*\right)\frac{d\overline{\rho_{e2}}}{dt}=\dot{m}_{ei}-\dot{m}_{eo} \tag{C-35}$$

The coefficients of matrix $\mathbf{E}$ and vector $\mathbf{F_e}$ in Equation (6-72) are shown below:

$$e_{12}=A_e\left\{L_e^*\overline{\rho_{e1}}+\left(L_e-L_e^*\right)\overline{\rho_{e2}}\right\}$$

$$e_{21}=A_e\left(\overline{\rho_{e1}h_{e1}}-\overline{\rho_{e1}}h_{eo}-\overline{\rho_{e2}h_{e2}}+\overline{\rho_{e2}}h_{eo}\right)$$

$$e_{22}=A_eL_e^*\left(\frac{\partial\overline{\rho_{e1}h_{e1}}}{\partial x_e}-h_{eo}\frac{\partial\overline{\rho_{e1}}}{\partial x_e}\right)+A_e\left(L_e-L_e^*\right)\left(\frac{\partial\overline{\rho_{e2}h_{e2}}}{\partial x_e}-h_{eo}\frac{\partial\overline{\rho_{e2}}}{\partial x_e}\right)$$

$$e_{33}=\left(\rho AC_p\right)_{ea}L_e$$

$$f_{e1}=\dot{m}_{ei}\left(x_{ei}-x_{eo}\right)$$

$$f_{e2}=\dot{m}_{ei}\left(h_{ei}-h_{eo}\right)+\left(UA\right)_e\left(T_{ea}-\overline{T_{er}}\right)$$
$$-A_e\left\{L_e^*\left(\frac{\partial\overline{\rho_{e1}h_{e1}}}{\partial P_e}-h_{eo}\frac{\partial\overline{\rho_{e1}}}{\partial P_e}-1\right)+\left(L_e-L_e^*\right)\left(\frac{\partial\overline{\rho_{e2}h_{e2}}}{\partial P_e}-h_{eo}\frac{\partial\overline{\rho_{e2}}}{\partial P_e}-1\right)\right\}\frac{dP_e}{dt}$$

$$f_{e3}=\dot{m}_{ea}C_{pea}\left(T_{eai}-T_{eao}\right)+\left(UA\right)_e\left(\overline{T_{er}}-T_{ea}\right)$$

Coefficients of Two-Component SHX Model

The coefficients of matrix $\mathbf{S}$ and vector $\mathbf{F_s}$ in Equation (6-82) are shown below:

$$s_{11}=A_{sc}L_s\left(x_{sc}-x_{sco}\right)\frac{\partial\rho_{sc}}{\partial T_{sc}}$$

$$s_{12}=A_{sc}L_s\left\{\left(x_{sc}-x_{sco}\right)\frac{\partial\rho_{sc}}{\partial x_{sc}}+\rho_{sc}\right\}$$

$$s_{21} = A_{sc} L_s \left\{ \left(h_{sc} - h_{sco} \right) \frac{\partial \rho_{sc}}{\partial T_{sc}} + \rho_{sc} \frac{\partial h_{sc}}{\partial T_{sc}} \right\}$$

$$s_{22} = A_{sc} L_s \left\{ \left(h_{sc} - h_{sco} \right) \frac{\partial \rho_{sc}}{\partial x_{sc}} + \rho_{sc} \frac{\partial h_{sc}}{\partial x_{sc}} \right\}$$

$$s_{33} = A_{sh} L_s \left(x_{sh} - x_{shi} \right) \frac{\partial \rho_{sh}}{\partial T_{sh}}$$

$$s_{34} = A_{sh} L_s \left\{ \left(x_{sh} - x_{shi} \right) \frac{\partial \rho_{sh}}{\partial x_{sh}} + \rho_{sh} \right\}$$

$$s_{43} = A_{sh} L_s \left\{ \left(h_{sh} - h_{shi} \right) \frac{\partial \rho_{sh}}{\partial T_{sh}} + \rho_{sh} \frac{\partial h_{sh}}{\partial T_{sh}} \right\}$$

$$s_{44} = A_{sh} L_s \left\{ \left(h_{sh} - h_{shi} \right) \frac{\partial \rho_{sh}}{\partial x_{sh}} + \rho_{sh} \frac{\partial h_{sh}}{\partial x_{sh}} \right\}$$

$$f_{s1} = \dot{m}_{sci} \left(x_{sci} - x_{sco} \right) - A_{sc} L_s \left(x_{sc} - x_{sco} \right) \frac{\partial \rho_{sc}}{\partial P_s} \frac{dP_s}{dt}$$

$$f_{s2} = \dot{m}_{sci} \left(h_{sci} - h_{sco} \right) + \left(UA \right)_s \left(T_{sh} - T_{sc} \right) - A_{sc} L_s \left\{ \left(h_{sc} - h_{sco} \right) \frac{\partial \rho_{sc}}{\partial P_s} + \rho_{sc} \frac{\partial h_{sc}}{\partial P_s} - 1 \right\} \frac{dP_s}{dt}$$

$$f_{s3} = \dot{m}_{sho} \left(x_{shi} - x_{sho} \right) - A_{sh} L_s \left(x_{sh} - x_{shi} \right) \frac{\partial \rho_{sh}}{\partial P_s} \frac{dP_s}{dt}$$

$$f_{s4} = \dot{m}_{sho} \left(h_{shi} - h_{sho} \right) + \left(UA \right)_s \left(T_{sc} - T_{sh} \right) - A_{sh} L_s \left\{ \left(h_{sh} - h_{shi} \right) \frac{\partial \rho_{sh}}{\partial P_s} + \rho_{sh} \frac{\partial h_{sh}}{\partial P_s} - 1 \right\} \frac{dP_s}{dt}$$

Computer Code for Moving Boundary Model

For dynamic modeling of two-component two-phase flow heat exchangers, Simulink and Matlab are used. Level 2 S-function in Simulink is employed for each heat exchanger to calculate continues dynamic problem. Bogart tables and Matlab codes introduced in Appendix A are applied to heat exchanger models. Each heat exchanger

model has a simulation file, **_Sim.mdl and **_Dynamics.m for its level 2 m-file. To run these files, the driver file, **_Driver.m is needed. This file has inputs, initial values, and geometric parameters for **_Sim.mdl and **_Dynamics.m. To obtain two-phase thermodynamic properties of ammonia-water mixture, **_2ph.m and its data file, **_pxz.mat files are necessary. For example, condenser simulation model, COND_Sim.mdl file, is shown in Figure C-1 and other models have similar Simulink files.

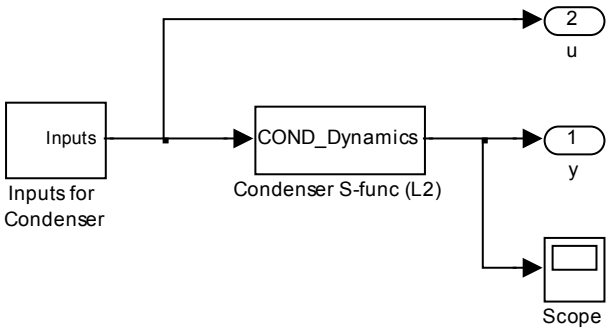

Figure C-1. Condenser Simulink model

Before running COND_Driver.m, open COND_Sim.mdl and Configuration Parameters. We can select a solver, simulation time, relative tolerance, etc in the window. After tuning input parameters, step inputs in COND_Driver.m, push F5 key. This allows runing COND_Sim.mdl. The results are shown at workspace. Source codes are shown below:

Condenser

- COND_Driver.m

```
% COND_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is a driver file to run a dynamic simulation of a condenser.
% Written by ChoonJae Ryu, Energy and Gas Dynamics Lab (Prof. William E.
% Lear).
% Department of Mechanical and Aerospace Engineering, University of Florida
```

```
% This format is modified from the original written by Prof. Oscar D.
% Crisalle (Chemical Egineering, % University of Florida).

clear all

%% Initial values %%
% Initial inputs %
Pc          = 1.791850466e+3;
xci         = 0.998;
[xcil Tci]  = y2x_Bogart(Pc,xci);
hci         = vapprop_ibrahim(Tci,Pc,xci);
mdci        = 0.06182711;
Tcai        = 2.887111e+2;
mdca        = 0.644232220068264;

% Geometry and Parameters%
Lc          = 1.3335;
Ac          = 0.004141911097410;
UDc1        = 3.097209791054786;
UDc2        = 1.909558741495225;

% Initial values of state variables%
lc1         = 1.313597926931102;
xc1         = 0.998;
xc2         = xc1;
Tcr2        = 3.167085762441687e+2;
Tca1        = 3.020517268311589e+2;
Tca2        = 2.889072768311589e+2;

%% Step Inputs %%
stime       = 0;           % step time or start time

% Pressure
Pcf         = Pc;
Pcslope     = 0;              % step input for dPcdt
Pcst        = stime;          % sec. - start time

% Ammonia mass fraction and enthalpy
xcif        = xci;            % final value
xcist       = stime;          % step time
[xcil Tci]  = y2x_Bogart(Pcf,xcif);
hcif        = vapprop_ibrahim(Tci,Pcf,xcif);

% Mass flow rate - refrigerant
mdcif       = mdci;           % final value
mdcist      = stime;          % step time

% Mass flow rate - secondary fluid
mdcaf       = mdca;           % final value
mdcast      = stime;          % step time

% Inlet temperature - secondary fluid
Tcaif       = Tcai;           % final value
Tcaist      = stime;          % step time
```

```
%% Run Simulink Model %%
sim('COND_Sim');
% End of COND_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- COND_Dynamics.m

```
% COND_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function COND_Dynamics(block)
% Dynamic modeling of a condenser
% Primary fluid is two-phase and two-component (two-phase in, sub-cooled
% liquid out) and secondary fluid is incompressible sub-cooled liquid.
%
% This S-Function is utilized in the SIMULINK file named COND_Sim.mdl
% and initialized by COND_Driver.m
% Written by ChoonJae Ryu, Energy and Gas Dynamics Lab (Prof. William E.
% Lear).
% Department of Mechanical and Aerospace Engineering, University of Florida

%%
%% The setup method is used to setup the basic attributes of the
%% S-function such as ports, parameters, etc. Do not add any other
%% calls to the main body of the function.
%%
setup(block);

%endfunction

%% Function: setup ===================================================
%% Abstract:
%%   Set up the S-function block's basic characteristics such as:
%%   - Input ports
%%   - Output ports
%%   - Dialog parameters
%%   - Options
%%
%%   Required         : Yes
%%   C-Mex counterpart: mdlInitializeSizes
%%
function setup(block)

% Register parameters
block.NumDialogPrms = 15;

% Register number of ports
block.NumInputPorts  = 1;
block.NumOutputPorts = 1;

% Setup port properties to be inherited or dynamic
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% Override input port properties
block.InputPort(1).Dimensions        = 7;       % size_inputs
%block.InputPort(1).DatatypeID  = 0;  % double
```

```
%block.InputPort(1).Complexity  = 'Real';
block.InputPort(1).DirectFeedthrough = 0;   % size_feedthrough

% Override output port properties
block.OutputPort(1).Dimensions       = 9;       % size_outputs
%block.OutputPort(1).DatatypeID  = 0; % double
%block.OutputPort(1).Complexity  = 'Real';

% Register sample times
%  [0 offset]            : Continuous sample time
%  [positive_num offset] : Discrete sample time
%
%  [-1, 0]               : Inherited sample time
%  [-2, 0]               : Variable sample time
block.SampleTimes = [0 0];

% States
block.NumContStates = 6;

% Specify the block simStateComliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume
DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'CustomSimState',  < Has GetSimState and SetSimState methods
%    'DisallowSimState' < Error out when saving or restoring the model sim
state
block.SimStateCompliance = 'DefaultSimState';

%% -----------------------------------------------------------------
%% The M-file S-function uses an internal registry for all
%% block methods. You should register all relevant methods
%% (optional and required) as illustrated below. You may choose
%% any suitable name for the methods and implement these methods
%% as local functions within the same file. See comments
%% provided for each function for more information.
%% -----------------------------------------------------------------

%block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
block.RegBlockMethod('InitializeConditions', @InitializeConditions);
%block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('Outputs', @Outputs);     % Required
%block.RegBlockMethod('Update', @Update);
block.RegBlockMethod('Derivatives', @Derivatives);
%block.RegBlockMethod('Terminate', @Terminate); % Required

%end setup

%%
%% PostPropagationSetup:
%%   Functionality    : Setup work areas and state variables. Can
%%                      also register run-time methods here
%%   Required         : No
%%   C-Mex counterpart: mdlSetWorkWidths
%%
```

```
%function DoPostPropSetup(block)
%block.NumDworks = 1;
%
%  block.Dwork(1).Name            = 'x1';
%  block.Dwork(1).Dimensions      = 1;
%  block.Dwork(1).DatatypeID      = 0;      % double
%  block.Dwork(1).Complexity      = 'Real'; % real
%  block.Dwork(1).UsedAsDiscState = true;
%
%  block.Dwork(2).Name            = 'x2';
%  block.Dwork(2).Dimensions      = 1;
%  block.Dwork(2).DatatypeID      = 0;      % double
%  block.Dwork(2).Complexity      = 'Real'; % real
%  block.Dwork(2).UsedAsDiscState = true;
%


%%
%% InitializeConditions:
%%   Functionality    : Called at the start of simulation and if it is
%%                      present in an enabled subsystem configured to reset
%%                      states, it will be called when the enabled subsystem
%%                      restarts execution to reset the states.
%%   Required         : No
%%   C-MEX counterpart: mdlInitializeConditions
%%
function InitializeConditions(block)

lc1_0   = block.DialogPrm(1).Data;
xc1_0   = block.DialogPrm(2).Data;
xc2_0   = block.DialogPrm(3).Data;
Tcr2_0  = block.DialogPrm(4).Data;
Tca1_0  = block.DialogPrm(5).Data;
Tca2_0  = block.DialogPrm(6).Data;

block.ContStates.Data = [lc1_0; xc1_0; xc2_0; Tcr2_0; Tca1_0; Tca2_0];

%end InitializeConditions

%%
%% Start:
%%   Functionality    : Called once at start of model execution. If you
%%                      have states that should be initialized once, this
%%                      is the place to do it.
%%   Required         : No
%%   C-MEX counterpart: mdlStart
%%
%function Start(block)

%endfunction

%%
%% Outputs:
%%   Functionality    : Called to generate block outputs in
%%                      simulation step
%%   Required         : Yes
```

```
%%   C-MEX counterpart: mdlOutputs
%%
function Outputs(block)
% parameters and constants
Lc      = block.DialogPrm(7).Data;          % m
Ac      = block.DialogPrm(8).Data;          % m2
UDc1    = block.DialogPrm(9).Data;          % kW/mK
UDc2    = block.DialogPrm(10).Data;         % kW/mK

% inputs [Pc dPcdt xci mdci mdca Tcai]
Pc      = block.InputPort(1).Data(1);
if Pc == 0 % only for initial output (t=0)
    Pc      = block.DialogPrm(11).Data;
    dPcdt   = 0;
    xci     = block.DialogPrm(12).Data;
    hci     = block.DialogPrm(13).Data;
    mdci    = block.DialogPrm(14).Data;
    Tcai    = block.DialogPrm(15).Data;
else
    dPcdt   = block.InputPort(1).Data(2);
    xci     = block.InputPort(1).Data(3);
    hci     = block.InputPort(1).Data(4);
    mdci    = block.InputPort(1).Data(5);
    Tcai    = block.InputPort(1).Data(7);
end

% state variables, xc = []'
lc1     = block.ContStates.Data(1);
xc1     = block.ContStates.Data(2);     if xc1>1 xc1=1; end
xc2     = block.ContStates.Data(3);     if xc2>1 xc2=1; end
Tcr2    = block.ContStates.Data(4);
Tca1    = block.ContStates.Data(5);
Tca2    = block.ContStates.Data(6);
lc2     = Lc - lc1;

% Thermodyanmic properties
[Tcr1 rhoc1 rhoc1hc1 drhoc1dPc drhoc1hc1dPc drhoc1dxc1 drhoc1hc1dxc1 hc12] ...
    = COND_2ph(Pc,xc1);

xc12    = xc1;
hc1     = (hci + hc12)/2;

[rhoc2 hc2 drhoc2dPc dhc2dPc drhoc2dxc2 dhc2dxc2 ...
    drhoc2dTcr2 dhc2dTcr2] = liqp(Tcr2,Pc,xc2);

xco     = xc2;
hco     = 2*hc2 - hc12;

Tca21   = 2*Tca2 - Tcai;
Tcao    = 2*Tca1 - Tca21;

c12     = Ac*lc1*rhoc1;

c21     = Ac*(rhoc1*(xc12 - xco) + rhoc2*(xco - xc2));
c22     = Ac*lc1*(xc12 - xco)*drhoc1dxc1;
```

```
c23     = Ac*lc2*((xc2 - xco)*drhoc2dxc2 + rhoc2);
c24     = Ac*lc2*(xc2 - xco)*drhoc2dTcr2;

c31     = Ac*(rhoc1hc1 - rhoc1*hc12);
c32     = Ac*lc1*(drhoc1hc1dxc1 - hc12*drhoc1dxc1);

c41     = Ac*(rhoc1*(hc12 - hco) + rhoc2*(hco - hc2));
c42     = Ac*lc1*(hc12 - hco)*drhoc1dxc1;
c43     = Ac*lc2*((hc2 - hco)*drhoc2dxc2 + rhoc2*dhc2dxc2);
c44     = Ac*lc2*((hc2 - hco)*drhoc2dTcr2 + rhoc2*dhc2dTcr2);

fc1     = mdci*(xci - xc12);
fc2     = mdci*(xc12 - xco) - (Ac*lc1*(xc12 - xco)*drhoc1dPc ...
        + Ac*lc2*(xc2 - xco)*drhoc2dPc)*dPcdt;
fc3     = mdci*(hci - hc12) + UDc1*lc1*(Tca1 - Tcr1) ...
        - Ac*lc1*(drhoc1hc1dPc - hc12*drhoc1dPc - 1)*dPcdt;
fc4     = mdci*(hc12 - hco) + UDc2*lc2*(Tca2 - Tcr2) ...
        - (Ac*lc1*(hc12 - hco)*drhoc1dPc ...
        + Ac*lc2*((hc2 - hco)*drhoc2dPc + rhoc2*dhc2dPc - 1))*dPcdt;

xdc1    = fc1/c12;
ldc1    = fc3/c31 - c32/c31*xdc1;
xdc2    = (c44*(fc2 - c21*ldc1 - c22*xdc1) - c24*(fc4 - c41*ldc1 ...
        - c42*xdc1))/(c23*c44 - c24*c43);
Tdcr2   = (-c43*(fc2 - c21*ldc1 - c22*xdc1) + c23*(fc4 - c41*ldc1 ...
        - c42*xdc1))/(c23*c44 - c24*c43);

mdco    = mdci - Ac*(rhoc1 - rhoc2)*ldc1 ...
        - Ac*lc1*(drhoc1dPc*dPcdt + drhoc1dxc1*xdc1) ...
        - Ac*lc2*(drhoc2dPc*dPcdt + drhoc2dxc2*xdc2 + drhoc2dTcr2*Tdcr2);

block.OutputPort(1).Data = [mdco; xco; hco; Tcao; lc1; xc1; xc2; hc12; hc2];
%block.OutputPort(1).Data = block.Dwork(1).Data + block.InputPort(1).Data;

%end Outputs

%%
%% Update:
%%   Functionality    : Called to update discrete states
%%                      during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlUpdate
%%
%function Update(block)

%block.Dwork(1).Data = block.InputPort(1).Data;

%end Update

%%
%% Derivatives:
%%   Functionality    : Called to update derivatives of
%%                      continuous states during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlDerivatives
```

```
%%
function Derivatives(block)
% parameters or constants
Lc      = block.DialogPrm(7).Data;          % m
Ac      = block.DialogPrm(8).Data;          % m2
UDc1    = block.DialogPrm(9).Data;          % kW/mK
UDc2    = block.DialogPrm(10).Data;         % kW/mK
rhoca   = 998;                              % kg/m3
Cpca    = 4.18;                             % kJ/kgK
Aca     = Ac;                               % m2

% inputs [Pc dPcdt xci mdci mdca Tcai]
Pc      = block.InputPort(1).Data(1);
dPcdt   = block.InputPort(1).Data(2);
xci     = block.InputPort(1).Data(3);
hci     = block.InputPort(1).Data(4);
mdci    = block.InputPort(1).Data(5);
mdca    = block.InputPort(1).Data(6);
Tcai    = block.InputPort(1).Data(7);

% state variables, xc = []'
lc1     = block.ContStates.Data(1);
xc1     = block.ContStates.Data(2);     if xc1>1 xc1=1; end
xc2     = block.ContStates.Data(3);     if xc2>1 xc2=1; end
Tcr2    = block.ContStates.Data(4);
Tca1    = block.ContStates.Data(5);
Tca2    = block.ContStates.Data(6);
lc2     = Lc - lc1;

% Thermodyanmic properties
[Tcr1 rhoc1 rhoc1hc1 drhoc1dPc drhoc1hc1dPc drhoc1dxc1 drhoc1hc1dxc1 hc12] ...
    = COND_2ph(Pc,xc1);

xc12    = xc1;

[rhoc2 hc2 drhoc2dPc dhc2dPc drhoc2dxc2 dhc2dxc2 ...
    drhoc2dTcr2 dhc2dTcr2] = liqp(Tcr2,Pc,xc2);

xco     = xc2;
hco     = 2*hc2 - hc12;

Tca21   = 2*Tca2 - Tcai;
Tcao    = 2*Tca1 - Tca21;

c12     = Ac*lc1*rhoc1;

c21     = Ac*(rhoc1*(xc12 - xco) + rhoc2*(xco - xc2));
c22     = Ac*lc1*(xc12 - xco)*drhoc1dxc1;
c23     = Ac*lc2*((xc2 - xco)*drhoc2dxc2 + rhoc2);
c24     = Ac*lc2*(xc2 - xco)*drhoc2dTcr2;

c31     = Ac*(rhoc1hc1 - rhoc1*hc12);
c32     = Ac*lc1*(drhoc1hc1dxc1 - hc12*drhoc1dxc1);

c41     = Ac*(rhoc1*(hc12 - hco) + rhoc2*(hco - hc2));
```

```
c42     = Ac*lc1*(hc12 - hco)*drhoc1dxc1;
c43     = Ac*lc2*((hc2 - hco)*drhoc2dxc2 + rhoc2*dhc2dxc2);
c44     = Ac*lc2*((hc2 - hco)*drhoc2dTcr2 + rhoc2*dhc2dTcr2);

c55     = rhoca*Aca*Cpca*lc1;

c66     = rhoca*Aca*Cpca*lc2;

fc1     = mdci*(xci - xc12);
fc2     = mdci*(xc12 - xco) - (Ac*lc1*(xc12 - xco)*drhoc1dPc ...
        + Ac*lc2*(xc2 - xco)*drhoc2dPc)*dPcdt;
fc3     = mdci*(hci - hc12) + UDc1*lc1*(Tca1 - Tcr1) ...
        - Ac*lc1*(drhoc1hc1dPc - hc12*drhoc1dPc - 1)*dPcdt;
fc4     = mdci*(hc12 - hco) + UDc2*lc2*(Tca2 - Tcr2) ...
        - (Ac*lc1*(hc12 - hco)*drhoc1dPc ...
        + Ac*lc2*((hc2 - hco)*drhoc2dPc + rhoc2*dhc2dPc - 1))*dPcdt;
fc5     = mdca*Cpca*(Tca21 - Tcao) + UDc1*lc1*(Tcr1 - Tca1);
fc6     = mdca*Cpca*(Tcai - Tca21) + UDc2*lc2*(Tcr2 - Tca2);

xdc1    = fc1/c12;
ldc1    = fc3/c31 - c32/c31*xdc1;
xdc2    = (c44*(fc2 - c21*ldc1 - c22*xdc1) - c24*(fc4 - c41*ldc1 ...
        - c42*xdc1))/(c23*c44 - c24*c43);
Tdcr2   = (-c43*(fc2 - c21*ldc1 - c22*xdc1) + c23*(fc4 - c41*ldc1 ...
        - c42*xdc1))/(c23*c44 - c24*c43);
Tdca1   = fc5/c55;
Tdca2   = fc6/c66;

block.Derivatives.Data = [ldc1; xdc1; xdc2; Tdcr2; Tdca1; Tdca2];

%end Derivatives

%%
%% Terminate:
%%   Functionality    : Called at the end of simulation for cleanup
%%   Required         : Yes
%%   C-MEX counterpart: mdlTerminate
%%
%function Terminate(block)

%end Terminate
% End of COND_Dynamic.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- COND_2ph.m

```
% COND_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Tcr rhoc rhochc drhocdPc drhochcdPc drhocdxc drhochcdxc hcb] ...
    = COND_2ph(P,xc1)
% Thermodyanmic properties of the two-phase fluid in a condenser.
%
% This function is utilized in COND_Dynamics.m and COND_pxz.mat file is
% necessary.
% Written by ChoonJae Ryu, Energy and Gas Dynamics Lab (Prof. William E.
```

```
% Lear).
% Department of Mechanical and Aerospace Engineering, University of Florida

load COND_pxz.mat

dP      = 1e-3;
xtol    = 1e-6;

P1      = P - dP/2;
P2      = P + dP/2;

if xc1 > 1-xtol
    x1  = xc1 - xtol/2;
    x2  = 1;
elseif xc1 < xtol
    x1  = 0;
    x2  = xc1 + xtol/2;
else
    x1  = xc1 - xtol/2;
    x2  = xc1 + xtol/2;
end
dx      = x2 - x1;

pp      = [P1 P P2];
xx      = [x1 xc1 x2];

rcp     = interp2(xc,Pc',rhoc1,xx,pp');
rhcp    = interp2(xc,Pc',rhoc1hc1,xx,pp');
Tcr     = interp2(xc,Pc',Tcr1,xx(2),pp(2));
hcb     = interp2(xc,Pc',hc12,xx(2),pp(2));
rhoc    = rcp(2,2);
rhochc  = rhcp(2,2);

drhocdPc    = (rcp(3,2) - rcp(1,2))/dP;
drhochcdPc  = (rhcp(3,2) - rhcp(1,2))/dP;
drhocdxc    = (rcp(2,3) - rcp(2,1))/dx;
drhochcdxc  = (rhcp(2,3) - rhcp(2,1))/dx;
% End of COND_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Absorber

- ABS_Driver.m

```
% ABS_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

%% initial values %%
% Initial inputs %
Pa          = 8.339618761000000e+002;
xai         = 0.62;
hai         = 3.306850577071115e+002;
mdai        = 0.216574986421376;
Taai        = 2.887111000000000e+002;
```

```
mdaa        = 1.658345174363795;

% Geometry and parameters %
La          = 1.3208;
Aa          = 0.007862719188542;
UDa1        = 3.089813894119425;
UDa2        = 4.269195056210765;

% Initial values of state variables %
la1         = 1.147922798833574;
xa1         = 0.620000000000000;
xa2         = 0.620000000000000;
Tar2        = 3.061617943044990e+002;
Taa1        = 2.963320428308150e+002;
Taa2        = 2.895931428308149e+002;

%% Step Inputs %%
stime       = 10;

% Pressure
Paf         = Pa;
Paslope     = 0;              % step input for dPadt
Past        = stime;          % sec. - start time

% Ammonia mass fraction and enthalpy
xaif        = xai;            % final value
xaist       = stime;          % step time

% Enthalpy
haif        = hai;
haist       = stime;

% Mass flow rate - refrigerant
mdaif       = mdai;           % final value
mdaist      = stime;          % step time

% Mass flow rate - secondary fluid
mdaaf       = mdaa;           % final value
mdaast      = stime;          % step time

% Inlet temperature - secondary fluid
Taaif       = Taai;           % final value
Taaist      = stime;          % step time

%% Run Simulink Model %%
sim('ABS_Sim');
% End of ABS_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- ABS_Dynamics.m

```
% ABS_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ABS_Dynamics(block)
% Dynamic modeling of an absorber
```

```
%
% Written by ChoonJae Ryu, Energy and Gas Dynamics Lab (Prof. Lear).
% Department of Mechanical and Aerospace Engineering, University of Florida

%%
%% The setup method is used to setup the basic attributes of the
%% S-function such as ports, parameters, etc. Do not add any other
%% calls to the main body of the function.
%%
setup(block);

%endfunction

%% Function: setup ===================================================
%% Abstract:
%%   Set up the S-function block's basic characteristics such as:
%%   - Input ports
%%   - Output ports
%%   - Dialog parameters
%%   - Options
%%
%%   Required         : Yes
%%   C-Mex counterpart: mdlInitializeSizes
%%
function setup(block)

% Register parameters
block.NumDialogPrms = 15;

% Register number of ports
block.NumInputPorts  = 1;
block.NumOutputPorts = 1;

% Setup port properties to be inherited or dynamic
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% Override input port properties
block.InputPort(1).Dimensions        = 7;       % size_inputs
%block.InputPort(1).DatatypeID  = 0;  % double
%block.InputPort(1).Complexity  = 'Real';
block.InputPort(1).DirectFeedthrough = 0;   % size_feedthrough

% Override output port properties
block.OutputPort(1).Dimensions       = 5;       % size_outputs
%block.OutputPort(1).DatatypeID  = 0; % double
%block.OutputPort(1).Complexity  = 'Real';

% Register sample times
%  [0 offset]            : Continuous sample time
%  [positive_num offset] : Discrete sample time
%
%  [-1, 0]               : Inherited sample time
%  [-2, 0]               : Variable sample time
block.SampleTimes = [0 0];
```

```
% States
block.NumContStates = 6; % xe = [le1 Pe heo Tew1 Tew2]'

% Specify the block simStateComliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume
DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'CustomSimState',  < Has GetSimState and SetSimState methods
%    'DisallowSimState' < Error out when saving or restoring the model sim
state
block.SimStateCompliance = 'DefaultSimState';

%% -----------------------------------------------------------------
%% The M-file S-function uses an internal registry for all
%% block methods. You should register all relevant methods
%% (optional and required) as illustrated below. You may choose
%% any suitable name for the methods and implement these methods
%% as local functions within the same file. See comments
%% provided for each function for more information.
%% -----------------------------------------------------------------

%block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
block.RegBlockMethod('InitializeConditions', @InitializeConditions);
%block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('Outputs', @Outputs);     % Required
%block.RegBlockMethod('Update', @Update);
block.RegBlockMethod('Derivatives', @Derivatives);
%block.RegBlockMethod('Terminate', @Terminate); % Required

%end setup

%%
%% PostPropagationSetup:
%%   Functionality    : Setup work areas and state variables. Can
%%                      also register run-time methods here
%%   Required         : No
%%   C-Mex counterpart: mdlSetWorkWidths
%%
%function DoPostPropSetup(block)
%block.NumDworks = 1;
%
%  block.Dwork(1).Name            = 'x1';
%  block.Dwork(1).Dimensions      = 1;
%  block.Dwork(1).DatatypeID      = 0;      % double
%  block.Dwork(1).Complexity      = 'Real'; % real
%  block.Dwork(1).UsedAsDiscState = true;
%
%  block.Dwork(2).Name            = 'x2';
%  block.Dwork(2).Dimensions      = 1;
%  block.Dwork(2).DatatypeID      = 0;      % double
%  block.Dwork(2).Complexity      = 'Real'; % real
%  block.Dwork(2).UsedAsDiscState = true;
%
```

```
%%
%% InitializeConditions:
%%   Functionality    : Called at the start of simulation and if it is
%%                      present in an enabled subsystem configured to reset
%%                      states, it will be called when the enabled subsystem
%%                      restarts execution to reset the states.
%%   Required         : No
%%   C-MEX counterpart: mdlInitializeConditions
%%
function InitializeConditions(block)

la1_0   = block.DialogPrm(1).Data;
xa1_0   = block.DialogPrm(2).Data;
xa2_0   = block.DialogPrm(3).Data;
Tar2_0  = block.DialogPrm(4).Data;
Taa1_0  = block.DialogPrm(5).Data;
Taa2_0  = block.DialogPrm(6).Data;

block.ContStates.Data = [la1_0; xa1_0; xa2_0; Tar2_0; Taa1_0; Taa2_0];

%end InitializeConditions

%%
%% Start:
%%   Functionality    : Called once at start of model execution. If you
%%                      have states that should be initialized once, this
%%                      is the place to do it.
%%   Required         : No
%%   C-MEX counterpart: mdlStart
%%
%function Start(block)

%endfunction

%%
%% Outputs:
%%   Functionality    : Called to generate block outputs in
%%                      simulation step
%%   Required         : Yes
%%   C-MEX counterpart: mdlOutputs
%%
function Outputs(block)
% parameters and constants
La      = block.DialogPrm(7).Data;          % m
Aa      = block.DialogPrm(8).Data;          % m2
UDa1    = block.DialogPrm(9).Data;          % kW/mK
UDa2    = block.DialogPrm(10).Data;         % kW/mK

% inputs
Pa      = block.InputPort(1).Data(1);
if Pa == 0 % only for initial output (t = 0)
    Pa      = block.DialogPrm(11).Data;
    dPadt   = 0;
    xai     = block.DialogPrm(12).Data;
```

```
    hai     = block.DialogPrm(13).Data;
    mdai    = block.DialogPrm(14).Data;
    Taai    = block.DialogPrm(15).Data;
else
    dPadt   = block.InputPort(1).Data(2);
    xai     = block.InputPort(1).Data(3);
    hai     = block.InputPort(1).Data(4);
    mdai    = block.InputPort(1).Data(5);
    Taai    = block.InputPort(1).Data(7);
end

% state variables, xc = []'
la1     = block.ContStates.Data(1);
xa1     = block.ContStates.Data(2);     if xa1>1 xa1=1; end
xa2     = block.ContStates.Data(3);     if xa2>1 xa2=1; end
Tar2    = block.ContStates.Data(4);
Taa1    = block.ContStates.Data(5);
Taa2    = block.ContStates.Data(6);
la2     = La - la1;

% Thermodyanmic properties
[Tar1 rhoa1 rhoa1ha1 drhoa1dPa drhoa1ha1dPa drhoa1dxa1 drhoa1ha1dxa1 ha12] ...
    = ABS_2ph(Pa,xa1,hai);

xa12    = xa1;
%Ta12    = Tb_Bogart(Pa,xa12);
%ha12    = liqprop_ibrahim(Ta12,Pa,xa12);

[rhoa2 ha2 drhoa2dPa dha2dPa drhoa2dxa2 dha2dxa2 ...
    drhoa2dTar2 dha2dTar2] = liqp(Tar2,Pa,xa2);

xao     = xa2;
hao     = 2*ha2 - ha12;
%Tao     = h2t_liq(Pa,hao,xao);

Taa21   = 2*Taa2 - Taai;
Taao    = 2*Taa1 - Taa21;

a12     = Aa*la1*rhoa1;

a21     = Aa*(rhoa1*(xa12 - xao) + rhoa2*(xao - xa2));
a22     = Aa*la1*(xa12 - xao)*drhoa1dxa1;
a23     = Aa*la2*((xa2 - xao)*drhoa2dxa2 + rhoa2);
a24     = Aa*la2*(xa2 - xao)*drhoa2dTar2;

a31     = Aa*(rhoa1ha1 - rhoa1*ha12);
a32     = Aa*la1*(drhoa1ha1dxa1 - ha12*drhoa1dxa1);

a41     = Aa*(rhoa1*(ha12 - hao) + rhoa2*(hao - ha2));
a42     = Aa*la1*(ha12 - hao)*drhoa1dxa1;
a43     = Aa*la2*((ha2 - hao)*drhoa2dxa2 + rhoa2*dha2dxa2);
a44     = Aa*la2*((ha2 - hao)*drhoa2dTar2 + rhoa2*dha2dTar2);

fa1     = mdai*(xai - xa12);
fa2     = mdai*(xa12 - xao) - (Aa*la1*(xa12 - xao)*drhoa1dPa ...
```

```
        + Aa*la2*(xa2 - xao)*drhoa2dPa)*dPadt;
fa3     = mdai*(hai - ha12) + UDa1*la1*(Taa1 - Tar1) ...
        - Aa*la1*(drhoa1ha1dPa - ha12*drhoa1dPa - 1)*dPadt;
fa4     = mdai*(ha12 - hao) + UDa2*la2*(Taa2 - Tar2) ...
        - (Aa*la1*(ha12 - hao)*drhoa1dPa ...
        + Aa*la2*((ha2 - hao)*drhoa2dPa + rhoa2*dha2dPa - 1))*dPadt;

xda1    = fa1/a12;
lda1    = fa3/a31 - a32/a31*xda1;
xda2    = (a44*(fa2 - a21*lda1 - a22*xda1) - a24*(fa4 - a41*lda1 ...
        - a42*xda1))/(a23*a44 - a24*a43);
Tdar2   = (-a43*(fa2 - a21*lda1 - a22*xda1) + a23*(fa4 - a41*lda1 ...
        - a42*xda1))/(a23*a44 - a24*a43);

mdao    = mdai - Aa*(rhoa1 - rhoa2)*lda1 ...
        - Aa*la1*(drhoa1dPa*dPadt + drhoa1dxa1*xda1) ...
        - Aa*la2*(drhoa2dPa*dPadt + drhoa2dxa2*xda2 + drhoa2dTar2*Tdar2);

block.OutputPort(1).Data = [mdao; xao; hao; Taao; la1];
%block.OutputPort(1).Data = block.Dwork(1).Data + block.InputPort(1).Data;

%end Outputs

%%
%% Update:
%%   Functionality    : Called to update discrete states
%%                      during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlUpdate
%%
%function Update(block)

%block.Dwork(1).Data = block.InputPort(1).Data;

%end Update

%%
%% Derivatives:
%%   Functionality    : Called to update derivatives of
%%                      continuous states during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlDerivatives
%%
function Derivatives(block)
% parameters or constants
La      = block.DialogPrm(7).Data;          % m
Aa      = block.DialogPrm(8).Data;          % m2
UDa1    = block.DialogPrm(9).Data;          % kW/mK
UDa2    = block.DialogPrm(10).Data;         % kW/mK
rhoaa   = 998;                              % kg/m3
Cpaa    = 4.18;                             % kJ/kgK
Aaa     = Aa;                               % m2

% inputs [Pc dPcdt xci mdci mdca Tcai]
Pa      = block.InputPort(1).Data(1);
```

```
dPadt   = block.InputPort(1).Data(2);
xai     = block.InputPort(1).Data(3);
hai     = block.InputPort(1).Data(4);
mdai    = block.InputPort(1).Data(5);
mdaa    = block.InputPort(1).Data(6);
Taai    = block.InputPort(1).Data(7);

% state variables, xc = []'
la1     = block.ContStates.Data(1);
xa1     = block.ContStates.Data(2);     if xa1>1 xa1=1; end
xa2     = block.ContStates.Data(3);     if xa2>1 xa2=1; end
Tar2    = block.ContStates.Data(4);
Taa1    = block.ContStates.Data(5);
Taa2    = block.ContStates.Data(6);
la2     = La - la1;

% Thermodyanmic properties
[Tar1 rhoa1 rhoa1ha1 drhoa1dPa drhoa1ha1dPa drhoa1dxa1 drhoa1ha1dxa1 ha12] ...
    = ABS_2ph(Pa,xa1,hai);

xa12    = xa1;
%Ta12    = Tb_Bogart(Pa,xa12);
%ha12    = liqprop_ibrahim(Ta12,Pa,xa12);

[rhoa2 ha2 drhoa2dPa dha2dPa drhoa2dxa2 dha2dxa2 ...
    drhoa2dTar2 dha2dTar2] = liqp(Tar2,Pa,xa2);

xao     = xa2;
hao     = 2*ha2 - ha12;

Taa21   = 2*Taa2 - Taai;
Taao    = 2*Taa1 - Taa21;

a12     = Aa*la1*rhoa1;

a21     = Aa*(rhoa1*(xa12 - xao) + rhoa2*(xao - xa2));
a22     = Aa*la1*(xa12 - xao)*drhoa1dxa1;
a23     = Aa*la2*((xa2 - xao)*drhoa2dxa2 + rhoa2);
a24     = Aa*la2*(xa2 - xao)*drhoa2dTar2;

a31     = Aa*(rhoa1ha1 - rhoa1*ha12);
a32     = Aa*la1*(drhoa1ha1dxa1 - ha12*drhoa1dxa1);

a41     = Aa*(rhoa1*(ha12 - hao) + rhoa2*(hao - ha2));
a42     = Aa*la1*(ha12 - hao)*drhoa1dxa1;
a43     = Aa*la2*((ha2 - hao)*drhoa2dxa2 + rhoa2*dha2dxa2);
a44     = Aa*la2*((ha2 - hao)*drhoa2dTar2 + rhoa2*dha2dTar2);

a55     = rhoaa*Aaa*Cpaa*la1;

a66     = rhoaa*Aaa*Cpaa*la2;

fa1     = mdai*(xai - xa12);
fa2     = mdai*(xa12 - xao) - (Aa*la1*(xa12 - xao)*drhoa1dPa ...
```

```
        + Aa*la2*(xa2 - xao)*drhoa2dPa)*dPadt;
fa3     = mdai*(hai - ha12) + UDa1*la1*(Taa1 - Tar1) ...
        - Aa*la1*(drhoa1ha1dPa - ha12*drhoa1dPa - 1)*dPadt;
fa4     = mdai*(ha12 - hao) + UDa2*la2*(Taa2 - Tar2) ...
        - (Aa*la1*(ha12 - hao)*drhoa1dPa ...
        + Aa*la2*((ha2 - hao)*drhoa2dPa + rhoa2*dha2dPa - 1))*dPadt;
fa5     = mdaa*Cpaa*(Taa21 - Taao) + UDa1*la1*(Tar1 - Taa1);
fa6     = mdaa*Cpaa*(Taai - Taa21) + UDa2*la2*(Tar2 - Taa2);

xda1    = fa1/a12;
lda1    = fa3/a31 - a32/a31*xda1;
xda2    = (a44*(fa2 - a21*lda1 - a22*xda1) - a24*(fa4 - a41*lda1 ...
        - a42*xda1))/(a23*a44 - a24*a43);
Tdar2   = (-a43*(fa2 - a21*lda1 - a22*xda1) + a23*(fa4 - a41*lda1 ...
        - a42*xda1))/(a23*a44 - a24*a43);
Tdaa1   = fa5/a55;
Tdaa2   = fa6/a66;

block.Derivatives.Data = [lda1; xda1; xda2; Tdar2; Tdaa1; Tdaa2];

%end Derivatives

%%
%% Terminate:
%%   Functionality    : Called at the end of simulation for cleanup
%%   Required         : Yes
%%   C-MEX counterpart: mdlTerminate
%%
%function Terminate(block)

%end Terminate
% End of ABS_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- ABS_2ph.m

```
% ABS_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Ta rhoa rhoaha drhoadPa drhoahadPa drhoadxa drhoahadxa hab] ...
    = ABS_2ph(P,xa1,hai)

load ABS_pxz1.mat

m = 100;
dP = 1e-3;
xtol = 1e-6;

P1 = P - dP/2;
P2 = P + dP/2;

if xa1 > 1-xtol
    x1 = xa1 - xtol/2;
    x2 = 1;
elseif xa1 < xtol
    x1 = 0;
```

```
    x2 = xa1 + xtol/2;
else
    x1 = xa1 - xtol/2;
    x2 = xa1 + xtol/2;
end
dx = x2 - x1;

za = 0:1/n:1;
[xxa ppa zza] = meshgrid(xa,Pa,za);
%z = 0:0.01:1;
z = [0 1];
[xi pii zi] = meshgrid(xa1,P,z);
%Taz = griddata3(xxa,ppa,zza,Tar1,xi,pi,zi);
ha1z = interp3(xxa,ppa,zza,ha1,xi,pii,zi);
%haz(1,:) = ha1z(1,1,:);
%zai = interp1(haz,z,hai);
%haz0 = vapprop_ibrahim(Taz(1),P,xa1);
%haz1 = liqprop_ibrahim(Taz(2),P,xa1);
zai = (hai - ha1z(1))/(ha1z(2) - ha1z(1));

xx = [x1 xa1 x2];
pp = [P1 P P2];
zz = zai:(1-zai)/m:1;
[xxi ppi zzi] = meshgrid(xx,pp,zz);

T1 = interp3(xxa,ppa,zza,Tar1,xxi,ppi,zzi);
Ta = mean(T1(2,2,:));

rho1 = interp3(xxa,ppa,zza,rhoa1,xxi,ppi,zzi);
rhoa = mean(rho1(2,2,:));
rp1 = mean(rho1(1,2,:));
rp2 = mean(rho1(3,2,:));
rx1 = mean(rho1(2,1,:));
rx2 = mean(rho1(2,3,:));

rho1h1 = interp3(xxa,ppa,zza,rhoa1ha1,xxi,ppi,zzi);
rhoaha = mean(rho1h1(2,2,:));
rhp1 = mean(rho1h1(1,2,:));
rhp2 = mean(rho1h1(3,2,:));
rhx1 = mean(rho1h1(2,1,:));
rhx2 = mean(rho1h1(2,3,:));

drhoadPa = (rp2 - rp1)/dP;
drhoahadPa = (rhp2 - rhp1)/dP;
drhoadxa = (rx2 - rx1)/dx;
drhoahadxa = (rhx2 - rhx1)/dx;

ha12(:,:) = ha1(:,:,101);
hab = interp2(xa,Pa',ha12,xa1,P);
% End of ABS_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Generator

- GEN_Driver.m

```
% GEN_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

%% initial values %%
% Initial inputs %
Pg          = 1.791850466000000e+003;
xgi         = 0.62;
hgi         = 90.175825309765756;
mdgi        = 0.185995325421376;
Tgai        = 5.728500000000000e+002;
mdga        = 1.152879667;

% Geometry and parameters %
Lg          = 0.6096;
Ag          = 0.041071438769110;
Cpga        = 1.251729794677541;
UDg1        = 0.965702895542424;
UDg2        = 0.853902219139207;

% Initial values of state variables %
lg1         = 0.009865125203645;
xg1         = 0.62;
xg2         = 0.62;
Tgr1        = 3.418070673356066e+002;
Tga1        = 5.062929379110602e+002;
Tga2        = 5.398429379110602e+002;

%% Step Inputs %%
stime       = 10;

% Pressure
Pgf         = Pg;
Pgslope     = 0;                % step input for dPadt
Pgst        = stime;            % sec. - start time

% Ammonia mass fraction and enthalpy
xgif        = xgi;              % final value
xgist       = stime;            % step time

% Enthalpy
hgif        = hgi;
hgist       = stime;

% Mass flow rate - refrigerant
mdgif       = mdgi;             % final value
mdgist      = stime;            % step time

% Mass flow rate - secondary fluid
mdgaf       = mdga;             % final value
mdgast      = stime;            % step time
```

```
% Inlet temperature - secondary fluid
Tgaif       = Tgai;           % final value
Tgaist      = stime;          % step time

%% Run Simulink Model %%
sim('GEN_Sim');
% End of GEN_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- GEN_Dynamics.m

```
% GEN_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function GEN_Dynamics(block)
% Dynamic modeling of a generator
%
% Written by ChoonJae Ryu, Energy and Gas Dynamics Lab (Prof. Lear).
% Department of Mechanical and Aerospace Engineering, University of Florida

%%
%% The setup method is used to setup the basic attributes of the
%% S-function such as ports, parameters, etc. Do not add any other
%% calls to the main body of the function.
%%
setup(block);

%endfunction

%% Function: setup ===================================================
%% Abstract:
%%   Set up the S-function block's basic characteristics such as:
%%   - Input ports
%%   - Output ports
%%   - Dialog parameters
%%   - Options
%%
%%   Required         : Yes
%%   C-Mex counterpart: mdlInitializeSizes
%%
function setup(block)

% Register parameters
block.NumDialogPrms = 17;

% Register number of ports
block.NumInputPorts  = 1;
block.NumOutputPorts = 1;

% Setup port properties to be inherited or dynamic
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% Override input port properties
block.InputPort(1).Dimensions        = 7;       % size_inputs
```

```
%block.InputPort(1).DatatypeID  = 0;  % double
%block.InputPort(1).Complexity  = 'Real';
block.InputPort(1).DirectFeedthrough = 0;   % size_feedthrough

% Override output port properties
block.OutputPort(1).Dimensions       = 9;       % size_outputs
%block.OutputPort(1).DatatypeID  = 0; % double
%block.OutputPort(1).Complexity  = 'Real';

% Register sample times
%  [0 offset]            : Continuous sample time
%  [positive_num offset] : Discrete sample time
%
%  [-1, 0]               : Inherited sample time
%  [-2, 0]               : Variable sample time
block.SampleTimes = [0 0];

% States
block.NumContStates = 6; % xe = [le1 Pe heo Tew1 Tew2]'

% Specify the block simStateComliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume
DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'CustomSimState',  < Has GetSimState and SetSimState methods
%    'DisallowSimState' < Error out when saving or restoring the model sim
state
block.SimStateCompliance = 'DefaultSimState';

%% -----------------------------------------------------------------
%% The M-file S-function uses an internal registry for all
%% block methods. You should register all relevant methods
%% (optional and required) as illustrated below. You may choose
%% any suitable name for the methods and implement these methods
%% as local functions within the same file. See comments
%% provided for each function for more information.
%% -----------------------------------------------------------------

%block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
block.RegBlockMethod('InitializeConditions', @InitializeConditions);
%block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('Outputs', @Outputs);     % Required
%block.RegBlockMethod('Update', @Update);
block.RegBlockMethod('Derivatives', @Derivatives);
%block.RegBlockMethod('Terminate', @Terminate); % Required

%end setup

%%
%% PostPropagationSetup:
%%   Functionality    : Setup work areas and state variables. Can
%%                      also register run-time methods here
%%   Required         : No
%%   C-Mex counterpart: mdlSetWorkWidths
```

```
%%
%function DoPostPropSetup(block)
%block.NumDworks = 1;
%
%  block.Dwork(1).Name            = 'x1';
%  block.Dwork(1).Dimensions      = 1;
%  block.Dwork(1).DatatypeID      = 0;      % double
%  block.Dwork(1).Complexity      = 'Real'; % real
%  block.Dwork(1).UsedAsDiscState = true;
%
%  block.Dwork(2).Name            = 'x2';
%  block.Dwork(2).Dimensions      = 1;
%  block.Dwork(2).DatatypeID      = 0;      % double
%  block.Dwork(2).Complexity      = 'Real'; % real
%  block.Dwork(2).UsedAsDiscState = true;
%


%%
%% InitializeConditions:
%%   Functionality    : Called at the start of simulation and if it is
%%                      present in an enabled subsystem configured to reset
%%                      states, it will be called when the enabled subsystem
%%                      restarts execution to reset the states.
%%   Required         : No
%%   C-MEX counterpart: mdlInitializeConditions
%%
function InitializeConditions(block)
lg1_0   = block.DialogPrm(1).Data;
Tgr1_0  = block.DialogPrm(2).Data;
xg1_0   = block.DialogPrm(3).Data;
xg2_0   = block.DialogPrm(4).Data;
Tga1_0  = block.DialogPrm(5).Data;
Tga2_0  = block.DialogPrm(6).Data;

block.ContStates.Data = [lg1_0; Tgr1_0; xg1_0; xg2_0; Tga1_0; Tga2_0];

%end InitializeConditions

%%
%% Start:
%%   Functionality    : Called once at start of model execution. If you
%%                      have states that should be initialized once, this
%%                      is the place to do it.
%%   Required         : No
%%   C-MEX counterpart: mdlStart
%%
%function Start(block)

%endfunction

%%
%% Outputs:
%%   Functionality    : Called to generate block outputs in
%%                      simulation step
%%   Required         : Yes
%%   C-MEX counterpart: mdlOutputs
```

```
%%
function Outputs(block)
% parameters and constants
Lg      = block.DialogPrm(7).Data;          % m
Ag      = block.DialogPrm(8).Data;          % m2
UDg1    = block.DialogPrm(9).Data;          % kW/mK
UDg2    = block.DialogPrm(10).Data;         % kW/mK
Cpga    = block.DialogPrm(11).Data;

% inputs
Pg      = block.InputPort(1).Data(1);
if Pg == 0
    Pg      = block.DialogPrm(12).Data;
    dPgdt   = 0;
    xgi     = block.DialogPrm(13).Data;
    hgi     = block.DialogPrm(14).Data;
    mdgi    = block.DialogPrm(15).Data;
    mdga    = block.DialogPrm(16).Data;
    Tgai    = block.DialogPrm(17).Data;
else
    dPgdt   = block.InputPort(1).Data(2);
    xgi     = block.InputPort(1).Data(3);
    hgi     = block.InputPort(1).Data(4);
    mdgi    = block.InputPort(1).Data(5);
    mdga    = block.InputPort(1).Data(6);
    Tgai    = block.InputPort(1).Data(7);
end

% state variables, xc = []'
lg1     = block.ContStates.Data(1);
Tgr1    = block.ContStates.Data(2);
xg1     = block.ContStates.Data(3);     if xg1>1 xg1=1; end
xg2     = block.ContStates.Data(4);     if xg2>1 xg2=1; end
Tga1    = block.ContStates.Data(5);
Tga2    = block.ContStates.Data(6);
lg2     = Lg - lg1;

% Thermodyanmic properties
[rhog1 hg1 drhog1dPg dhg1dPg drhog1dxg1 dhg1dxg1 drhog1dTgr1 dhg1dTgr1] ...
    = liqp(Tgr1,Pg,xg1);

Tgr2    = Tga2 + 2*(Tga2 - Tgai)*mdga*Cpga/UDg2/lg2;
[rhog2 hgo rhog2hg2 drhog2dPg drhog2hg2dPg drhog2dxg2 drhog2hg2dxg2 ...
    hgov hgol xgov xgol] = GEN_2ph(Pg,xg2,Tgr2);

xg12    = xg1;
Tg12    = Tb_Bogart(Pg,xg12);
hg12    = liqprop_ibrahim(Tg12,Pg,xg12);

xgo     = xg2;

Tga21   = 2*Tga2 - Tgai;
Tgao    = 2*Tga1 - Tga21;

g13     = Ag*lg1*rhog1;
```

```
g21     = Ag*rhog1*(xg12 - xgo);
g22     = Ag*lg1*(xg12 - xgo)*drhog1dTgr1;
g23     = Ag*lg1*(xg12 - xgo)*drhog1dxg1;
g24     = Ag*lg2*rhog2;

g31     = Ag*rhog1*(hg1 - hg12);
g32     = Ag*lg1*((hg1 - hg12)*drhog1dTgr1 + rhog1*dhg1dTgr1);
g33     = Ag*lg1*((hg1 - hg12)*drhog1dxg1 + rhog1*dhg1dxg1);

g41     = Ag*(rhog1*(hg12 - hgo) - (rhog2hg2 - rhog2*hgo));
g42     = Ag*lg1*(hg12 - hgo)*drhog1dTgr1;
g43     = Ag*lg1*(hg12 - hgo)*drhog1dxg1;
g44     = Ag*lg2*(drhog2hg2dxg2 - hgo*drhog2dxg2);

fg1     = mdgi*(xgi - xg12);
fg2     = mdgi*(xg12 - xgo) - Ag*lg1*(xg12 - xgo)*drhog1dPg*dPgdt;
fg3     = mdgi*(hgi - hg12) + UDg1*lg1*(Tga1 - Tgr1) ...
        - Ag*lg1*((hg1 - hg12)*drhog1dPg + rhog1*dhg1dPg - 1)*dPgdt;
    %if fg3 < 1e-10 fg3=0; end
fg4     = mdgi*(hg12 - hgo) + UDg2*lg2*(Tga2 - Tgr2) ...
        - (Ag*lg1*(hg12 - hgo)*drhog1dPg ...
        + Ag*lg2*(drhog2hg2dPg - hgo*drhog2dPg - 1))*dPgdt;
    %if fg4 < 1e-10 fg4=0; end

G       = [0 0 g13 0; g21 g22 g23 g24; g31 g32 g33 0; g41 g42 g43 g44];
Fg      = [fg1; fg2; fg3; fg4];
xdg     = G\Fg;
ldg1    = xdg(1);
Tdgr1   = xdg(2);
xdg1    = xdg(3);
xdg2    = xdg(4);

mdgo    = mdgi - Ag*(rhog1 - rhog2)*ldg1 ...
        - Ag*lg1*(drhog1dPg*dPgdt + drhog1dxg1*xdg1 + drhog1dTgr1*Tdgr1) ...
        - Ag*lg2*(drhog2dPg*dPgdt + drhog2dxg2*xdg2);

block.OutputPort(1).Data = [mdgo; xgo; hgo; Tgao; lg1; hgov; hgol; xgov;
xgol];
%block.OutputPort(1).Data = block.Dwork(1).Data + block.InputPort(1).Data;

%end Outputs

%%
%% Update:
%%   Functionality    : Called to update discrete states
%%                      during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlUpdate
%%
%function Update(block)

%block.Dwork(1).Data = block.InputPort(1).Data;

%end Update
```

```
%%
%% Derivatives:
%%   Functionality    : Called to update derivatives of
%%                      continuous states during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlDerivatives
%%
function Derivatives(block)
% parameters or constants
Lg      = block.DialogPrm(7).Data;          % m
Ag      = block.DialogPrm(8).Data;          % m2
UDg1    = block.DialogPrm(9).Data;          % kW/mK
UDg2    = block.DialogPrm(10).Data;         % kW/mK
rhoga   = 2;                                % kg/m3
Cpga    = block.DialogPrm(11).Data;         % kJ/kgK
Aga     = Ag;                               % m2

% inputs [Pc dPcdt xci mdci mdca Tcai]
Pg      = block.InputPort(1).Data(1);
dPgdt   = block.InputPort(1).Data(2);
xgi     = block.InputPort(1).Data(3);
hgi     = block.InputPort(1).Data(4);
mdgi    = block.InputPort(1).Data(5);
mdga    = block.InputPort(1).Data(6);
Tgai    = block.InputPort(1).Data(7);

% state variables, xc = []'
lg1     = block.ContStates.Data(1);
Tgr1    = block.ContStates.Data(2);
xg1     = block.ContStates.Data(3);     if xg1>1 xg1=1; end
xg2     = block.ContStates.Data(4);     if xg2>1 xg2=1; end
Tga1    = block.ContStates.Data(5);
Tga2    = block.ContStates.Data(6);
lg2     = Lg - lg1;

% Thermodyanmic properties
[rhog1 hg1 drhog1dPg dhg1dPg drhog1dxg1 dhg1dxg1 drhog1dTgr1 dhg1dTgr1] ...
    = liqp(Tgr1,Pg,xg1);

Tgr2    = Tga2 + 2*(Tga2 - Tgai)*mdga*Cpga/UDg2/lg2;
[rhog2 hgo rhog2hg2 drhog2dPg drhog2hg2dPg drhog2dxg2 drhog2hg2dxg2 ...
    hgov hgol xgov xgol] = GEN_2ph(Pg,xg2,Tgr2);
xg12    = xg1;
Tg12    = Tb_Bogart(Pg,xg12);
hg12    = liqprop_ibrahim(Tg12,Pg,xg12);

xgo     = xg2;

Tga21   = 2*Tga2 - Tgai;
Tgao    = 2*Tga1 - Tga21;

g13     = Ag*lg1*rhog1;

g21     = Ag*rhog1*(xg12 - xgo);
```

```
g22     = Ag*lg1*(xg12 - xgo)*drhog1dTgr1;
g23     = Ag*lg1*(xg12 - xgo)*drhog1dxg1;
g24     = Ag*lg2*rhog2;

g31     = Ag*rhog1*(hg1 - hg12);
g32     = Ag*lg1*((hg1 - hg12)*drhog1dTgr1 + rhog1*dhg1dTgr1);
g33     = Ag*lg1*((hg1 - hg12)*drhog1dxg1 + rhog1*dhg1dxg1);

g41     = Ag*(rhog1*(hg12 - hgo) - (rhog2hg2 - rhog2*hgo));
g42     = Ag*lg1*(hg12 - hgo)*drhog1dTgr1;
g43     = Ag*lg1*(hg12 - hgo)*drhog1dxg1;
g44     = Ag*lg2*(drhog2hg2dxg2 - hgo*drhog2dxg2);

g55     = rhoga*Aga*Cpga*lg1;

g66     = rhoga*Aga*Cpga*lg2;

fg1     = mdgi*(xgi - xg12);
fg2     = mdgi*(xg12 - xgo) - Ag*lg1*(xg12 - xgo)*drhog1dPg*dPgdt;
fg3     = mdgi*(hgi - hg12) + UDg1*lg1*(Tga1 - Tgr1) ...
        - Ag*lg1*((hg1 - hg12)*drhog1dPg + rhog1*dhg1dPg - 1)*dPgdt;
    %if fg3 < 1e-10 fg3=0; end
fg4     = mdgi*(hg12 - hgo) + UDg2*lg2*(Tga2 - Tgr2) ...
        - (Ag*lg1*(hg12 - hgo)*drhog1dPg ...
        + Ag*lg2*(drhog2hg2dPg - hgo*drhog2dPg - 1))*dPgdt;
    %if fg4 < 1e-10 fg4=0; end
fg5     = mdga*Cpga*(Tga21 - Tgao) + UDg1*lg1*(Tgr1 - Tga1);
fg6     = mdga*Cpga*(Tgai - Tga21) + UDg2*lg2*(Tgr2 - Tga2);

G       = [0 0 g13 0; g21 g22 g23 g24; g31 g32 g33 0; g41 g42 g43 g44];
Fg      = [fg1; fg2; fg3; fg4];
xdg     = G\Fg;
Tdga1   = fg5/g55;
Tdga2   = fg6/g66;

block.Derivatives.Data = [xdg; Tdga1; Tdga2];

%end Derivatives

%%
%% Terminate:
%%   Functionality    : Called at the end of simulation for cleanup
%%   Required         : Yes
%%   C-MEX counterpart: mdlTerminate
%%
%function Terminate(block)

%end Terminate
% End of GEN_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- GEN_2ph.m

```
% GEN_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [rhog ho rhoghg drhogdPg drhoghgdPg drhogdxg drhoghgdxg ...
    hog hol xog xol] = GEN_2ph(P,xg2,T2)

load GEN_pxz2.mat

m = 100;
dP = 1e-3;
xtol = 1e-6;

P1 = P - dP/2;
P2 = P + dP/2;

if xg2 > 1-xtol
    x1 = xg2 - xtol/2;
    x2 = 1;
elseif xg2 < xtol
    x1 = 0;
    x2 = xg2 + xtol/2;
else
    x1 = xg2 - xtol/2;
    x2 = xg2 + xtol/2;
end
dx = x2 - x1;

zg = 0:1/n:1;
[xxg ppg zzg] = meshgrid(xg,Pg,zg);
z = 0:0.01:1;
[xi pii zi] = meshgrid(xg2,P,z);
Tgmeanz = interp3(xxg,ppg,zzg,Tmean,xi,pii,zi);
Tgmz(1,:) = Tgmeanz(1,1,:);
zgo = interp1(Tgmz,z,T2);
[xhi phi zhi] = meshgrid(xg2,P,zgo);
ho = interp3(xxg,ppg,zzg,hg2,xhi,phi,zhi);
hog = interp3(xxg,ppg,zzg,hgg,xhi,phi,zhi);
hol = interp3(xxg,ppg,zzg,hgl,xhi,phi,zhi);
xog = interp3(xxg,ppg,zzg,xgg,xhi,phi,zhi);
xol = interp3(xxg,ppg,zzg,xgl,xhi,phi,zhi);

xx = [x1 xg2 x2];
pp = [P1 P P2];
zz = 0:zgo/m:zgo;
[xxi ppi zzi] = meshgrid(xx,pp,zz);

rho2 = interp3(xxg,ppg,zzg,rhog2,xxi,ppi,zzi);
rhog = mean(rho2(2,2,:));
rp1 = mean(rho2(1,2,:));
rp2 = mean(rho2(3,2,:));
rx1 = mean(rho2(2,1,:));
rx2 = mean(rho2(2,3,:));

rho2h2 = interp3(xxg,ppg,zzg,rhog2hg2,xxi,ppi,zzi);
rhoghg = mean(rho2h2(2,2,:));
rhp1 = mean(rho2h2(1,2,:));
rhp2 = mean(rho2h2(3,2,:));
rhx1 = mean(rho2h2(2,1,:));
```

```
rhx2 = mean(rho2h2(2,3,:));

drhogdPg = (rp2 - rp1)/dP;
drhoghgdPg = (rhp2 - rhp1)/dP;
drhogdxg = (rx2 - rx1)/dx;
drhoghgdxg = (rhx2 - rhx1)/dx;
% End of GEN_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Evaporator

- EVAP_Driver.m

```
% EVAP_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

%% initial values %%
% initial inputs %
Pe          = 8.339618761000000e+002;
xei         = 0.998000000000000;
hei         = 1.252714784168311e+002;
mdei        = 0.061827110000000;
Teai        = 3.288222000000000e+002;
mdea        = 1.152879667000000;

% Geometry and parameters %
Le          = 0.914400000000000;
Ae          = 0.059148967930983;
Cpea        = 2.382488973429181;
UAe         = 2.783884462820257;
Les         = 1.004577863913866;

% Initial values of state variables %
xe          = 0.998000000000000;
Tea         = 3.166055500000000e+002;

%% Step inputs %%
stime       = 10;

% Pressure
Pef         = Pe;%/1.01;
Peslope     = 0;%0.5;          % step input for dPadt
Pest        = stime;        % sec. - start time

% Ammonia mass fraction and enthalpy
xeif        = xei*1.001;    % final value
xeist       = stime;        % step time

% Enthalpy
heif        = hei;%*1.1;
heist       = stime;

% Mass flow rate - refrigerant
mdeif       = mdei;%*1.1;    % final value
```

```
mdeist      = stime;        % step time

% Mass flow rate - secondary fluid
mdeaf       = mdea;%*1.1;    % final value
mdeast      = stime;        % step time

% Inlet temperature - secondary fluid
Teaif       = Teai;     % final value
Teaist      = stime;        % step time

%% Run Simulink Model %%
sim('EVAP_Sim');
% EVAP_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- EVAP_Dynamics.m

```
% EVAP_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function EVAP_Dynamics(block)
% Dynamic modeling of an evaporator
%
% Written by ChoonJae Ryu, Energy and Gas Dynamics Lab (Prof. Lear).
% Department of Mechanical and Aerospace Engineering, University of Florida

%%
%% The setup method is used to setup the basic attributes of the
%% S-function such as ports, parameters, etc. Do not add any other
%% calls to the main body of the function.
%%
setup(block);

%endfunction

%% Function: setup ===================================================
%% Abstract:
%%   Set up the S-function block's basic characteristics such as:
%%   - Input ports
%%   - Output ports
%%   - Dialog parameters
%%   - Options
%%
%%   Required         : Yes
%%   C-Mex counterpart: mdlInitializeSizes
%%
function setup(block)

% Register parameters
block.NumDialogPrms = 12;

% Register number of ports
block.NumInputPorts  = 1;
block.NumOutputPorts = 1;

% Setup port properties to be inherited or dynamic
```

```
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% Override input port properties
block.InputPort(1).Dimensions        = 7;       % size_inputs
%block.InputPort(1).DatatypeID  = 0;  % double
%block.InputPort(1).Complexity  = 'Real';
block.InputPort(1).DirectFeedthrough = 0;   % size_feedthrough

% Override output port properties
block.OutputPort(1).Dimensions       = 5;       % size_outputs
%block.OutputPort(1).DatatypeID  = 0; % double
%block.OutputPort(1).Complexity  = 'Real';

% Register sample times
%  [0 offset]            : Continuous sample time
%  [positive_num offset] : Discrete sample time
%
%  [-1, 0]               : Inherited sample time
%  [-2, 0]               : Variable sample time
block.SampleTimes = [0 0];

% States
block.NumContStates = 3; % xe = [le1 Pe heo Tew1 Tew2]'

% Specify the block simStateComliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume
DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'CustomSimState',  < Has GetSimState and SetSimState methods
%    'DisallowSimState' < Error out when saving or restoring the model sim
state
block.SimStateCompliance = 'DefaultSimState';

%% -----------------------------------------------------------------
%% The M-file S-function uses an internal registry for all
%% block methods. You should register all relevant methods
%% (optional and required) as illustrated below. You may choose
%% any suitable name for the methods and implement these methods
%% as local functions within the same file. See comments
%% provided for each function for more information.
%% -----------------------------------------------------------------

%block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
block.RegBlockMethod('InitializeConditions', @InitializeConditions);
%block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('Outputs', @Outputs);     % Required
%block.RegBlockMethod('Update', @Update);
block.RegBlockMethod('Derivatives', @Derivatives);
%block.RegBlockMethod('Terminate', @Terminate); % Required

%end setup

%%
```

```
%% PostPropagationSetup:
%%   Functionality    : Setup work areas and state variables. Can
%%                      also register run-time methods here
%%   Required         : No
%%   C-Mex counterpart: mdlSetWorkWidths
%%
%function DoPostPropSetup(block)
%block.NumDworks = 1;
%
%  block.Dwork(1).Name            = 'x1';
%  block.Dwork(1).Dimensions      = 1;
%  block.Dwork(1).DatatypeID      = 0;      % double
%  block.Dwork(1).Complexity      = 'Real'; % real
%  block.Dwork(1).UsedAsDiscState = true;
%
%  block.Dwork(2).Name            = 'x2';
%  block.Dwork(2).Dimensions      = 1;
%  block.Dwork(2).DatatypeID      = 0;      % double
%  block.Dwork(2).Complexity      = 'Real'; % real
%  block.Dwork(2).UsedAsDiscState = true;
%


%%
%% InitializeConditions:
%%   Functionality    : Called at the start of simulation and if it is
%%                      present in an enabled subsystem configured to reset
%%                      states, it will be called when the enabled subsystem
%%                      restarts execution to reset the states.
%%   Required         : No
%%   C-MEX counterpart: mdlInitializeConditions
%%
function InitializeConditions(block)

Les_0   = block.DialogPrm(1).Data;
xe_0    = block.DialogPrm(2).Data;
Tea_0   = block.DialogPrm(3).Data;

block.ContStates.Data = [Les_0; xe_0; Tea_0];

%end InitializeConditions

%%
%% Start:
%%   Functionality    : Called once at start of model execution. If you
%%                      have states that should be initialized once, this
%%                      is the place to do it.
%%   Required         : No
%%   C-MEX counterpart: mdlStart
%%
%function Start(block)

%endfunction

%%
%% Outputs:
```

```
%%   Functionality    : Called to generate block outputs in
%%                       simulation step
%%   Required         : Yes
%%   C-MEX counterpart: mdlOutputs
%%
function Outputs(block)
% parameters or constants
Le      = block.DialogPrm(4).Data;          % m
Ae      = block.DialogPrm(5).Data;          % m2
UAe     = block.DialogPrm(6).Data;          % kW/mK

% inputs
Pe      = block.InputPort(1).Data(1);
if Pe == 0
    Pe      = block.DialogPrm(8).Data;
    dPedt   = 0;
    xei     = block.DialogPrm(9).Data;
    hei     = block.DialogPrm(10).Data;
    mdei    = block.DialogPrm(11).Data;
    Teai    = block.DialogPrm(12).Data;
else
    dPedt   = block.InputPort(1).Data(2);
    xei     = block.InputPort(1).Data(3);
    hei     = block.InputPort(1).Data(4);
    mdei    = block.InputPort(1).Data(5);
    Teai    = block.InputPort(1).Data(7);
end

% state variables, xc = []'
Les     = block.ContStates.Data(1);
xe      = block.ContStates.Data(2);     if xe>1 xe=1; end
Tea     = block.ContStates.Data(3);

% Thermodyanmic properties
xeo     = xe;
[Ter1 heo rhoe1 rhoe1he1 drhoe1dPe drhoe1he1dPe drhoe1dxe drhoe1he1dxe ...
    rhoe2 rhoe2he2 drhoe2dPe drhoe2he2dPe drhoe2dxe drhoe2he2dxe] ...
    = EVAP_2ph(Pe,xe,hei,Le,Les);

Teao    = 2*Tea - Teai;

e12     = Ae*(Les*rhoe1 + (Le - Les)*rhoe2);

e21     = Ae*(rhoe1he1 - rhoe1*heo + rhoe2*heo - rhoe2he2);
e22     = Ae*Les*(drhoe1he1dxe - heo*drhoe1dxe) ...
        + Ae*(Le - Les)*(drhoe2he2dxe - heo*drhoe2dxe);

fe1     = mdei*(xei - xeo);
fe2     = mdei*(hei - heo) + UAe*(Tea - Ter1) ...
        - (Ae*Les*(drhoe1he1dPe - heo*drhoe1dPe - 1) ...
        + Ae*(Le - Les)*(drhoe2he2dPe - heo*drhoe2dPe - 1))*dPedt;

xde     = fe1/e12;
Ldes    = (e12*fe2 - e22*fe1)/e12/e21;
```

```
mdeo    = mdei - Ae*(rhoe1 - rhoe2)*Ldes ...
        - (Ae*Les*drhoe1dxe + Ae*(Le - Les)*drhoe2dxe)*xde ...
        - (Ae*Les*drhoe1dPe + Ae*(Le - Les)*drhoe2dPe)*dPedt;

block.OutputPort(1).Data = [mdeo; xeo; heo; Teao; Les];
%block.OutputPort(1).Data = block.Dwork(1).Data + block.InputPort(1).Data;

%end Outputs

%%
%% Update:
%%   Functionality    : Called to update discrete states
%%                      during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlUpdate
%%
%function Update(block)

%block.Dwork(1).Data = block.InputPort(1).Data;

%end Update

%%
%% Derivatives:
%%   Functionality    : Called to update derivatives of
%%                      continuous states during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlDerivatives
%%
function Derivatives(block)
% parameters or constants
Le      = block.DialogPrm(4).Data;          % m
Ae      = block.DialogPrm(5).Data;          % m2
UAe     = block.DialogPrm(6).Data;          % kW/mK
rhoea   = 3;                                % kg/m3
Cpea    = block.DialogPrm(7).Data;                % kJ/kgK
Aea     = Ae;                               % m2

% inputs
Pe      = block.InputPort(1).Data(1);
dPedt   = block.InputPort(1).Data(2);
xei     = block.InputPort(1).Data(3);
hei     = block.InputPort(1).Data(4);
mdei    = block.InputPort(1).Data(5);
mdea    = block.InputPort(1).Data(6);
Teai    = block.InputPort(1).Data(7);

% state variables, xc = []'
Les     = block.ContStates.Data(1);
xe      = block.ContStates.Data(2);     if xe>1 xe=1; end
Tea     = block.ContStates.Data(3);

% Thermodyanmic properties
xeo     = xe;
[Ter1 heo rhoe1 rhoe1he1 drhoe1dPe drhoe1he1dPe drhoe1dxe drhoe1he1dxe ...
```

```
    rhoe2 rhoe2he2 drhoe2dPe drhoe2he2dPe drhoe2dxe drhoe2he2dxe] ...
    = EVAP_2ph(Pe,xe,hei,Le,Les);

Teao    = 2*Tea - Teai;

e12     = Ae*(Les*rhoe1 + (Le - Les)*rhoe2);

e21     = Ae*(rhoe1he1 - rhoe1*heo + rhoe2*heo - rhoe2he2);
e22     = Ae*Les*(drhoe1he1dxe - heo*drhoe1dxe) ...
        + Ae*(Le - Les)*(drhoe2he2dxe - heo*drhoe2dxe);

e33     = rhoea*Aea*Cpea*Le;

fe1     = mdei*(xei - xeo);
fe2     = mdei*(hei - heo) + UAe*(Tea - Ter1) ...
        - (Ae*Les*(drhoe1he1dPe - heo*drhoe1dPe - 1) ...
        + Ae*(Le - Les)*(drhoe2he2dPe - heo*drhoe2dPe - 1))*dPedt;
fe3     = mdea*Cpea*(Teai - Teao) + UAe*(Ter1 - Tea);

xde     = fe1/e12;
Tdea    = fe3/e33;
Ldes    = (e12*fe2 - e22*fe1)/e12/e21;

block.Derivatives.Data = [Ldes; xde; Tdea];

%end Derivatives

%%
%% Terminate:
%%   Functionality    : Called at the end of simulation for cleanup
%%   Required         : Yes
%%   C-MEX counterpart: mdlTerminate
%%
%function Terminate(block)

%end Terminate
% End of EVAP_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- EVAP_2ph.m

```
% EVAP_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Tmean ho r1 rh1 dr1dP drh1dP dr1dx drh1dx ...
    r2 rh2 dr2dP drh2dP dr2dx drh2dx] = EVAP_2ph(P,xe1,hei,Le,Les)

load EVAP_pxz2.mat
m = 100;
dP = 1e-3;
xtol = 1e-6;

P1 = P - dP/2;
P2 = P + dP/2;
```

```
if xe1 > 1-xtol
    x1 = xe1 - xtol/2;
    x2 = 1;
elseif xe1 < xtol
    x1 = 0;
    x2 = xe1 + xtol/2;
else
    x1 = xe1 - xtol/2;
    x2 = xe1 + xtol/2;
end
dx = x2 - x1;

ze = 0:1/n:1;
[xxe ppe zze] = meshgrid(xe,Pe,ze);
z = [0 1];
[xi pii zi] = meshgrid(xe1,P,z);
he1z = interp3(xxe,ppe,zze,he,xi,pii,zi);
zei = (hei - he1z(1))/(he1z(2) - he1z(1));
zeo = zei + Le/Les*(1 - zei);
[xhi phi zhi] = meshgrid(xe1,P,zeo);
ho = interp3(xxe,ppe,zze,he,xhi,phi,zhi);

xx = [x1 xe1 x2];
pp = [P1 P P2];
z1 = zei:(1-zei)/m:1;
z2 = zeo:(1-zeo)/m:1;
zz = zei:(zeo-zei)/m:zeo; % For temperature only

[xi1 pi1 zi1] = meshgrid(xx,pp,z1);
rho1 = interp3(xxe,ppe,zze,rhoe,xi1,pi1,zi1);
r1 = mean(rho1(2,2,:));
rp1 = mean(rho1(1,2,:));
rp2 = mean(rho1(3,2,:));
rx1 = mean(rho1(2,1,:));
rx2 = mean(rho1(2,3,:));
rho1h1 = interp3(xxe,ppe,zze,rhoehe,xi1,pi1,zi1);
rh1 = mean(rho1h1(2,2,:));
rhp1 = mean(rho1h1(1,2,:));
rhp2 = mean(rho1h1(3,2,:));
rhx1 = mean(rho1h1(2,1,:));
rhx2 = mean(rho1h1(2,3,:));
dr1dP = (rp2 - rp1)/dP;
drh1dP = (rhp2 - rhp1)/dP;
dr1dx = (rx2 - rx1)/dx;
drh1dx = (rhx2 - rhx1)/dx;

[xi2 pi2 zi2] = meshgrid(xx,pp,z2);
rho2 = interp3(xxe,ppe,zze,rhoe,xi2,pi2,zi2);
r2 = mean(rho2(2,2,:));
rp1 = mean(rho2(1,2,:));
rp2 = mean(rho2(3,2,:));
rx1 = mean(rho2(2,1,:));
rx2 = mean(rho2(2,3,:));
rho2h2 = interp3(xxe,ppe,zze,rhoehe,xi2,pi2,zi2);
rh2 = mean(rho2h2(2,2,:));
rhp1 = mean(rho2h2(1,2,:));
```

```
rhp2 = mean(rho2h2(3,2,:));
rhx1 = mean(rho2h2(2,1,:));
rhx2 = mean(rho2h2(2,3,:));
dr2dP = (rp2 - rp1)/dP;
drh2dP = (rhp2 - rhp1)/dP;
dr2dx = (rx2 - rx1)/dx;
drh2dx = (rhx2 - rhx1)/dx;

[xxi ppi zzi] = meshgrid(xe1,P,zz);
Te1 = interp3(xxe,ppe,zze,Te,xxi,ppi,zzi);
Tmean = mean(Te1(1,1,:));
% End of EVAP_2ph.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Solution Heat Exchanger

- SHX_Driver.m

```
% SHX_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

%% initial values %%
% initial inputs %
Ph          = 1.791850466000000e+003;
hsci        = -99.908784049784416;
xsci        = 0.62;
hshi        = 1.777520026131992e+002;
xshi        = 0.468975972268840;
mdsci       = 0.185995325421376;
mdsho       = 0.154747876421376;

% Geometry and parameters %
Ls          = 2.032;
Asc         = 0.002696964962284;
UAs         = 1.795847738116563;

% Initial values of state variables %
Tsc         = 3.206828516611801e+002;
xsc         = 0.62;
Tsh         = 3.403698485178869e+002;
xsh         = 0.46897597226884;

 %% Step inputs %%
stime       = 0;

% Pressure
Phf         = Ph;%*1.01;
Phslope     = 0;%0.5;          % step input for dPadt
Phst        = stime;        % sec. - start time

% Ammonia mass fraction and enthalpy
xscif       = xsci;%*1.1;    % final value
xscist      = stime;        % step time

% Enthalpy
```

```
hscif       = hsci;%*0.9;
hscist      = stime;

% Ammonia mass fraction and enthalpy
xshif       = xshi;%*1.1;    % final value
xshist      = stime;        % step time

% Enthalpy
hshif       = hshi;%*1.1;
hshist      = stime;

% Mass flowrate - refrigerant
mdscif      = mdsci*1.1;    % final value
mdscist     = stime;        % step time

% Mass flow rate - secondary fluid
mdshof      = mdsho;%*1.1;    % final value
mdshost     = stime;        % step time

%% Run Simulink Model %%
sim('SHX_Sim');
% End of SHX_Driver.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- SHX_Driver.m

```
% SHX_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function SHX_Dynamics(block)
% Dynamic modeling of an SHX
% This file works with SHX_sfunc2.mdl and SHX_ts_init1.m
%
% This S-Function is utilized in the SIMULINK file named SHX_sfunc2.mdl
% Written by ChoonJae Ryu, Energy and Gas Dynamics Lab (Prof. Lear).
% Department of Mechanical and Aerospace Engineering, University of Florida

%%
%% The setup method is used to setup the basic attributes of the
%% S-function such as ports, parameters, etc. Do not add any other
%% calls to the main body of the function.
%%
setup(block);

%endfunction

%% Function: setup ===================================================
%% Abstract:
%%   Set up the S-function block's basic characteristics such as:
%%   - Input ports
%%   - Output ports
%%   - Dialog parameters
%%   - Options
%%
%%   Required         : Yes
%%   C-Mex counterpart: mdlInitializeSizes
%%
```

```
function setup(block)

% Register parameters
block.NumDialogPrms = 14;

% Register number of ports
block.NumInputPorts  = 1;
block.NumOutputPorts = 1;

% Setup port properties to be inherited or dynamic
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

% Override input port properties
block.InputPort(1).Dimensions        = 8;       % size_inputs
%block.InputPort(1).DatatypeID  = 0;  % double
%block.InputPort(1).Complexity  = 'Real';
block.InputPort(1).DirectFeedthrough = 0;   % size_feedthrough

% Override output port properties
block.OutputPort(1).Dimensions       = 6;       % size_outputs
%block.OutputPort(1).DatatypeID  = 0; % double
%block.OutputPort(1).Complexity  = 'Real';

% Register sample times
%  [0 offset]            : Continuous sample time
%  [positive_num offset] : Discrete sample time
%
%  [-1, 0]               : Inherited sample time
%  [-2, 0]               : Variable sample time
block.SampleTimes = [0 0];

% States
block.NumContStates = 4; % xe = [le1 Pe heo Tew1 Tew2]'

% Specify the block simStateComliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume
DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'CustomSimState',  < Has GetSimState and SetSimState methods
%    'DisallowSimState' < Error out when saving or restoring the model sim
state
block.SimStateCompliance = 'DefaultSimState';

%% -----------------------------------------------------------------
%% The M-file S-function uses an internal registry for all
%% block methods. You should register all relevant methods
%% (optional and required) as illustrated below. You may choose
%% any suitable name for the methods and implement these methods
%% as local functions within the same file. See comments
%% provided for each function for more information.
%% -----------------------------------------------------------------

%block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
```

```
block.RegBlockMethod('InitializeConditions', @InitializeConditions);
%block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('Outputs', @Outputs);     % Required
%block.RegBlockMethod('Update', @Update);
block.RegBlockMethod('Derivatives', @Derivatives);
%block.RegBlockMethod('Terminate', @Terminate); % Required

%end setup

%%
%% PostPropagationSetup:
%%   Functionality    : Setup work areas and state variables. Can
%%                      also register run-time methods here
%%   Required         : No
%%   C-Mex counterpart: mdlSetWorkWidths
%%
%function DoPostPropSetup(block)
%block.NumDworks = 1;
%
%  block.Dwork(1).Name            = 'x1';
%  block.Dwork(1).Dimensions      = 1;
%  block.Dwork(1).DatatypeID      = 0;      % double
%  block.Dwork(1).Complexity      = 'Real'; % real
%  block.Dwork(1).UsedAsDiscState = true;
%
%  block.Dwork(2).Name            = 'x2';
%  block.Dwork(2).Dimensions      = 1;
%  block.Dwork(2).DatatypeID      = 0;      % double
%  block.Dwork(2).Complexity      = 'Real'; % real
%  block.Dwork(2).UsedAsDiscState = true;
%


%%
%% InitializeConditions:
%%   Functionality    : Called at the start of simulation and if it is
%%                      present in an enabled subsystem configured to reset
%%                      states, it will be called when the enabled subsystem
%%                      restarts execution to reset the states.
%%   Required         : No
%%   C-MEX counterpart: mdlInitializeConditions
%%
function InitializeConditions(block)

hsc_0   = block.DialogPrm(1).Data;
xsc_0   = block.DialogPrm(2).Data;
hsh_0   = block.DialogPrm(3).Data;
xsh_0   = block.DialogPrm(4).Data;

block.ContStates.Data = [hsc_0; xsc_0; hsh_0; xsh_0];

%end InitializeConditions

%%
%% Start:
%%   Functionality    : Called once at start of model execution. If you
```

```
%%                      have states that should be initialized once, this
%%                      is the place to do it.
%%   Required         : No
%%   C-MEX counterpart: mdlStart
%%
%function Start(block)

%endfunction

%%
%% Outputs:
%%   Functionality    : Called to generate block outputs in
%%                      simulation step
%%   Required         : Yes
%%   C-MEX counterpart: mdlOutputs
%%
function Outputs(block)

Ls      = block.DialogPrm(5).Data;          % m
Asc     = block.DialogPrm(6).Data;          % m2
UAs     = block.DialogPrm(7).Data;          % kW/mK
Ash     = Asc;                              % m2

Ph      = block.InputPort(1).Data(1);
if Ph == 0
    Ph      = block.DialogPrm(8).Data;
    dPhdt   = 0;
    hsci    = block.DialogPrm(9).Data;
    xsci    = block.DialogPrm(10).Data;
    hshi    = block.DialogPrm(11).Data;
    xshi    = block.DialogPrm(12).Data;
    mdsci   = block.DialogPrm(13).Data;
    mdsho   = block.DialogPrm(14).Data;
else
    dPhdt   = block.InputPort(1).Data(2);
    hsci    = block.InputPort(1).Data(3);
    xsci    = block.InputPort(1).Data(4);
    hshi    = block.InputPort(1).Data(5);
    xshi    = block.InputPort(1).Data(6);
    mdsci   = block.InputPort(1).Data(7);
    mdsho   = block.InputPort(1).Data(8);
end

Tsc     = block.ContStates.Data(1);
xsc     = block.ContStates.Data(2);     if xsc>1 xsc=1; end
Tsh     = block.ContStates.Data(3);
xsh     = block.ContStates.Data(4);     if xsh>1 xsh=1; end

hsc     = liqprop_ibrahim(Tsc,Ph,xsc);
hsh     = liqprop_ibrahim(Tsh,Ph,xsh);

xsco    = xsc;                  if xsco>1 xsco=1; end
xsho    = xsh;                  if xsho>1 xsho=1; end
hsco    = 2*hsc - hsci;
hsho    = 2*hsh - hshi;
```

```
[rhosc hsc drhoscdPh dhscdPh drhoscdxsc dhscdxsc ...
    drhoscdTsc dhscdTsc] = liqp(Tsc,Ph,xsc);
[rhosh hsh drhoshdPh dhshdPh drhoshdxsh dhshdxsh ...
    drhoshdTsh dhshdTsh] = liqp(Tsh,Ph,xsh);

s11     = Asc*Ls*(xsc - xsco)*drhoscdTsc;
s12     = Asc*Ls*((xsc - xsco)*drhoscdxsc + rhosc);
s1p     = Asc*Ls*(xsc - xsco)*drhoscdPh;

s21     = Asc*Ls*((hsc - hsco)*drhoscdTsc + rhosc*dhscdTsc);
s22     = Asc*Ls*((hsc - hsco)*drhoscdxsc + rhosc*dhscdxsc);
s2p     = Asc*Ls*((hsc - hsco)*drhoscdPh + rhosc*dhscdPh - 1);

s33     = Ash*Ls*(xsh - xshi)*drhoshdTsh;
s34     = Ash*Ls*((xsh - xshi)*drhoshdxsh + rhosh);
s3p     = Ash*Ls*(xsh - xshi)*drhoshdPh;

s43     = Ash*Ls*((hsh - hshi)*drhoshdTsh + rhosh*dhshdTsh);
s44     = Ash*Ls*((hsh - hshi)*drhoshdxsh + rhosh*dhshdxsh);
s4p     = Ash*Ls*((hsh - hshi)*drhoshdPh + rhosh*dhshdPh - 1);

fs1     = mdsci*(xsci - xsco) - s1p*dPhdt;
fs2     = mdsci*(hsci - hsco) + UAs*(Tsh - Tsc) - s2p*dPhdt;
fs3     = mdsho*(xshi - xsho) - s3p*dPhdt;
fs4     = mdsho*(hshi - hsho) + UAs*(Tsc - Tsh) - s4p*dPhdt;

detSC   = s11*s22 - s12*s21;
Tdsc   = (s22*fs1 - s12*fs2)/detSC;
xdsc   = (s11*fs2 - s21*fs1)/detSC;
detSH   = s33*s44 - s34*s43;
Tdsh   = (s44*fs3 - s34*fs4)/detSH;
xdsh   = (s33*fs4 - s43*fs3)/detSH;

mdsco   = mdsci - Asc*Ls*(drhoscdPh*dPhdt + drhoscdTsc*Tdsc ...
        + drhoscdxsc*xdsc);
mdshi   = mdsho + Ash*Ls*(drhoshdPh*dPhdt + drhoshdTsh*Tdsh ...
        + drhoshdxsh*xdsh);

block.OutputPort(1).Data = [mdsco; xsco; hsco; mdshi; xsho; hsho];
%block.OutputPort(1).Data = block.Dwork(1).Data + block.InputPort(1).Data;

%end Outputs

%%
%% Update:
%%   Functionality    : Called to update discrete states
%%                      during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlUpdate
%%
%function Update(block)

%block.Dwork(1).Data = block.InputPort(1).Data;
```

```
%end Update

%%
%% Derivatives:
%%   Functionality    : Called to update derivatives of
%%                      continuous states during simulation step
%%   Required         : No
%%   C-MEX counterpart: mdlDerivatives
%%
function Derivatives(block)
% parameters or constants
Ls      = block.DialogPrm(5).Data;          % m
Asc     = block.DialogPrm(6).Data;          % m2
UAs     = block.DialogPrm(7).Data;          % kW/mK
Ash     = Asc;                              % m2

% inputs [Pc dPcdt xci mdci mdca Tcai]
Ph      = block.InputPort(1).Data(1);
dPhdt   = block.InputPort(1).Data(2);
hsci    = block.InputPort(1).Data(3);
xsci    = block.InputPort(1).Data(4);
hshi    = block.InputPort(1).Data(5);
xshi    = block.InputPort(1).Data(6);
mdsci   = block.InputPort(1).Data(7);
mdsho   = block.InputPort(1).Data(8);

% state variables, xc = []'
Tsc     = block.ContStates.Data(1);
xsc     = block.ContStates.Data(2);     if xsc>1 xsc=1; end
Tsh     = block.ContStates.Data(3);
xsh     = block.ContStates.Data(4);     if xsh>1 xsh=1; end

hsc     = liqprop_ibrahim(Tsc,Ph,xsc);
hsh     = liqprop_ibrahim(Tsh,Ph,xsh);

xsco    = xsc;                  if xsco>1 xsco=1; end
xsho    = xsh;                  if xsho>1 xsho=1; end
hsco    = 2*hsc - hsci;
hsho    = 2*hsh - hshi;

[rhosc hsc drhoscdPh dhscdPh drhoscdxsc dhscdxsc ...
    drhoscdTsc dhscdTsc] = liqp(Tsc,Ph,xsc);
[rhosh hsh drhoshdPh dhshdPh drhoshdxsh dhshdxsh ...
    drhoshdTsh dhshdTsh] = liqp(Tsh,Ph,xsh);

s11     = Asc*Ls*(xsc - xsco)*drhoscdTsc;
s12     = Asc*Ls*((xsc - xsco)*drhoscdxsc + rhosc);
s1p     = Asc*Ls*(xsc - xsco)*drhoscdPh;

s21     = Asc*Ls*((hsc - hsco)*drhoscdTsc + rhosc*dhscdTsc);
s22     = Asc*Ls*((hsc - hsco)*drhoscdxsc + rhosc*dhscdxsc);
s2p     = Asc*Ls*((hsc - hsco)*drhoscdPh + rhosc*dhscdPh - 1);

s33     = Ash*Ls*(xsh - xshi)*drhoshdTsh;
```

```
s34     = Ash*Ls*((xsh - xshi)*drhoshdxsh + rhosh);
s3p     = Ash*Ls*(xsh - xshi)*drhoshdPh;

s43     = Ash*Ls*((hsh - hshi)*drhoshdTsh + rhosh*dhshdTsh);
s44     = Ash*Ls*((hsh - hshi)*drhoshdxsh + rhosh*dhshdxsh);
s4p     = Ash*Ls*((hsh - hshi)*drhoshdPh + rhosh*dhshdPh - 1);

fs1     = mdsci*(xsci - xsco) - s1p*dPhdt;
fs2     = mdsci*(hsci - hsco) + UAs*(Tsh - Tsc) - s2p*dPhdt;
fs3     = mdsho*(xshi - xsho) - s3p*dPhdt;
fs4     = mdsho*(hshi - hsho) + UAs*(Tsc - Tsh) - s4p*dPhdt;

detSC   = s11*s22 - s12*s21;
Tdsc    = (s22*fs1 - s12*fs2)/detSC;
xdsc    = (s11*fs2 - s21*fs1)/detSC;
detSH   = s33*s44 - s34*s43;
Tdsh    = (s44*fs3 - s34*fs4)/detSH;
xdsh    = (s33*fs4 - s43*fs3)/detSH;

block.Derivatives.Data = [Tdsc; xdsc; Tdsh; xdsh];
%end Derivatives

%%
%% Terminate:
%%   Functionality    : Called at the end of simulation for cleanup
%%   Required         : Yes
%%   C-MEX counterpart: mdlTerminate
%%
%function Terminate(block)

%end Terminate
% End of SHX_Dynamics.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

LIST OF REFERENCES

[1] Anxionnaz, R., 1945, “Installation a Turbines a Gaz a Circuit Semi-Ouvert,” French Patent 999-133.

[2] Anxionnaz, R., 1948, “Improvements in or Relating to Gas Turbine Plant with Semi-Open Circuit,” British Patent 651-166.

[3] Lear, W. E., and Sherif, S. A., “Combined Cooling and Power Plant with Water Extraction,” US Patent 20060037337.

[4] Kalina, A. I., 1983, “Combined Cycle and Waste Heat Recovery Power Systems Based on a Novel Thermodynamic Energy Cycle Utilizing Low-Temperature Heat for Power Generation,” ASME Paper No. 83-JPGC-GT-3.

[5] Pátek, J., and Klomfar, J., 1995, “Simple Functions for Fast Calculations of Selected Thermodynamic Properties of the Ammonia-Water System,” Int. J. Refrig., **18**(4), pp. 228-234.

[6] Tillner-Roth, R., and Friend, D. G., 1998, “Survey and Assessment of Available Measurements on Thermodynamic Properties of the Mixture {Water + Ammonia},” J. Phys. Chem. Ref. Data, **27**(1), pp. 45-61.

[7] Gillespi, P. C., Wilding, W. V., and Wilson, G. M., 1985, “Vapor-Liquid Equilibrium Measurements on the Ammonia-Water System from 313K to 589K,” Project 758-81, Itec Research Co., Inc. Provo, Utah.

[8] Macriss, R. A., Eakin, B. E., Ellington, R. T., and Huebler, J., 1964, “Physical and Thermodynamic Properties of Ammonia-Water Mixtures, Research bulletin, No. 34, Institute of Gas Technology, Chicago, IL.

[9] Park, Y. M., and Sonntag, R. E., 1990, “Thermodynamic Properties of Ammonia-Water Mixtures: a Generalized Equation-of-State Approach,” ASHRAE Trans., **96**, pp. 150-159.

[10] Tillner-Roth, R., and Friend, D. G., 1998, “A Helmholtz Free Energy Formulation of the Thermodynamic Properties of the Mixture {Water + Ammonia},” J. Phys. Chem. Ref. Data, **27**(1), pp. 63-96.

[11] El-Sayed, Y. M., and Tribus, M., 1985, “Thermodynamic Properties of Water-Ammonia Mixtures Theoretical Implementation for Use in Power Cycles Analysis,” ASME Pub. AES **1**, pp. 89-95.

[12] Bogart, M., 1981, *Ammonia Absorption Refrigeration in Industrial Processes*, Gulf Pub. Co., Houston, TX.

[13] Xu, F., and Goswami, D. Y., 1999, “Thermodynamic Properties of Ammonia-Water Mixtures for Power-Cycle Applications,” Energy, **24**(6), pp. 525-536.

[14] Tamm, G. O., 2003, “Experimental Investigation of An Ammonia-Based Combined Power and Cooling Cycle,” Ph.D. Dissertation, University of Florida, Gainesville, FL.

[15] Ibrahim, O. M., and Klein, S. A., 1993, “Thermodynamic Properties of Ammonia-Water Mixtures,” ASHRAE Trans., **99**, pp. 1495-1502.

[16] Ryu, C., Tiffany, D. R., Crittenden, J. F., Lear, W. E., and Sherif, S. A., 2010, “Dynamic Modeling of A Novel Cooling, Heat, Power, and Water Microturbine Combined Cycle,” J. Energ. Resour.-ASME, **132**(2), pp. 021006-1-9.

[17] Khan, J., 2006, “Design and Optimization of A Distributed Generation System with the Production of Water and Refrigeration,” Ph.D. Dissertation, University of Florida, Gainesville, FL.

[18] Adewusi, S. A., and Zubair, S. M., 2004, “Second Law Based Thermodynamic Analysis of Ammonia-Water Absorption Systems,” Energ. Convers. Manage., **45**(15-16) pp. 2355-2369.

[19] Kim, B., and Park, J., 2007, “Dynamic Simulation of A Single-Effect Ammonia-Water Absorption Chiller,” Int. J. Refrig., **30**(3) pp. 535-545.

[20] MacArthur, J. W., and Grald, E. W., 1989, “Unsteady Compressible Two-Phase Flow Model for Predicting Cyclic Heat Pump Performance and A Comparison with Experimental Data,” Int. J. Refrig., **12**(1), pp. 29-41.

[21] He, X. D., Liu, S., and Asada, H. H., 1997, “Modeling of Vapor Compression Cycles for Multivariable Feedback Control of HVAC Systems,” J. Dyn. Syst.-T ASME, **119**(2), pp. 183-191.

[22] Jensen, J. M., and Tummescheit, H., 2002, “Moving Boundary Models of Dynamic Simulations of Two-Phase Flows,” Proceedings Int. Modelica Conf. **2**, pp. 235-244.

[23] Wedekind, G. L., Bhatt, B. L., and Beck, B. T., 1978, “A System Mean Void Fraction Model for Predicting Various Transient Phenomena Associated with Two-Phase Evaporating and Condensing Flows,” Int. J. Multiphas. Flow, **4**(1), pp. 97-114.

[24] Zivi, S. M., 1964, “Estimation of Steady-State Steam Void Fraction by Means of the Principle of Minimum Entropy Production,” J. Heat Transfer, **86**, pp. 247-252.

[25] Schulz, S. C. G., 1971, “Equations of State for the System Ammonia-Water for Use with Computers,” Proceedings Int. Congr. Refrig. **2**, pp. 431-436.

[26] Ziegler, B., and Trepp, Ch.,1984, “Equation of State for Ammonia-Water Mixtures,” Int. J. Refrig., **7**(2), pp. 101-106.

[27] M. Conde Eng., 2006, “Thermophysical Properties of {NH3 + H2O} Mixtures for the Industrial Design of Absorption Refrigeration Equipment,” Manuel Conde Engineering, Switzerland, Zurich.

[28] Lear, W. E., and Laganelli, A. L., 1999, “High Pressure Regenerative Turbine Engine: 21st Century Propulsion,” Final Test Report, Contract No. NAS3-27396.14

[29] Khan, J. R., Lear, W. E., Sherif, S. A., and Crittenden, J. F., 2008, "Performance of a Novel Combined Cooling and Power Gas Turbine with Water Harvesting," J. Eng. Gas. Turb. Power, **130**(4), pp. 041702-1-10

[30] Goswami, D. Y., Kreith, F., and Kreider, J. F., 2000, *Principles of Solar Engineering*, 2nd ed., Taylor and Francis, Philadelphia, PA.

[31] Parsons, R. A., 1998, *ASHRAE Handbook – Refrigeration*, SI ed., ASHRAE, Atlanta, GA.

[32] Gustafson, M. W., Baylor, J. S., and Epstein, G., 1993, "Estimating Air Conditioning Load Control Effectiveness Using An Engineering Model," IEEE T. Power Syst. **8**(3), pp. 972-978.

[33] Threlkeld, J. L., 1962, *Thermal Environmental Engineering*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

[34] MacFarlane, R. S., and Lear, W. E., 1997, “System Impact of H2O Production and Injection on a Novel Semi-Closed Cycle Gas Turbine,” Paper No. AIAA-97-0374.

[35] Mattingly, J. D., 1996, *Elements of Gas Turbine Propulsion*, McGraw-Hill, Inc., New York.

[36] Carcasci, C., and Facchini, B., 2000, “Comparison between Two Gas Turbine Solutions to Increase Combined Power Plant Efficiency,” Energ. Convers. Manage. **41**(8), pp. 757-773.

[37] Fiaschi, D., Lombardi, L., and Tapinassi, L., 2004, “The Recuperative Auto Thermal Reforming and Recuperative Reforming Gas Turbine Power Cycles with CO2 Removal - Part II: The Recuperative Reforming Cycle,” J. Eng. Gas. Turb. Power, **126**(1), pp. 62-68.

[38] Boza, J. J., Lear, W. E., and Sherif, S. A., 2003, “Performance of a Novel Semi-Closed Gas Turbine Refrigeration Combined Cycle,” ASME Paper No. 2003-GT-38576.

[39] Proeschel, R., 2002, http://www.proepowersystems.com/proe90.htm

[40] Massardo, F., Cazzola, W., and Lagorio, G., 2004, “Widget-Temp: A Novel Web-Based Approach for the Thermoeconomic Analysis and Optimization of Conventional and Innovative Cycles,” ASME Paper No. 2004-GT-54115

[41] Lear, W. E., Ryu, C. J., Crittenden, J. F., Srinivasan, A., Ellis, W., Tiffany, D. R., and Sherif, S. A., 2008, “System Design of a Novel Combined Cooling, Heat, Power, and Water Microturbine Combined Cycle,” ASME Paper No. GT2008-51454

[42] Huang, M. C., Chen, B. R., Hsiao, M. J., and Chen, S. L., 2007, “Application of Thermal Battery in the Ice Storage Air-Conditioning System as a Subcooler,” Int. J. Refrig. **30**(2), pp. 245-253

[43] MacArthur, J. W. and Grald, E. W., 1989, “Unsteady Compressible Two-Phase Flow Model for Predicting Cyclic Heat Pump Performance and a Comparison with Experimental Data,” Int. J. Refrig. **12**(1), pp. 29-41.

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

BIOGRAPHICAL SKETCH

Choon Jae Ryu was born in Busan, South Korea, in 1977. He graduated from Gyeongsang University auxiliary high school in Jinju, Gyeongnam Province, South Korea, in 1995. In 2001, he received a Bachelor of Engineering in automotive engineering from Korea Maritime University with a Summa Cum Laude honor in engineering. He joined Dr. Rodney S. Rouff's research group in 2002 as a pre-doctoral fellow and studied about carbon nano tube in Northwestern University in Evanston, IL. After finishing research in Northwestern University, he joined Dr. SungTak Ro's research group in 2003. His research area was about mild hybrid electric vehicle. In 2005, he received a Master of Science in mechanical and aerospace engineering from Seoul National University. He worked for Siemens Automotive Company in Icheon, Gyeonggi Province, South Korea and Next Generation Vehicle (NGV), a division of Hyundai Motor Company, in Seoul, South Korea after graduation. He started his Ph.D. studies at the University of Florida in the Department of Mechanical and Aerospace Engineering and joined Dr. William E. Lear's group in 2007. He obtained his Doctor of Philosophy in mechanical engineering from University of Florida in 2011.