2017

Modeling Two Phase Flow Heat Exchangers for Next Generation Aircraft

Hayder Hasan Jaafar Al-sarraf
Wright State University

MODELING TWO PHASE FLOW HEAT EXCHANGERS FOR NEXT GENERATION AIRCRAFT

A thesis submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Mechanical Engineering

By

HAYDER HASAN JAAFAR AL-SARRAF

B.Sc. Mechanical Engineering, Kufa University, 2005

2017

Wright State University

WRIGHT STATE UNIVERSITY
GRADUATE SCHOOL

July 3, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Hayder Hasan Jaafar Al-sarraf Entitled Modeling Two Phase Flow Heat Exchangers for Next Generation Aircraft BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Mechanical Engineering.

_____

Rory Roberts, Ph.D.
Thesis Director

_____

Joseph C. Slater, Ph.D., P.E.
Department Chair

Committee on Final Examination

_____

Rory Roberts, Ph.D.

_____

James Menart, Ph.D.

_____

Mitch Wolff, Ph.D.

_____

Robert E. W. Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate

ABSTRACT

Al-sarraf, Hayder Hasan Jaafar. M.S.M.E. Department of Mechanical and Materials Engineering, Wright State University, 2017. *Modeling Two Phase Flow Heat Exchangers for Next Generation Aircraft.*

Two-phase heat exchangers offer the potential of significant energy transfer by taking advantage of the latent heat of vaporization as the working fluid changes phase. Unfortunately, the flow physics of the phase change process is very complex and there are significant gaps in the fundamental knowledge of how several key parameters are affected by the phase change process. Therefore, an initial investigation modeling a two-phase flow heat exchanger has been accomplished. Many key assumptions have been defined which are critical to modeling two-phase flows. This research lays an initial foundation on which further investigations can build upon. Two-phase heat exchangers will be a critical enabling technology for several key aerospace advancements in the 21st century.

In this research, modeling two- phase flow heat exchangers to be used in modeling of NASA's next generation aircraft (N3- X) is accomplished. The heat exchanger model, which could be a condenser or an evaporator, currently accommodates two working fluids; kerosene (jet fuel) and a refrigerant (R134a).

The primary goal is to obtain a dynamic, robust model by using numerical simulation tools (MATLAB/ SIMULINK) which can simulate the system efficiently and would be

used in the conceptual aircraft (N3-X) model. The final goal of this project is to investigate the influence of pressure and enthalpy perturbations on the system. In other words, how quickly this system responds to change to perturbations, therefore the model will be transient.

Two examples are used for demonstration of the transient response of a two- phase heat exchanger to a perturbation in pressure and enthalpy. Initially, pressure perturbation variation effects on how the quality of R134a effects the magnitude of the two- phase flow heat transfer coefficient, therefore the two- phase heat transfer rate calculated. This changing pressure approach used to provide a rapid thermal response to a rapid thermal load variation. Other conventional thermal methods (decreasing the temperature of the cold fluid or increasing the mass flow rate) results in slower response times than changing the pressure. For this analysis, a sample time of 0.000001 seconds was used.

In addition, an enthalpy perturbation was investigated. Since, changing pressure suddenly from higher value (650 kPa) to the lower value (555 kPa) is not a real, physical scenario in life, the pressure change with transfer function would be employed to transform the system into first order system with two different time constants. Eventually, the time constant of the system plays a significant role in obtaining a quicker response.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

BLI	Boundary Layer Ingestion
BWB	Blended Wing Body
HTS	High Temperature Superconductors
HWB	Hybrid Wing Body
N3-X	NASA Next Generation Aircraft
PHE	Plate Heat Exchanger
OSF	Offset Strip fin heat exchanger
NASA	National Aeronautics and Space Administration
s	Transverse spacing (free flow width), mm
h	Free flow height, mm
t	Fin thickness, mm
l	Fin length, mm
t_f	Plate thickness, mm
L	Heat exchanger length, mm
α	s/h ratio
δ	t/l ratio
γ	t/s ratio
β	surface compactness factor, (m^2/m^3)
f	Fanning friction factor, dimensionless
Re	Reynolds number, dimensionless
Pr	Prandtl number, dimensionless
Nu	Nusselt number, dimensionless
μ	Dynamic viscosity, pa- sec
c_p	Specific heat at constant pressure, j/kg-k
c_v	Specific heat at constant volume, j/kg-k

k	Thermal conductivity, w/m-k
ΔP	Pressure drop, pa
ṁ	Mass flow rate, kg/sec
D_h	Hydraulic diameter, m
A_c	Cross sectional area, m^2
A_s	Surface area, m^2
h_l	Convective heat transfer coefficient @ liquid phase, w/m^2-k
h_G	Convective heat transfer coefficient @ vapor phase, w/m^2-k
h_{TP}	Heat transfer coefficient @ two phase, w/m^2-k
Co	Convective number, dimensionless
Bo	Boiling number, dimensionless
F_{fl}	Fluid dependent parameter, dimensionless
rho_l	Liquid density, kg/m^3
rho_G	Gas density, kg/m^3
x	Quality, dimensionless
q''	Heat transfer per unit area, w/m^2
G	Mass flux, kg/m^2- sec
h_{lG}	Enthalpy of evaporation, j/kg
h	Specific enthalpy, j/kg
u	Internal energy, j/kg
p	Pressure, pa
v	Specific volume, m^3/kg
T_o	Dead-state temperature, k
Tc	Cryogenic temperature, k
hin	Enthalpy in, j/kg
hout	Enthalpy out, j/kg
Q̇	Heat transfer rate, w/sec
nm	Nautical mile

Subscripts

m	Average value for the two-phase mixture
wall	Fluid near the wall
l	Liquid phase
G	Gas phase
TP	Two- phase
HX	Heat exchanger
c	Cold side
h	Hot side
in	Inlet
out	Outlet
irr	Irreversible

ACKNOWLEDGEMENTS

Primarily, I would like to give big thanks to my thesis directors; Dr. Mitch Wolff, and Dr. Rory Roberts. They have not treated me as a student, they have worked with me as a colleague. I have learned lots of knowledge from them regarding research skills, selecting appropriate courses, and becoming a good researcher. Both have already given me many and many of hours to help me out solve programming issues, and teach me even fundamentals. In the beginning of doing this research, they encouraged me to learn new software enabling me to accomplish my thesis smoothly such as MATLAB/ Simulink software, and Engineering Equations Solver, EES. They guided me wisely and considered my time sufficiently. The most wonderful thing, they have open – door policy so that I could stop by to ask them and then solve all problems I had.

Additionally, I would also like to thank my fellow students whom have been working on the electric aircraft project headed by Dr. Mitch Wolff and Dr. Rory Roberts for their support and advice. These indispensable students are: Jay Vora, Abada, Hashim Hameed, Al Agele, Saif Shamil Hamzah, and Foshee, Robert.

Also, my thanks go towards my sponsor, Higher Committee of Education Development in Iraq, HCED who granted me full funded scholarship. This amazing opportunity enables me to get advanced education from Wright state University.

Finally, unlimited thanks to my father, mother, brother, and sisters for their efforts to make this study successful and stress less.

In short, I am so gratitude that I am a person who surrounded with this spectacular environment.

INTRODUCTION

Overview – NASA 2035 Commercial Aircraft Concept

There has been and continues noticeable increases in aviation demands over several decades [1-3], Figure 1. This increased number of flights results in the amount of hydrocarbon fuels burned increasing as well. The hydrocarbon- fossil fuels worldwide are a limited resource. Typically, fossil fuel becomes more expensive day by day [1]. Figure 2 illustrates the rising of price and consumption for different types of fuels. From another perspective, the global concerns are rising regarding the environment, less pollution (i.e., the greenhouse gases emissions). Therefore, NASA has proposed a new generation of aircraft (N3-X) to address these various issues. Figure 2 shows how much increases are in aviation demands.

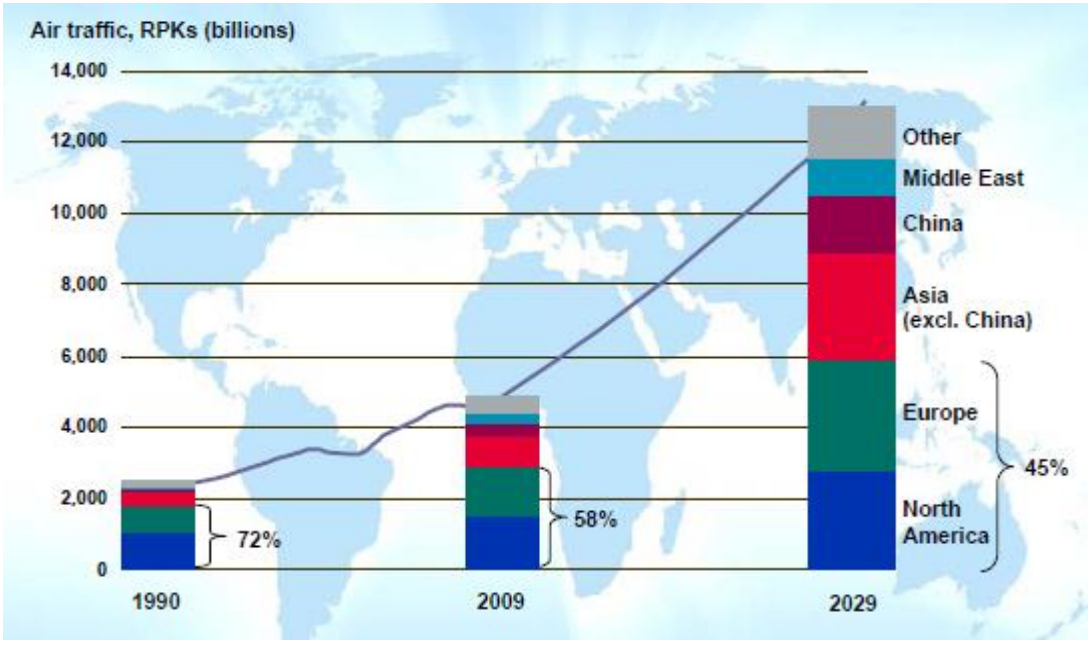


Figure 1: The increase in aviation demands [2]

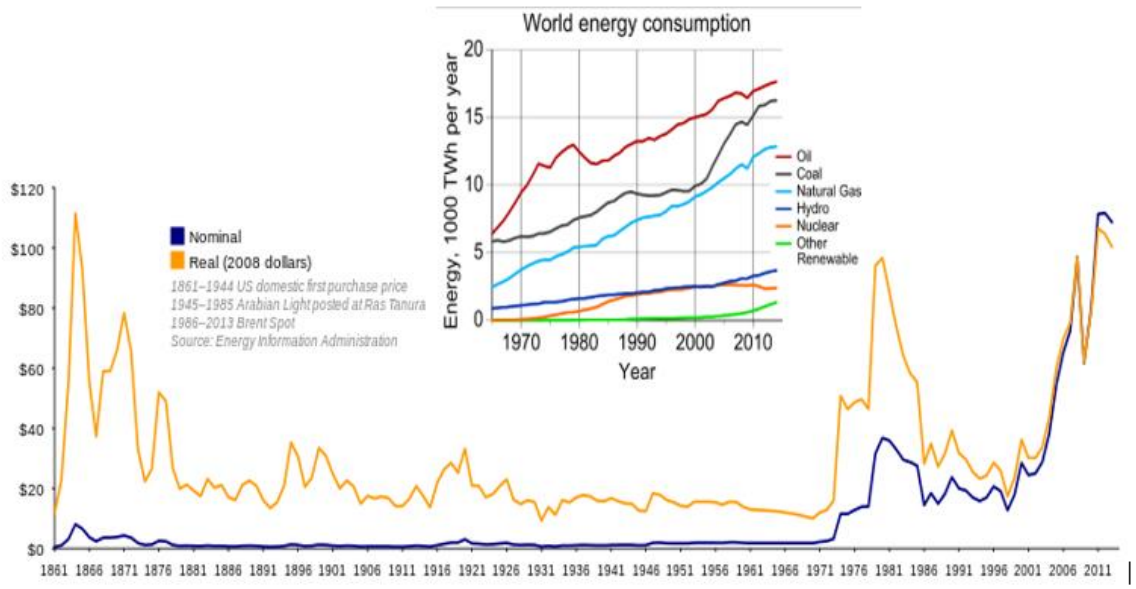


Figure 2: The increase in price and consumption of miscellaneous types of fuels [2]

The main features of this aircraft are a significant departure from conventional aircraft. To address the various performance issues, the new aircraft model must have many improvements in comparison with current aircraft. The existing baseline aircraft used by NASA is the Boeing 777-200LR [1] [2]. NASA has proposed a commercial subsonic aircraft working completely by electric power to obtain the desired performance goals. There are no hydraulic and pneumatic subsystems on the board of aircraft. Consequently, minimizing weight, complexity, operating and maintenance cost. In addition, they propose a hybrid wing body (HWB) or blended wing body (BWB). An advantage of this type of aircraft design is by increasing the boundary layer ingestion (BLI) into the propulsion system overall aircraft drag is decreased [3]. These advancements are on the external shape level. Internally, they will make use of high temperature superconductor (HTS) technology to substantially improve the usage of the electric power generated [4]. The electric power comes from two wingtip turbo generators and passes through HTS transmission cables. Figure 3 shows the proposed N3-X aircraft concept. Figure 4 gives

details of the wingtip turbo generators and transmission cables.

Figure 3: Proposed NASA N3-X aircraft.

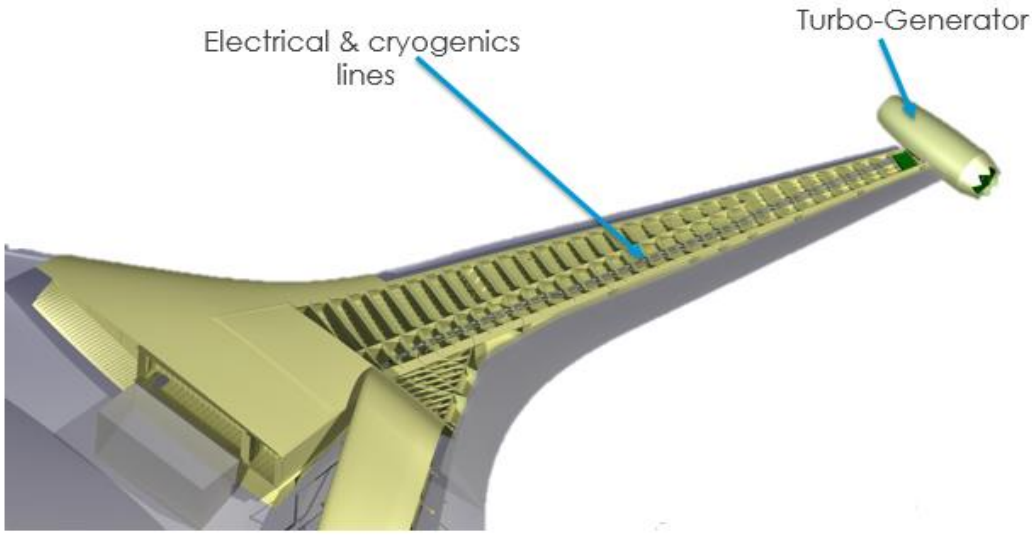


Figure 4: N3-X wingtip turbo generator and transmission cables [2]

The various goals of this conceptual aircraft are to address noise, fuel consumption, pollution, and mission length [5] [4]. These goals are listed in Table 1.

Table 1: NASA conceptual design goals [4]

CORNERS OF THE TRADE SPACE	N+1 (2015)*** Technology Benefits Relative to a Single Aisle Reference Configuration	N+2 (2020)*** Technology Benefits Relative to a Large Twin Aisle Reference Configuration	N+3 (2025)*** Technology Benefits
Noise (cum below Stage 4)	- 32 dB	- 42 dB	- 71 dB
LTO NOx Emissions (below CAEP 6)	-60%	-75%	better than -75%
Performance: Aircraft Fuel Burn	-33%**	-50%**	better than -70%
Performance: Field Length	-33%	-50%	exploit metroplex* concepts

*** Technology Readiness Level for key technologies = 4-6
** Additional gains may be possible through operational improvements
* Concepts that enable optimal use of runways at multiple airports within the metropolitan areas

Technical Specifications

This electric aircraft is a significant departure from traditional aircraft. NASA has proposed several innovative technologies, which will be highlighted. The propulsion system will be changed to a distributed propulsion system consisting of 14 electric fans driven by HTS motors [6]. This array of fans is located in the rear of the fuselage maximizing the boundary layer ingestion. Consequently, the thrust is obtained with reduced drag. In addition, high temperature superconductor technology will be employed in this aircraft. Therefore, motors, rectifiers, inverters, and power lines will be made from superconductor components only [7] [8]. The beauty of using these types of conductors is that the resistance will be zero if the operating temperature is at the appropriate level (i.e. a cryogenic temperature) [9]. Figure 5 demonstrates the relationship between the resistance and temperature for normal and super conductor power transmission.

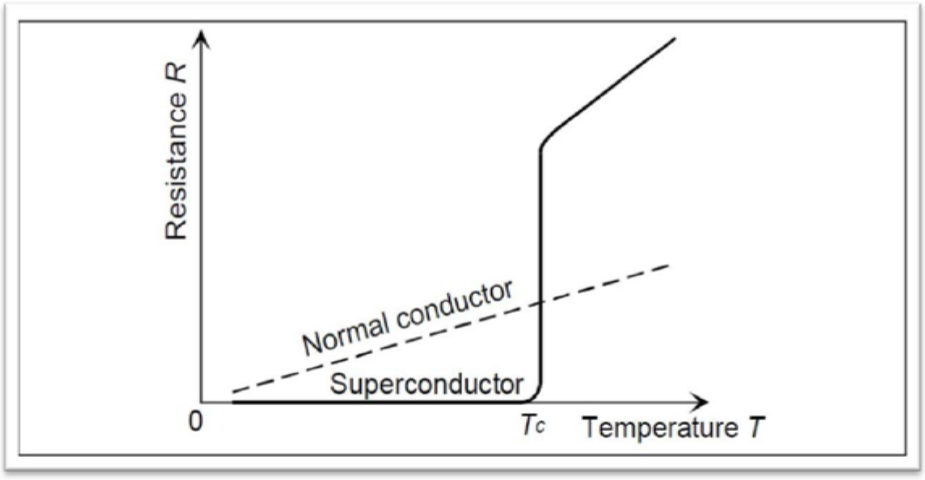


Figure 5: Resistance versus temperature [2]

There are three kinds of HTS lines, which can be used as shown in Table 2

Table 2: Possible HTS materials for use in the N3-X

	HTS	Definition	Operating temperature (˚K)
1.	BSCCO	Bismuth Strontium Calcium Copper Oxide	below 59
2.	YBCO	Yttrium Barium Copper Oxide	60 ~77
3.	MgB_2	Magnesium Diboride	30 ~ 39

From the operating temperatures, the need of a cryogenic system to maintain these low temperatures is evident [10]. The need of a cryogenic system to provide the thermal loads required for superconductor operation has led to the need for a two-phase heat exchanger. The ability to use a two-phase heat exchanger will substantially reduce the size of the required thermal management system. Unfortunately, there is not much known about how a two-phase heat exchanger performs – basically the phase change is a transient phenomenon therefore a transient heat exchanger.

Two Phase Flow Heat Exchangers

Selecting an appropriate heat exchanger that can manage these thermal loads and provide a suitable cooling rate will be addressed next. Plate fin heat exchangers, PHE will be used because of their compactness, lightweight, and high heat transfer rate which match aerospace industry needs [14] [15] [16] [17] [18]. The value of heat transfer coefficient of PHE heat exchangers is nine times the heat transfer coefficient in typical circular tubes heat exchangers for the same Reynolds number [15]. Enhancing heat exchange through the heat exchanger is essential. Namely, the input energy is conserved. [19] [20]. Figure 6 illustrates the basic components of a PHE.

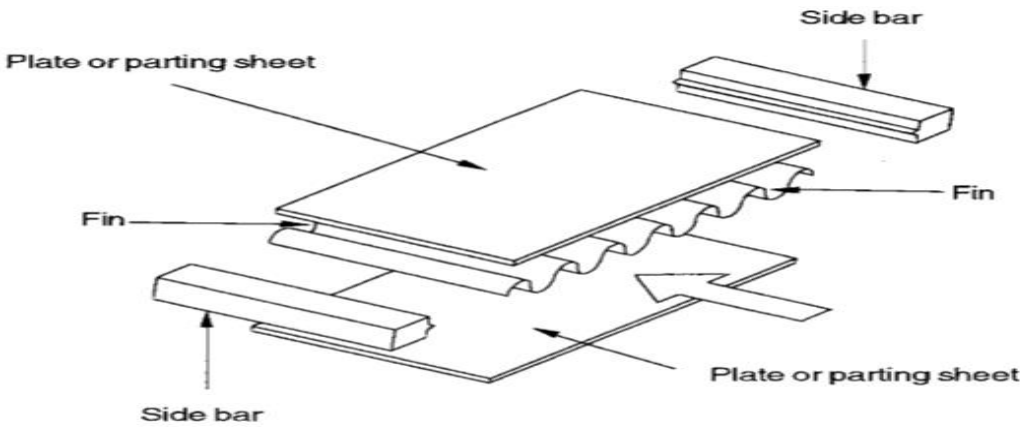


Figure 6: Basic components of a PHE [10]

There are several kinds of PHE heat exchangers as listed below [21]

1. Plain rectangular
2. Plain trapezoidal
3. Wavy
4. Serrated or offset strip fin
5. Louvered
6. Perforated

An offset strip- fin PHE will be used in this investigation. Figure 7 is given to illustrate the geometry parameters of an offset strip- fin PHE.

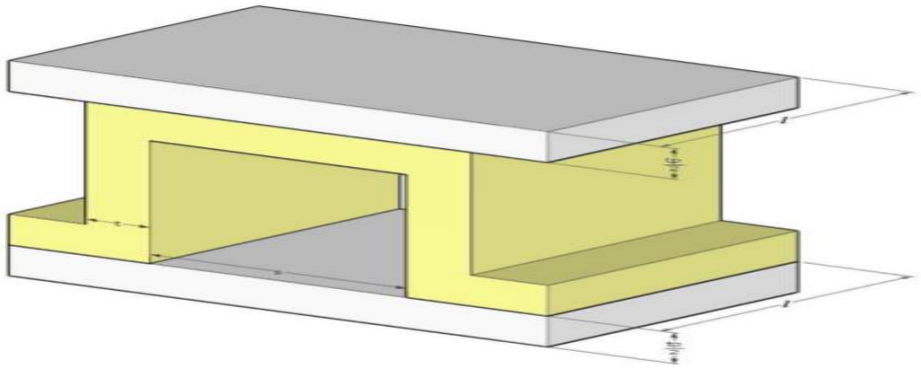


Figure 7: Offset strip- fin configuration [16]

Hence, the flow is 1D and counter flow. Figure 8 shows a typical offset strip-fin (OSF) configuration.

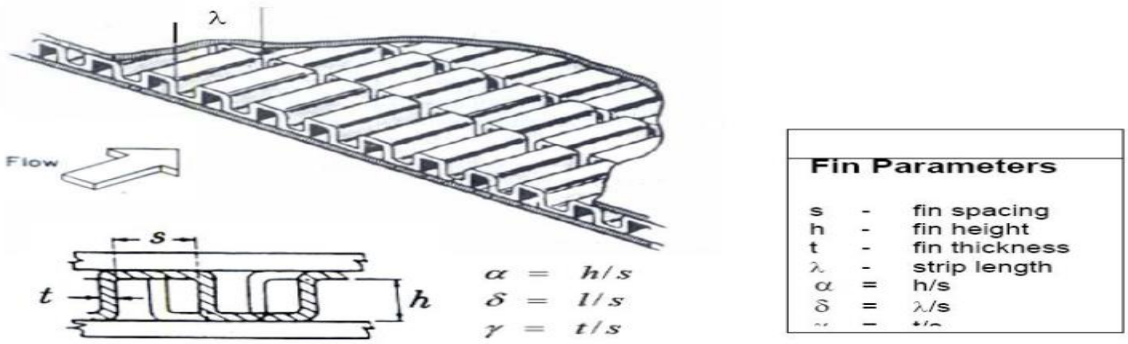


Figure 8: The regular configuration of OSF

Two different working fluids are used in this investigation, which are kerosene (jet fuel) and R134a. R134a is the hot fluid, while kerosene is cold fluid. Since R134a enters the heat exchanger as a superheated vapor, it will undergo a condensation process in the heat exchanger. This process occurs inside the R134a saturated liquid/- vapor dome, the refrigerant will be two- phases (liquid + vapor). Figure 9 shows R134a phases and the saturated dome.

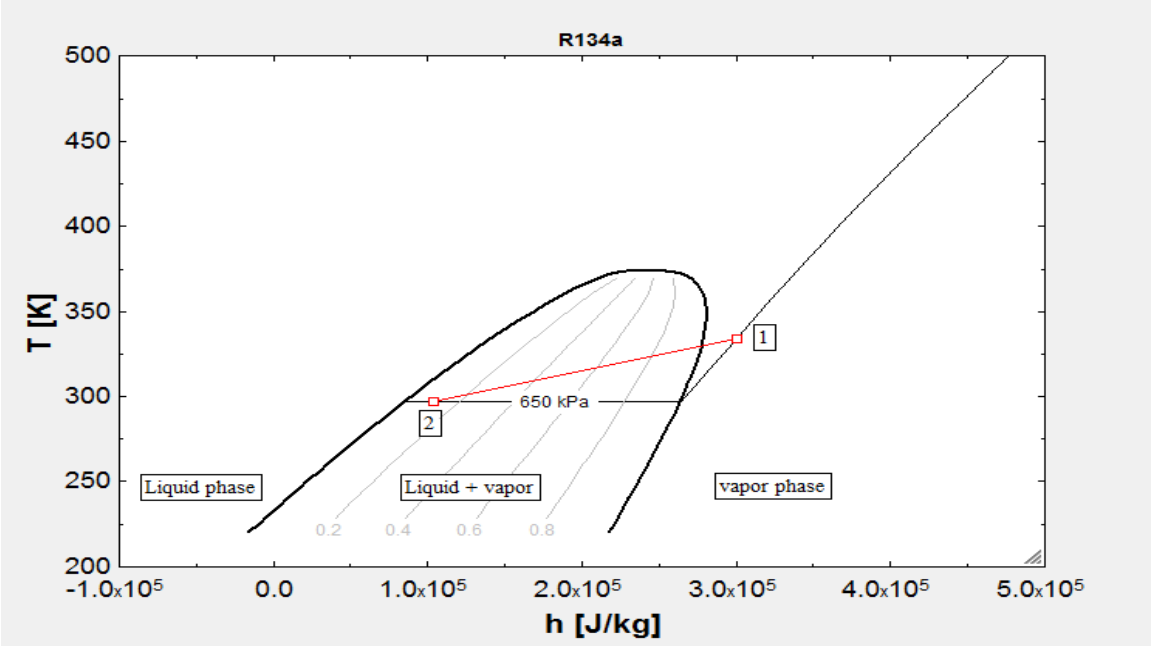


Figure 9: R134a phases and the saturated dome

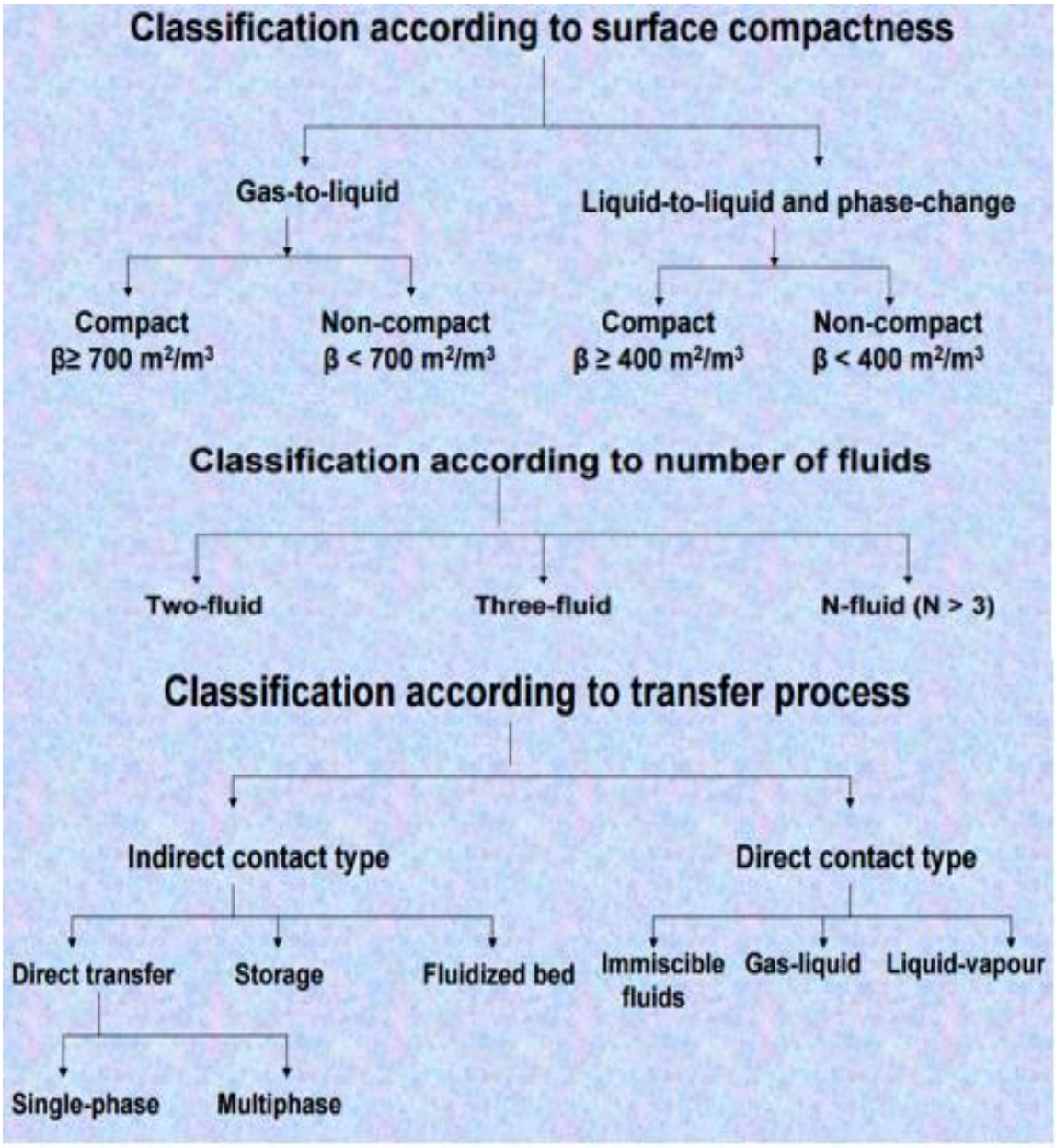
BACKGROUND

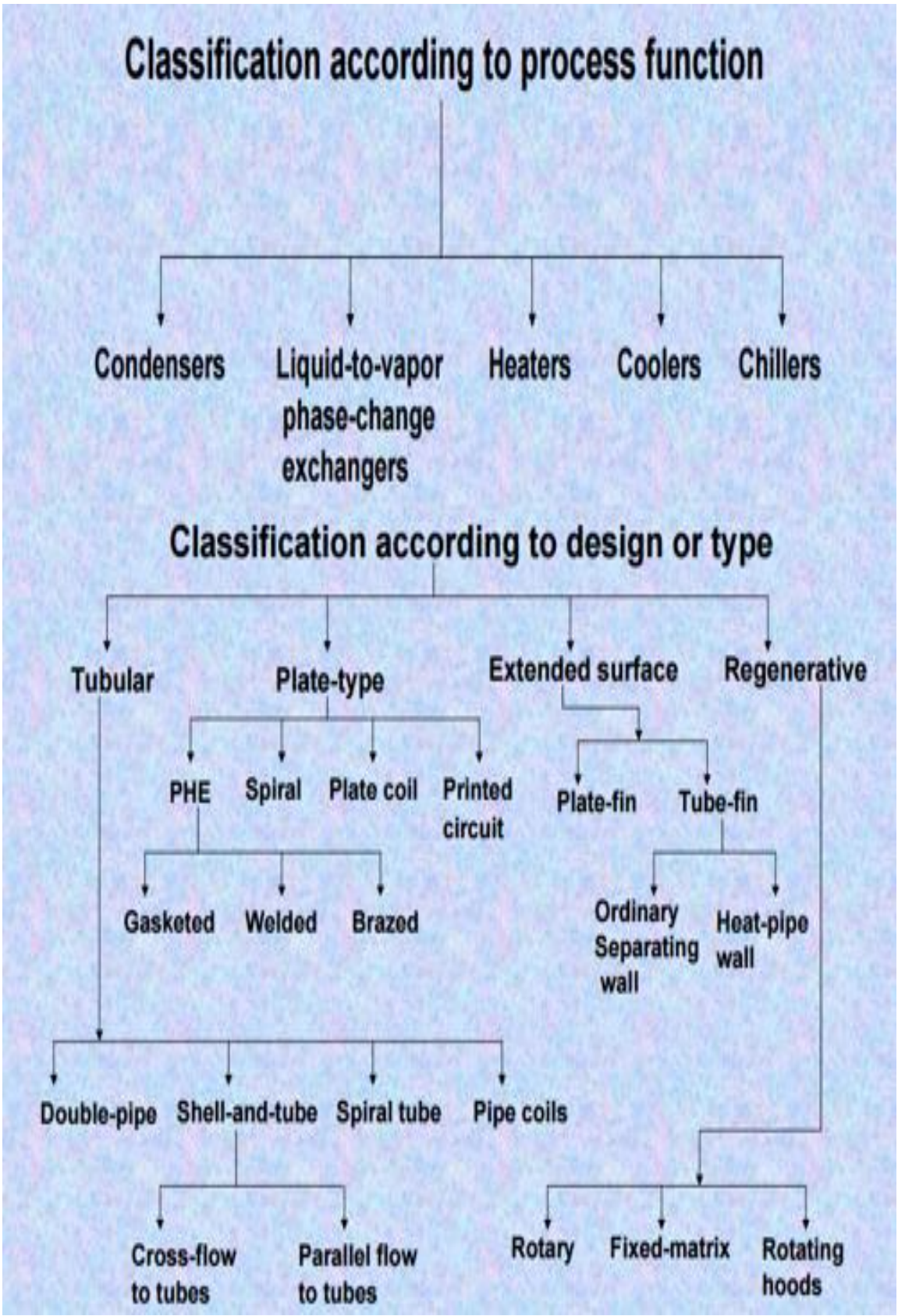
A heat exchanger is one of most wide spread pieces of equipment used in industry. It is used in oil refineries, food processing, and cooling/ refrigeration applications [15] [16]. The geometry of heat exchangers varies depending on the use and types of fluids for the heat exchange. A high percentage of heat exchangers used in different industries work with a two- phase flow [22]. Therefore, providing an essential need to clearly understand phase change through heat exchangers [19].

The background of two- phase flow heat exchangers begins in very ancient eras when people first used cooking- vessels to prepare their food. In addition, Archimedes used two- phase flow heat exchangers in the steam- gun, which has been considered the first application of two phase flow heat exchangers in the military. Heron created a turning sphere, which was operated by steam. Figure 10 shows Heron's turning sphere invention.

Figure 10: Heron's turning sphere invention [22]

Ancient Egyptians also used two- phase flow heat exchangers in wine production. In the nineteenth century- industrial revolution, James Watt discovered the steam engine which opened many applications to use two- phase flow in so many fields [22]. Heat exchangers can be classified according to following criteria shown in Figure 11.




Classification according to process function
Condensers
Liquid-to-vapor phase-change exchangers
Heaters
Coolers
Chillers
Classification according to design or type
Tubular
Plate-type
Extended surface
Regenerative
PHE
Spiral
Plate coil
Printed circuit
Plate-fin
Tube-fin
Gasketed
Welded
Brazed
Ordinary Separating wall
Heat-pipe wall
Double-pipe
Shell-and-tube
Spiral tube
Pipe coils
Cross-flow to tubes
Parallel flow to tubes
Rotary
Fixed-matrix
Rotating hoods

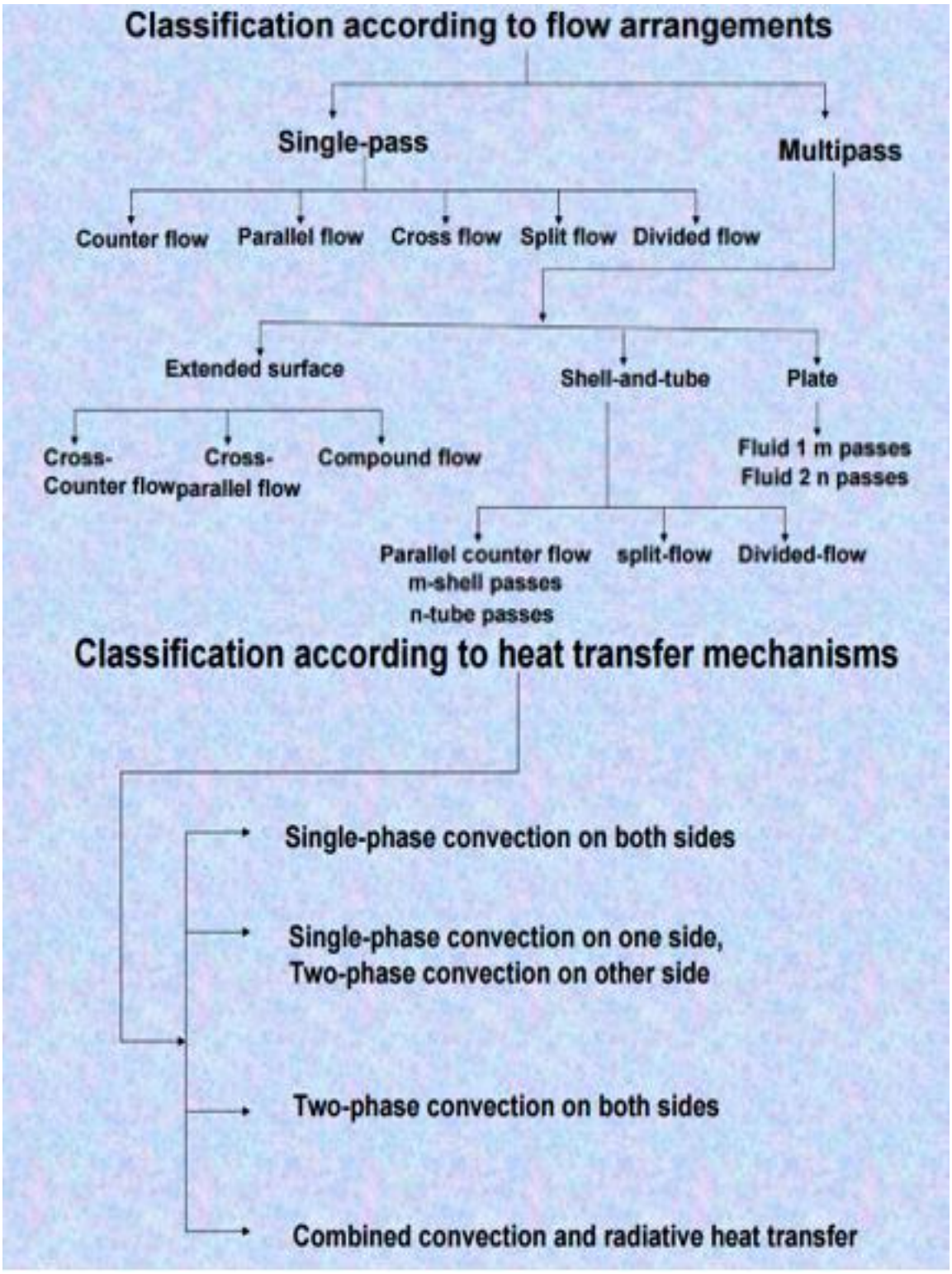


Figure 11: Heat exchangers classifications [23]

As previously discussed, for the aircraft industry a PHE heat exchanger is typically utilized to address the various application requirements, therefore a PHE heat

exchanger with a typical offset strip- fin (OSF) configuration is used for this research. As shown in Figure 9, the OSF causes the boundary layer to be recreated after every couple of strips. In addition, the OSF design causes the flow to be unsteady. Unsteadiness in the flow will expedite the transition to turbulent flow which enhances heat transfer rate [23].

It is obvious the importance of understanding two phase mode and analyzing its characteristics [19]. The first step requires coming up with a suitable correlation of the heat transfer coefficient. The initial attempts to correlate heat transfer coefficient were for flow in tubes, but for single phase flow only. Dittus and Boelter, Incropera and Dewitt, and Gnielinski arrived at the following equations to find Nusselt number used to calculate heat transfer coefficient. Their three equations respectively are:

$$\text{Nu} = 0.023\, Re^{0.8}\, Pr^{0.4} \tag{1}$$

$$\text{Nu} = 0.023\, Re^{0.8}\, Pr^{0.4} \left(\frac{\mu_m}{\mu_{wall}}\right)^{0.14} \tag{2}$$

$$\text{Nu} = \frac{\zeta/8\ (\text{Re} - 1000)\ \text{Pr}}{12.7\sqrt{\zeta/8}\left(Pr^{2/3}-1\right)+1.07} \tag{3}$$

$$\text{where } \zeta = (0.79\ln(Re) - 1.64)^{-2} \tag{4}$$

Various studies have been made to obtain suitable correlation of two- phase flow heat transfer coefficient in plate heat exchangers with Kandlikar proposing the best correlation (17% error) [24]. A fluid dependent parameter (correction factor [25]) is included in this equation in addition to the convective and boiling number. Table 3 lists values of the fluid dependent parameter for various types of fluids in copper and brass tubes.

Table 3: Fluid dependent parameter for various types of fluids in copper and brass tubes [25]

ref	Fluid	Fluid dependent parameter
1.	Water	1
2.	R-11	1.30
3.	R-12	1.50
4.	R-13B1	1.31
5.	R-22	2.20
6.	R-113	1.30
7.	R-114	1.24
8.	R-124	1.9
9.	R-134a	1.63
10.	R-152a	1.10

It is interesting to note that for stainless-steel , the fluid dependent parameter is 1 for all fluids [26].

MATHEMATICAL MODELING

A physics based model of a two-phase flow heat exchanger has been developed. The model was developed in the MATLAB/Simulink software environment. Various assumptions have been made in this model. The main characteristics of this model are as follows:

- One dimensional flow
- Indirect contact type (not mixing)
- Two fluids used; kerosene/ jet fuel (cold side) and R134a (hot side)
- Compact surface ($\beta \geq 700\ m^2/\ m^3$)
- Extended surface (plate fin) heat exchangers
- Single pass (counter flow)
- Single- phase convection for kerosene and two-phase convection for R134a
- Process function is as condensers

Figure 12 illustrates the geometry of the heat exchanger modeled.

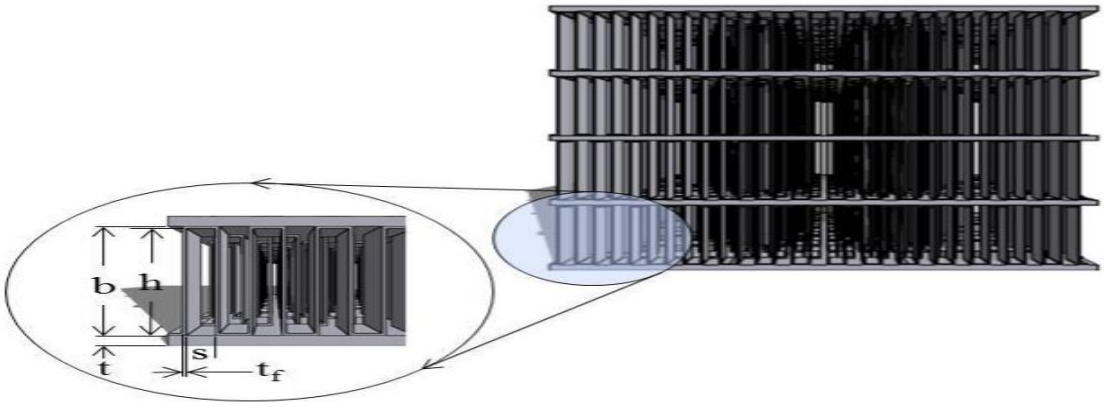


Figure 12: Offset strip fin heat exchanger geometry [27]

Table 4 lists the mass flow rate and temperature of the two fluids used in this research.

Table 4: Mass flow rate and temperature for both sides

ref	side	$\dot{m}$ (kg/sec)	Temperature (k)
1.	Cold	1	280
2.	Hot	1.5	360

The model is developed such that the heat exchanger could be made from one of the three materials; Stainless Steel 316, Copper, or Aluminum. Table 5 lists the material characteristics for each of these metals.

Table 5: C_p, rho, and k for SS-316, Cu, and Al

ref	Material	Cp (J/kg- k)	ρ (kg/m^3)	k (w/m- k)
1.	Stainless steel- 316	502	8027	16.26
2.	Aluminum	896	2707	220
3.	Copper	380	8954	386

A basic diagram of the heat exchanger model is shown in Figure 13. The model uses one node to model the flow from inlet to the exit of each fluid.

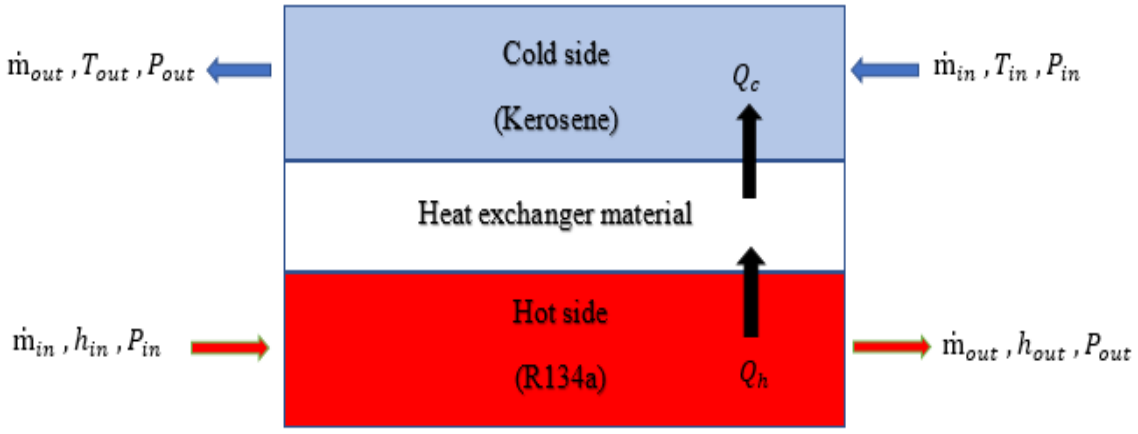


Figure 13: Diagram of the heat exchanger model

The assumptions made in this model are:

- Adiabatic system (heat in and heat out are zero)
- Kinetic and potential energy are negligible
- Linear interpolation between pressure lines inside the dome

Primarily, this model consists of four subsystems. The subsystems are:

- Cold balance
- Hot balance
- Heat exchanger balance
- Exergy analysis

To investigate these subsystems, each subsystem will be studied discretely to understand the input and output parameters.

Cold Subsystem Balance

This subsystem simply takes care of the kerosene/ jet fuel analysis. In general, it is made up of a subsystem to calculate the fanning friction factor, a subsystem to compute the pressure drop, and a subsystem to calculate the energy balance analysis. For this purpose, Manglik and Bergles correlations are used with the following equations [28].

$$f = 9.6243 * Re^{-0.7422} * \alpha^{-0.1858} * \delta^{0.3053} * \gamma^{-0.2859} * (1 + 7.669 * 10^{-8} * Re^{4.429} * \alpha^{0.92} * \delta^{3.787} * \gamma^{0.238})^{0.1} \quad (5)$$

$$D_h = \frac{4shl}{2(sl+hl+th)+ts} \quad (6)$$

$$\mathrm{Re} = \frac{\dot{m} * D_h}{\mu * A_c} \quad (7)$$

$$\Delta P = \frac{f}{2} * \frac{L}{D_h} * \frac{1}{\rho} * \left(\frac{\dot{m}}{A_c}\right)^2 \quad (8)$$

Applying an energy balance for the kerosene will allow the outlet temperature to be calculated, utilizing the following equations:

$$\frac{dE}{dt} = \dot{Q} + \dot{m}*(h_{in} - h_{out}) \quad (9)$$

$$\frac{dE}{dt} = m * c_v * \frac{dT}{dt} \quad (10)$$

The left-hand side of Equation 10 is the time rate of change of energy which equals the time rate of change in internal energy only because the kinetic and potential energy are negligible.

Recalling c_p and c_v for incompressible fluids are identical, so using either c_p or c_v in (10) wouldn't affect the result. By integrating the $\frac{dT}{dt}$ term, the outlet kerosene temperature is determined. Equation (11) is used to calculate the rate of entropy change for kerosene.

$$\frac{dS}{dt} = \frac{m * cp}{T} * \frac{dT}{dt} \quad (11)$$

Hot Subsystem Balance

The hot fluid, R134a, model is now presented. The following parameters are calculated using the same equations as the cold subsystem: the fanning friction factor, the hydraulic diameter, the pressure drop, and the energy balance models use the same equations as the cold subsystem balance. However, the outcome of the energy balance is the internal energy not the temperature. Using the thermodynamic definition of enthalpy (Eqn. 12), the model formulated an enthalpy- based model.

$$h = u + pv \quad (12)$$

Many different pressure lines have been selected to approximate the thermodynamic properties. Hence, 66.19, 132.8, 243.5, 414.9, and 665.8 kPa have been used to get specific volume, density, enthalpy, enthalpy of evaporation, internal energy,

temperature, quality, specific heat, dynamic viscosity, thermal conductivity, and entropy for R134a. Appendix A contains a list of the lookup tables used in the model. The lookup tables have been determined using EES. Various pressure lines have been taken to diminish errors. But, errors still exist because the assumption of a linear interpolation is not always valid. In addition, some thermodynamic properties cannot be determined directly inside the dome. Dynamic viscosity, specific heat, and thermal conductivity are examples of these properties. To address this problem, the following equations are used [29].

$$\mu_{TP} = \mu_l + x\,(\mu_G - \mu_l) \tag{13}$$

$$cp_{TP} = cp_l + x\,(cp_G - cp_l) \tag{14}$$

$$k_{TP} = k_l + x\,(k_G - k_l) \tag{15}$$

Utilizing Equations (13) - (15), the respective properties are easily calculated.

Heat Exchanger Subsystem

From the previous subsystems, temperature distributions for kerosene and R134a are known to use in the heat exchanger calculations. This subsystem will be divided into two parts.

Part 1: Kerosene heat transfer coefficient

Part 2: R134a heat transfer coefficient or two-phase flow heat exchanger

Initially, the Prandtl and Nusselt number, heat transfer equations are used to implement these calculations.

$$\mathrm{Pr} = \frac{\mu\ cp}{k} \tag{16}$$

$$h = \frac{Nu\ k}{D_h} \tag{17}$$

For part 1, one would use (Eqn. 3) to compute Nusselt number using the Prandtl number. Then, utilizing (Eqn. 17) the heat transfer coefficient is calculated. Note, the dynamic

viscosity varies according to temperature. This variation is handled via a lookup table (Appendix A). Next, heat transfer rate is calculated by

$$Q_c = h_c\, A_s\, (T_{HX} - T_c) \tag{18}$$

For part 2, since R134a enters the heat exchanger as superheated vapor, there are three regions R134a could reach because of condensation. In other words, the amount of heat transferred from R134a to kerosene will determine the phase of refrigerant at exit. Therefore, there are three states R134a could exit the heat exchanger.

- Vapor
- Saturated mixture
- Liquid

In such cases, three simulation models have been created to address this issue. If R134a comes out in one phase (vapor or liquid), then (Eqn. 3) is used in the model. However, for the saturated mixture (liquid +vapor), a two- phase heat transfer coefficient correlation proposed by Kandlikar is used. The basis for that comparison depends on quality, which specifies the state of the refrigerant. This is modeled using a Simulink "State flow logic". The advantage of state flow use is making logic, much cleaner, simpler, and more maintainable. Hence, transition between states would be easier by state flow logic. Furthermore, ability to see the transition is available for any values of quality. Hence, the condition placed is that if $x \geq 1$, the Nusselt number at vapor phase would be implemented. While if $0 < x < 1$, the two- phase heat transfer coefficient would be computed. Otherwise, $x = 0$, then, the Nusselt number at liquid phase would be calculated. The next equations would be used to execute the model.

$$h_{TP} = [(1.183744\ Co^{-0.3} + 225.5474\ Bo^{2.8}\ F_{fl})]\ (1-x)^{0.003}\ h_l \quad (19)$$

$$\text{Co} = \left[\frac{rho_G}{rho_l}\right]^{0.5} * \left[\frac{1-x}{x}\right]^{0.8} \quad (20)$$

$$\text{Bo} = \frac{q''}{\text{G} * h_{lG}} \quad (21)$$

Once h has been calculated, finding the heat transfer rate is straightforward. Equation (22) is used.

$$Q_h = h\ A_s\ (T_h - T_{HX}) \quad (22)$$

Now, the magnitude of Q for both kerosene and R134a is known. By applying an energy balance on the heat exchanger material, the heat exchanger temperature, T_{HX}, is calculated by integrating (Eqn. 24).

$$Q_{HX} = Q_{refrigerant} + Q_{kerosene} \quad (23)$$

$$\frac{dT}{dt} = \frac{Q_{HX}}{m\ cp} \quad (24)$$

Exergy Analysis Subsystem

An exergy model of the two-phase heat exchanger is developed for future research where the heat exchanger model is incorporated with other component models to model a more complex system like an aircraft. An exergy analysis is consistent with the 2nd law of thermodynamics. The model is consistent with the previous heat exchanger model based on the 1st law of thermodynamics. The two-phase heat exchanger will be divided into three parts, the hot side, cold side, and heat exchanger materials.

The appropriate exergy equations as follows.

$$m_c\ \frac{cp_c}{T_c}\ \frac{dT}{dt} = \frac{-\dot{Q}_c}{T_{HX}} + \dot{m}\ (s_{in,c} - s_{out,c}) + \dot{S}_{irr,c} \quad (25)$$

$$m_h \frac{cp_h}{T_h} \frac{dT}{dt} = \frac{-\dot{Q}_h}{T_{HX}} + \dot{m}\,(s_{in,h} - s_{out,h}) + \dot{S}_{irr,h} \tag{26}$$

$$m_{HX} \frac{cp_{HX}}{T_{HX}} \frac{dT}{dt} = \frac{\dot{Q}_c + \dot{Q}_h}{T_{HX}} + \dot{S}_{irr,HX} \tag{27}$$

$$\dot{X} = T_o \left[\dot{S}_{irr,HX} + \dot{S}_{irr,h} + \dot{S}_{irr,c}\right] \tag{28}$$

$$\dot{X} = T_o \left[m_{HX} \frac{cp_{HX}}{T_{HX}} \frac{dT}{dt} + m_h \frac{cp_h}{T_h} \frac{dT}{dt} - \dot{m}\,(s_{in,h} - s_{out,h}) + m_c \frac{cp_c}{T_c} \frac{dT}{dt} - \dot{m}\,(s_{in,c} - s_{out,c})\right] \tag{29}$$

For the R134a analysis, a lookup table (appendix A) is used to find the entropy. Combining all of the various subsystem models together results in the complete two-phase heat exchanger model shown in Figure 14.

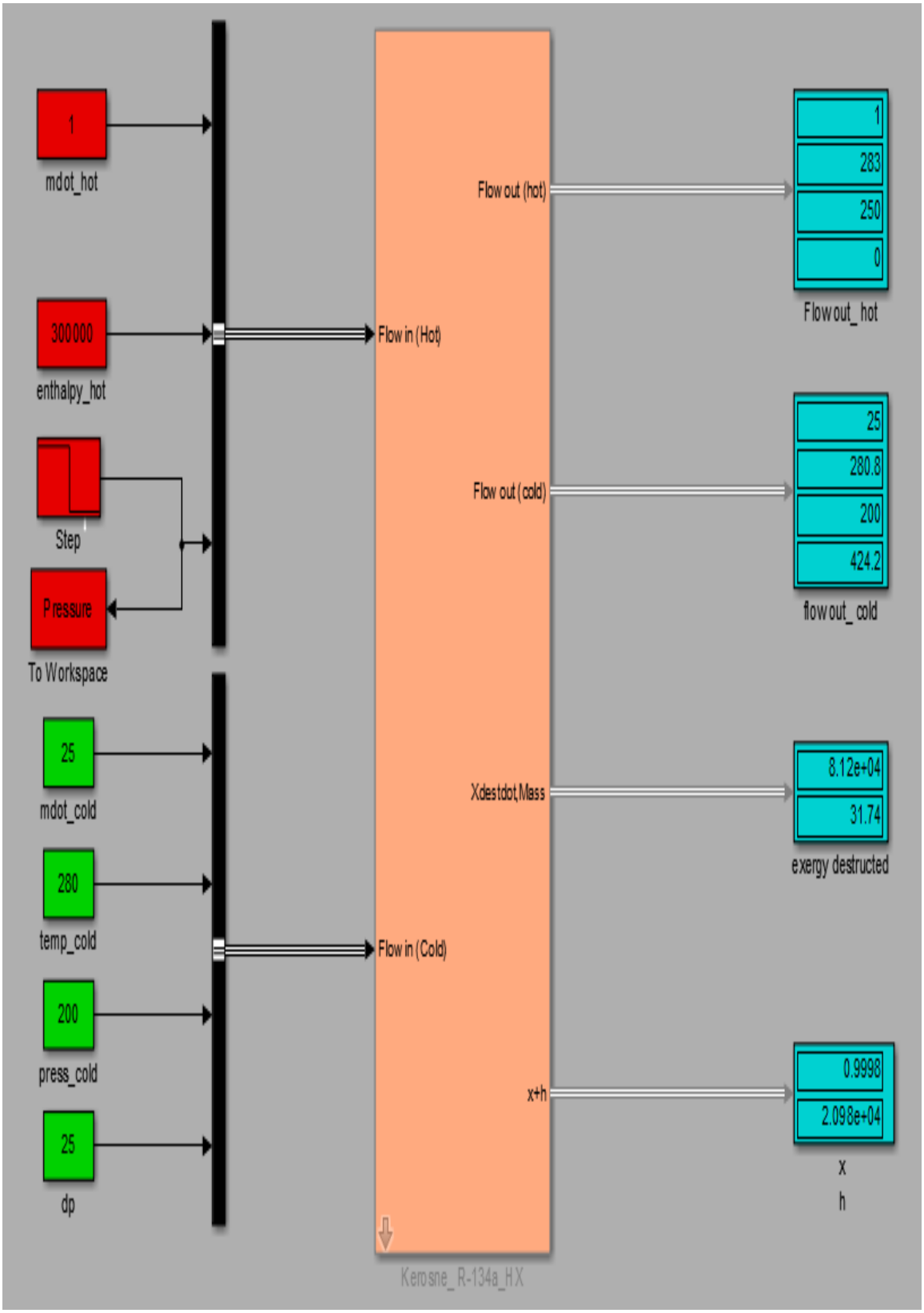


Figure 14: The whole heat exchanger simulation model

RESULTS

Transient analysis of a two-phase heat exchanger is critical to managing the thermal loads on next generation aircraft especially the NASA N3-X concept vehicle. A step change is typically a limiting transient input. Therefore, a step change in both pressure and enthalpy will be initially investigated to define the limits of the model. The overall duration of the simulation will be 6 seconds with a time step of 10^{-6} seconds. A simulation time step of 10^{-4} seconds is stable but to ensure model stability a time step two orders of magnitude lower was used for all of the results presented. The pressure step change is from 650 to 555 kPa and the enthalpy step change is from 350 to 300 kJ/kg.

Pressure Perturbation

A step change in the heat exchanger pressure for the cold side is presented. Figure 15 shows the pressure step change for R134a from 650 to 555 kPa. The enthalpy of the R134a is shown in Figure 16. The enthalpy of the R134a is constant as the pressure is changed.

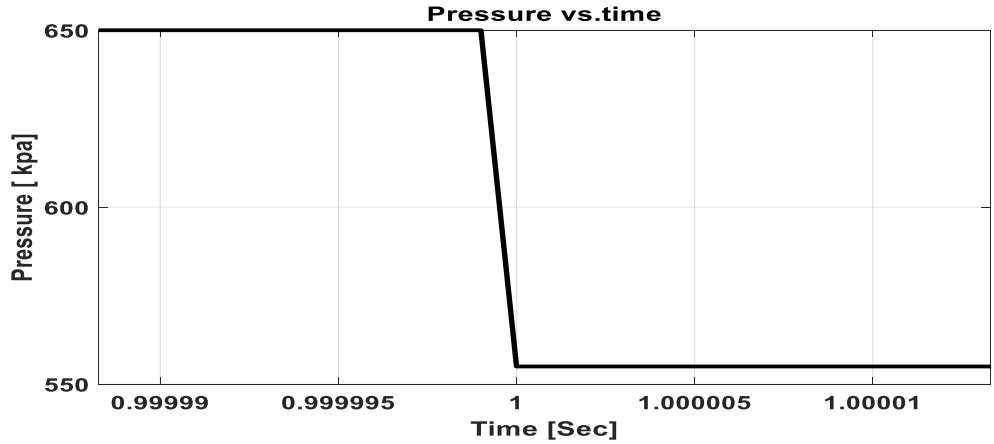


Figure 15: Hot side (R134a) pressure step change

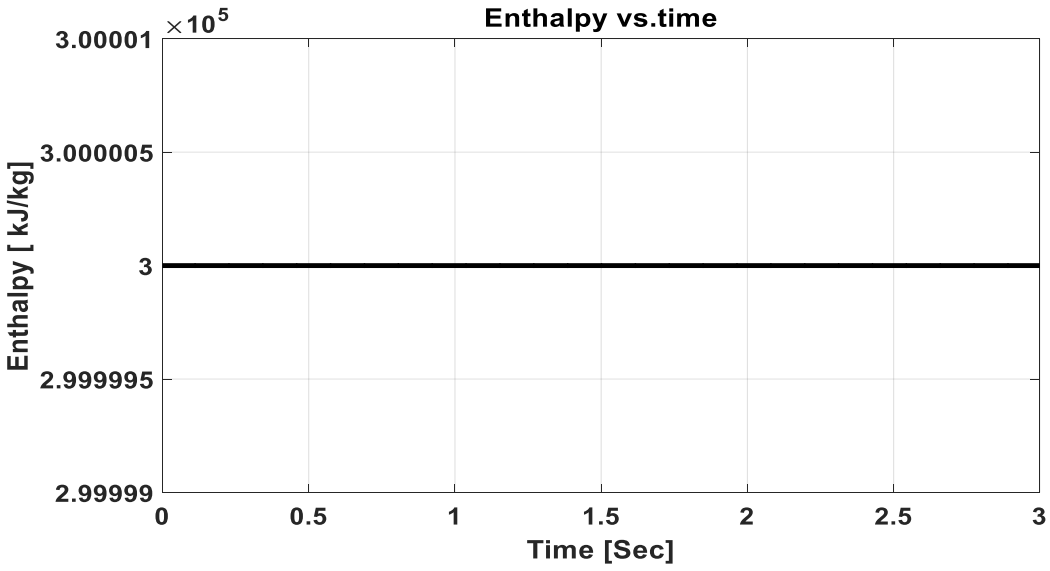


Figure 16: Hot side (R134a) enthalpy with pressure step change

The kerosene temperature out of the heat exchanger is 282.3 K before the step change in R134a pressure, while $T_{c,out}$ is 281.8 K after the pressure step changes. Therefore, dropping the refrigerant pressure results in a decrease in kerosene temperature out of the heat exchanger of 0.5 °K. The reduction in operating pressure of the condensing heat exchanger reduces the saturation temperature of the refrigerant, which in turn reduces the amount of heat rejected from the refrigerant to kerosene through the heat exchanger.

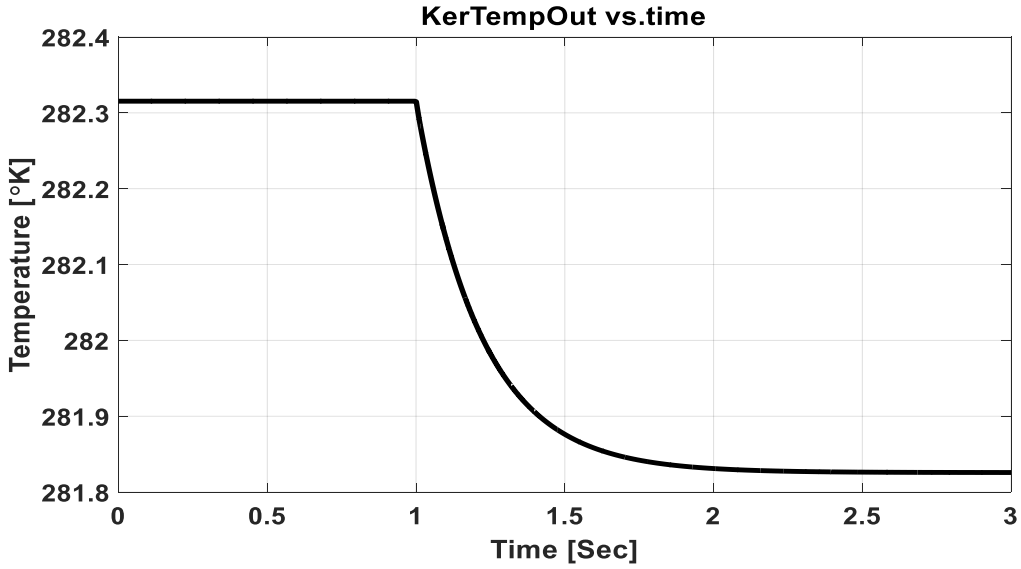


Figure 17: Cold flow (kerosene) temperature out with pressure step change

The effect of the step change in pressure of the R134a on the R134a temperature is shown in Figure 18. The temperature out decreases due to the change in saturation temperature caused by the drop in pressure. As expected, a step decrease in the pressure of the R134a will affect the R134a temperature ($T_{h,out}$) directly. In the steady- state regions, when the pressure is 650 kPa, $T_{h,out}$ is 297 K, while at 555 kPa, $T_{h,out}$ is 292 K, which is due to the change in saturation pressure and temperature. The transient response predicted by the model between the two steady state values is of significant interest. Note: the temperature is calculated using pressure and internal energy through a lookup table. The model predicts the transient change to occur over 0.000001 seconds as result of the sudden change in pressure. This is due to the temperature being a non-dynamic state in the model. With respect to the conservation of energy, the internal energy is in a dynamic state in the model. Therefore, the transient response is calculated for temperature is a result of the step change in pressure. The model is capturing some trends in the transient region of the two- phase fluid. The initial response of over 0.1 seconds is driven by the internal energy of the mass in the control volume. The slower response is due the thermal equilibrium of the heat exchanger temperatures. Will the refrigerant temperature actually change to 292 K this quickly as shown? If not, how much? Experimentation and further analysis at the molecular level is needed to determine non-equilibrium physics of the fluid. Figure 19 presents the response of the heat exchanger material temperature to the step change. The temperature of the heat exchanger material is a dynamic state and the step change results in an appropriate first order response.

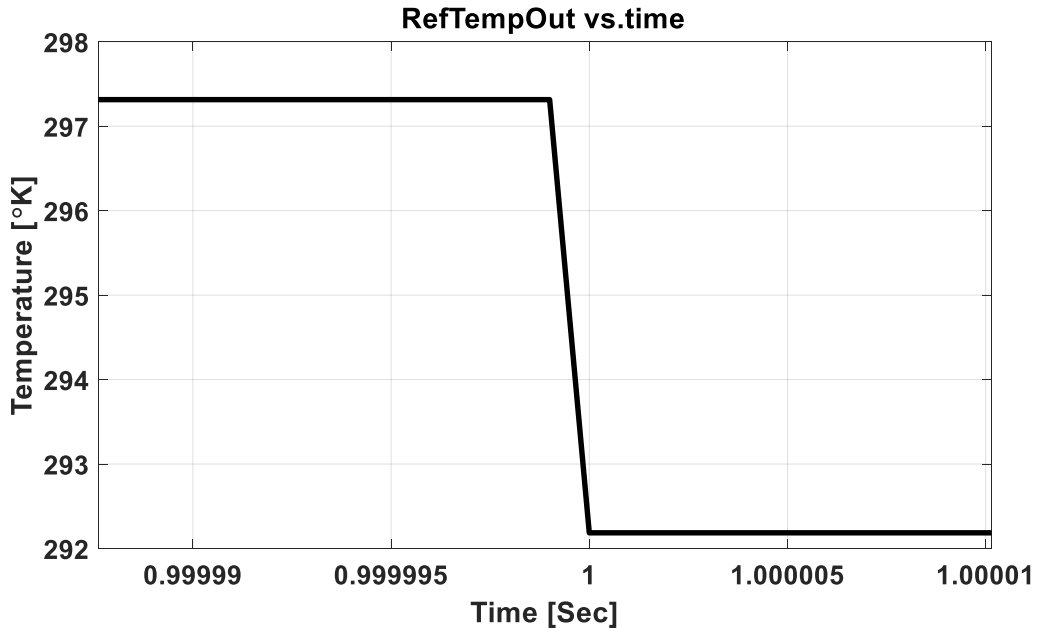


Figure 18: Hot flow (R 134a) temperature out with pressure step change

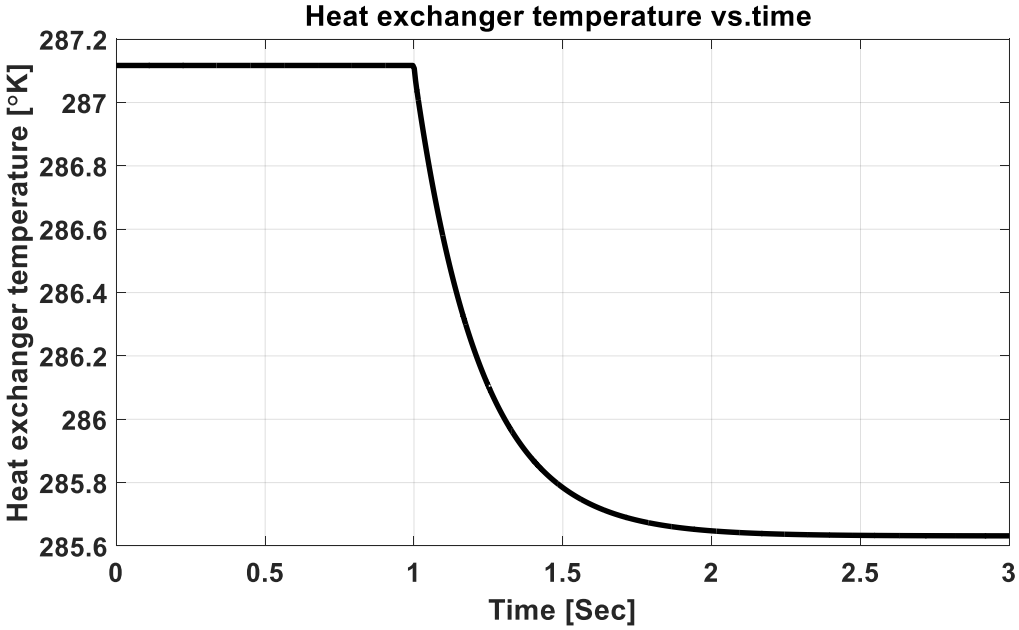


Figure 19: Heat exchanger material temperature with a pressure change

Figure 20 shows the heat transfer coefficient of kerosene, which only changes slightly due to the fluid properties changing with temperature.

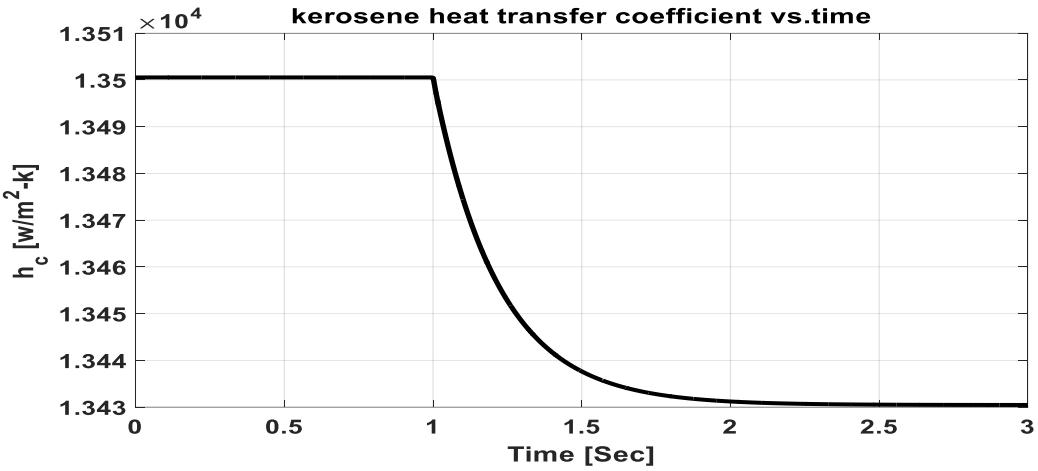


Figure 20: Cold flow (kerosene) heat transfer coefficient, h_c, with pressure step change

On the contrary, the heat transfer coefficient of the two- phase refrigerant, h_{TP} changes significantly with the pressure drop, Figure 21. Table 6 provides the steady- state values, which are nearly twice. This reinforces the idea of rapid pressure change approach for achieving a rapid thermal response to a transient heat load.

Table 6: Steady state hot side heat transfer coefficient (h_{TP}) with pressure step change

Ref	Pressure (kPa)	$\boldsymbol{h_{TP}}$ **(w/**$\boldsymbol{m^2 - k}$**)**
1.	650	3140
2.	555	4360

Figure 22 highlights the dependency of the quality on the pressure (as the pressure is decreased because the step change; the quality quickly increases). Hence, the quality is 0.6241 at a pressure of 650 kPa, while at a pressure of 555 kPa, the quality is 0.8023. As the pressure drops, R134a quality increases for the same value of internal energy. The quality does not change instantly with the pressure, but starts to increase as the internal energy of the fluid increases. The initial response takes approximately 0.01 seconds. The two- phase flow heat transfer coefficient is influenced significantly by the value of the quality, therefore, the heat transfer coefficient (h_{TP}) will increase even if temperature

decreases. Basically, h_{TP} is a function of temperature and quality. The quality has a slight overshoot. The internal energy is not in equilibrium and it increases as the refrigerant is evaporating instead of condensing during the transient region. The quality reaches a steady-state value as the temperature of the heat exchanger settles to 285.6 K. This takes approximately 0.2 seconds.

Figure 23 shows how much heat is transferred from the hot side (R134a) through the heat exchanger material into the cold side (Kerosene). The cold side heat transfer, Q_c, decreases because h_c decreases and the temperature difference between the two flows decreases.

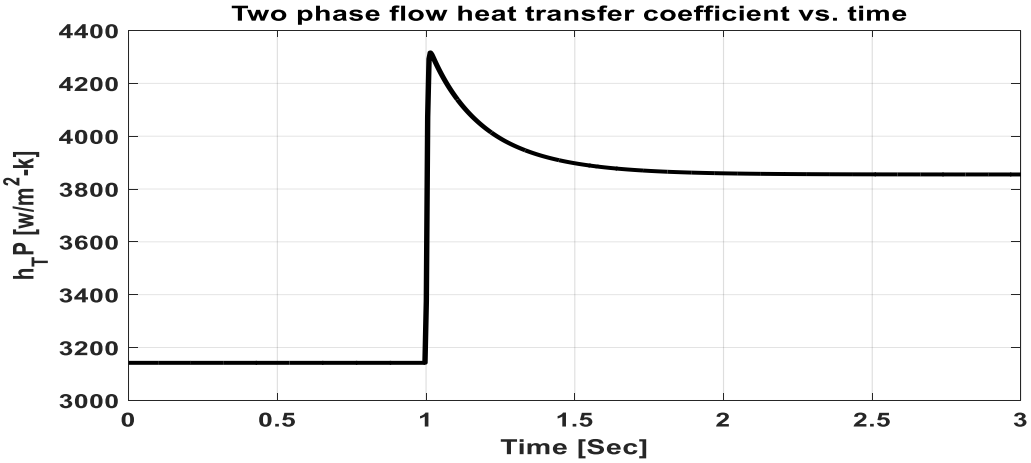


Figure 21: Hot side (R134a) heat transfer coefficient (h_{TP}) with pressure step change

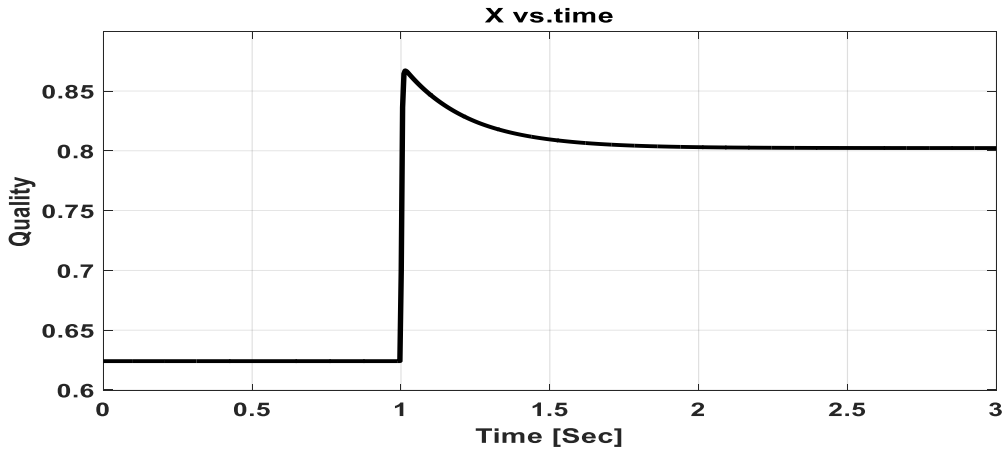


Figure 22: Hot side (R134a) quality with pressure step change

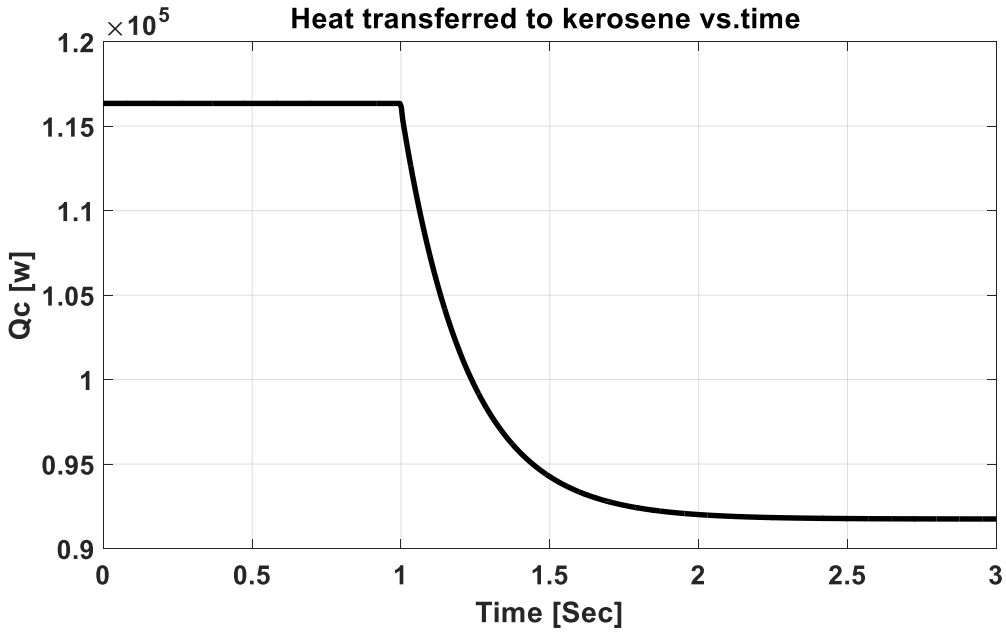


Figure 23: Heat transferred to the cold fluid (Kerosene) with pressure step change

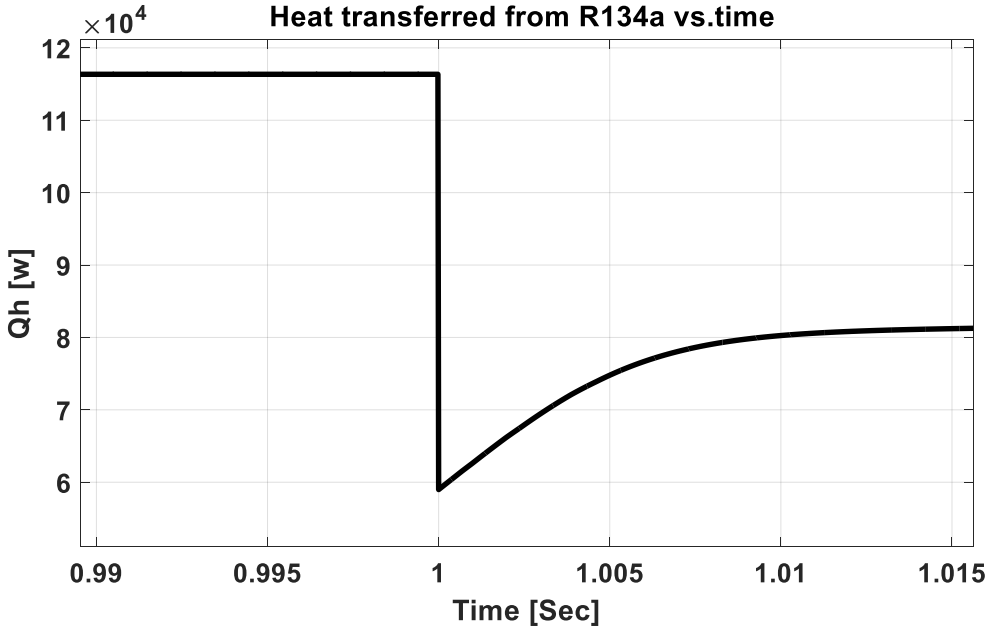


Figure 24: Heat transferred from hot side (R134a) with pressure step change

The heat transfer on the hot side (R134a) increases initially as the quality increases resulting in evaporation instead of condensation, but decreases with the reduction of quality as the heat exchanger becomes in thermal equilibrium. Once equilibrium in the

quality is reached, heat leaves the refrigerant resulting in condensing of the refrigerant once again. The steady state values of Q_h for the respective pressures of 650 and 555 kPa are presented in Table 7. Even though the heat transfer coefficient increases for the refrigerant the overall heat transfer decreases. This is because the temperature difference between the heat exchanger and refrigerant decreases from 10.2 K to 6.6 K. Table 7 demonstrates Q_h with respect to pressure 650 kPa and 555 kPa.

Table 7: Steady state Q_h

Ref	**Pressure (kPa)**	**Q_h (w)**
1.	650	116000
2.	250	91700

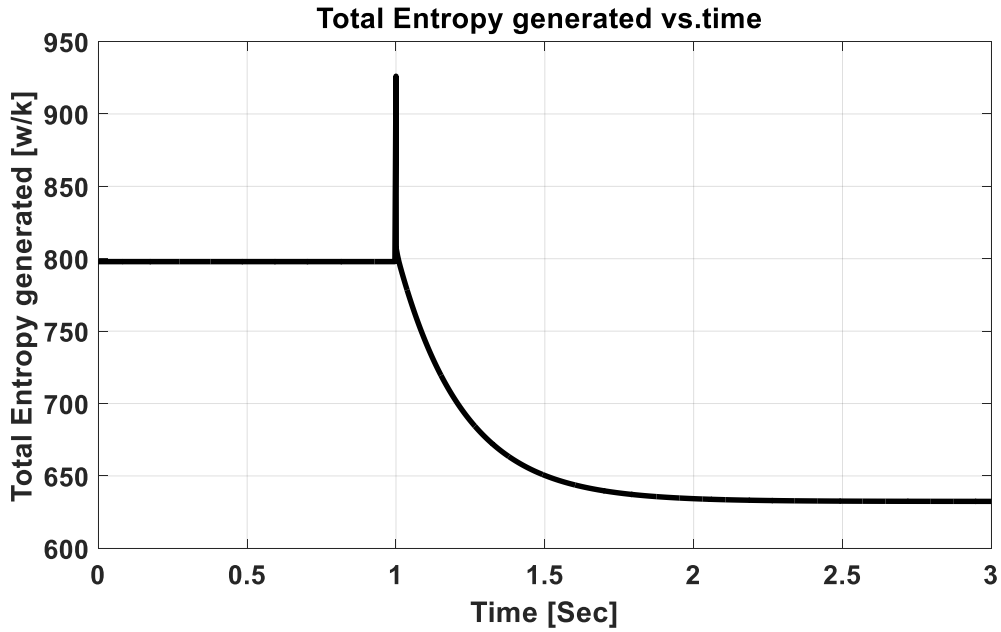


Figure 25:Total entropy generated with pressure step change

The steady state regions in Figure 25 show that entropy generated equals to 798 (w/k) at pressure 650 kPa, but at pressure of 555 kPa, $\dot{S}_{gen}$ = 632 (w/k). Figure 26 presents the enthalpy out for R134a. R134a enthalpy out at 650 kPa is 184 kJ/kg. while at 555 kPa it is 208 kJ/kg.

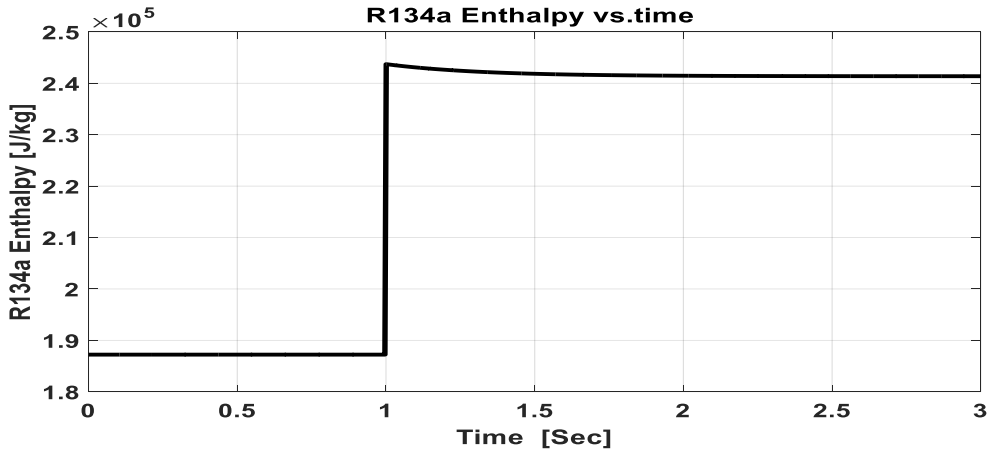


Figure 26: Hot side (R134a) enthalpy out with pressure step change

The step change in pressure results in a sudden change in enthalpy of the fluid. The sudden step in enthalpy follows the 0.01 second response in quality. Therefore, there are two time scales in the quality and enthalpy results. The time constants are from the equilibrium in the fluid and the heat exchanger temperatures.

Enthalpy Perturbation

The two-phase heat exchanger model will be used to analyze a step change in enthalpy. The hot side enthalpy for the R134a is reduced from 350 kJ/kg to 300 kJ/kg by a step change, Figure 27. This change in enthalpy does not result in any change of the hot side pressure, Figure 28. The effect of the enthalpy step change on the cold side (Kerosene) temperature out is shown in Figure 29. It is clear from this figure that the impact of an enthalpy step change is similar to the pressure step change effect. Hence, both cause a decrease in the cold side (Kerosene) temperature out. The steady state cold side temperatures are given in Table 8.

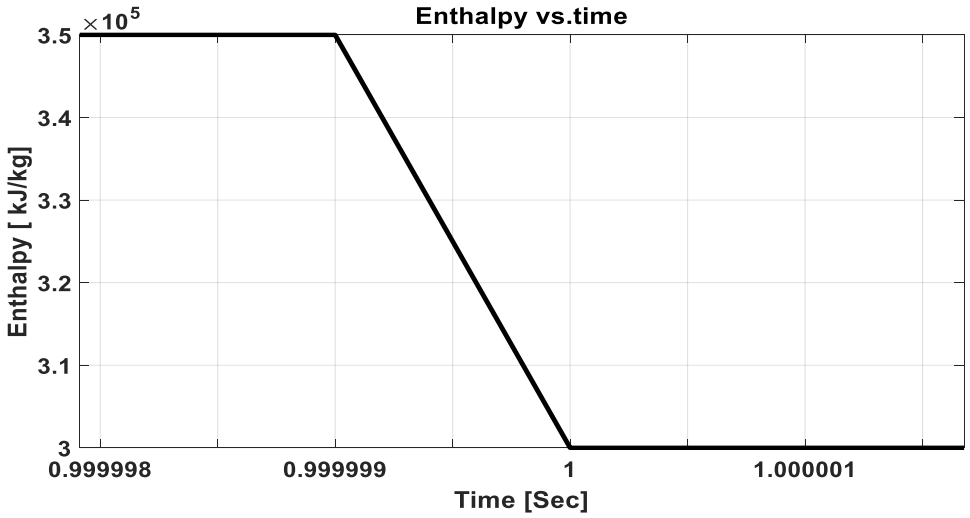


Figure 27: Hot side (R134a) enthalpy step change

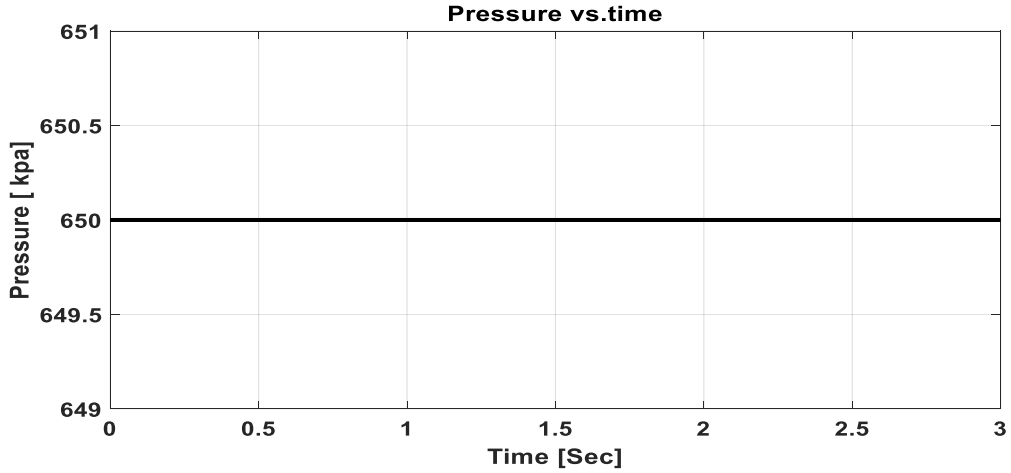


Figure 28: Hot side (R134a) pressure with enthalpy step change

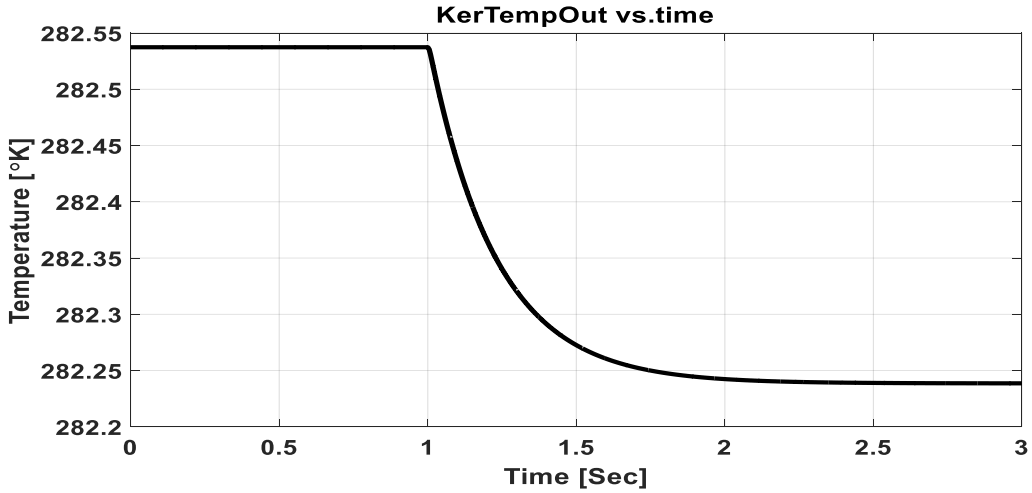


Figure 29: Cold flow (Kerosene) temperature out with enthalpy step change

Table 8: Steady state $T_{c,\,out}$

ref	Enthalpy (kJ/kg)	$T_{c,out}$ (k)
1.	350	282.5
2.	300	282.2

Since the pressure has not changed, the hot side (R134a) temperature out will remain at the saturated temperature as shown in Figure 30 as long as the quality is less than 1.0. This means there will be no opportunity for any error in the hot side (R134a) temperature out calculations because pressure is constant and the internal energy is dynamically calculated with a step change in enthalpy.

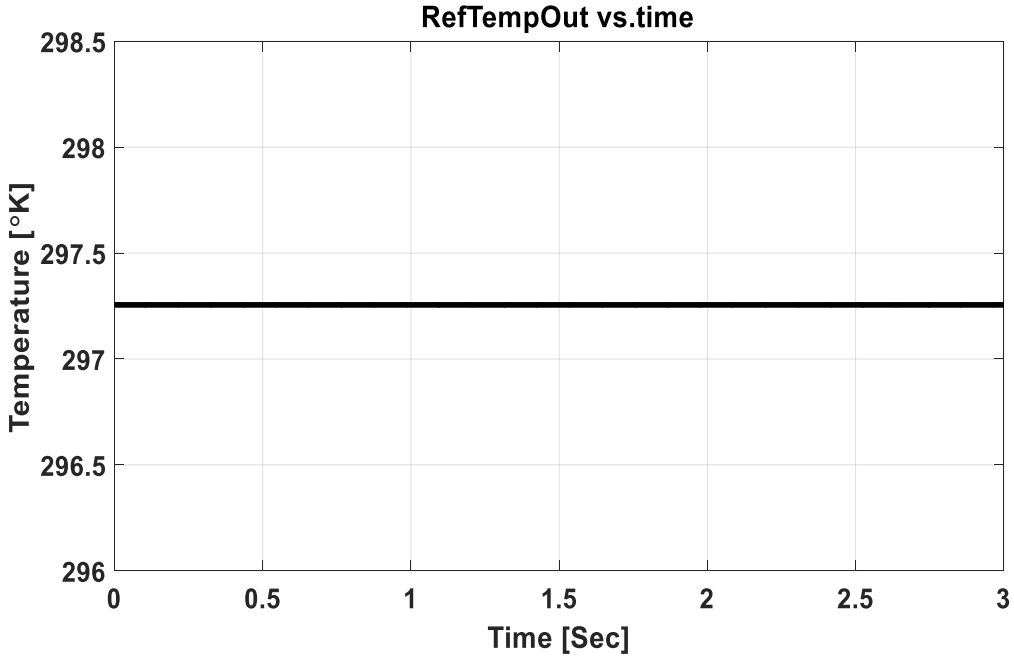


Figure 30: Hot side (R134a) temperature out with enthalpy step change

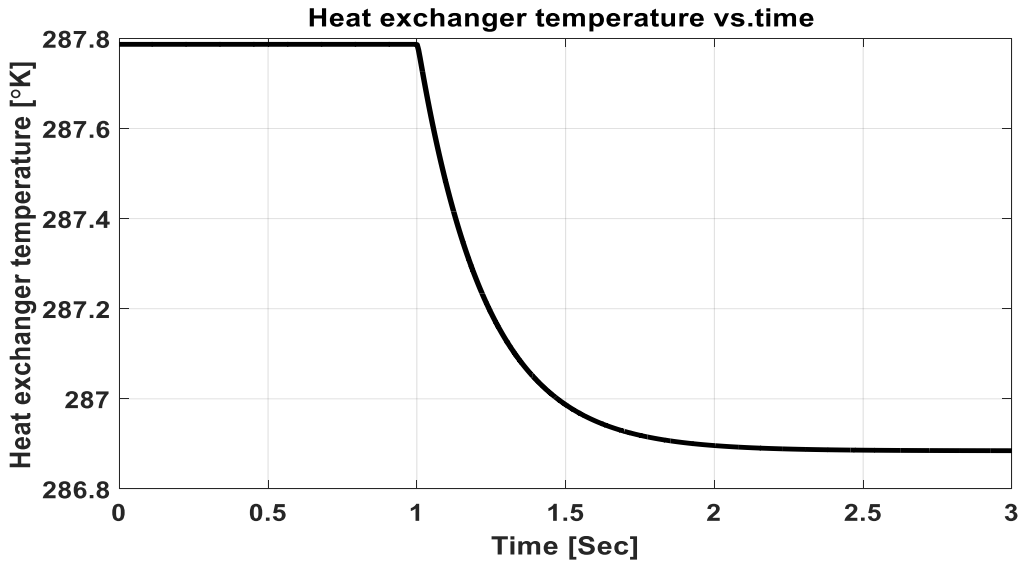


Figure 31: Heat exchanger material temperature with enthalpy step change

The enthalpy step change results in a slight change in the heat exchanger material temperature as shown in Figure 31. The heat exchanger material temperature shows a first order response for the enthalpy perturbation, which is similar to the pressure perturbation response in the previous section.

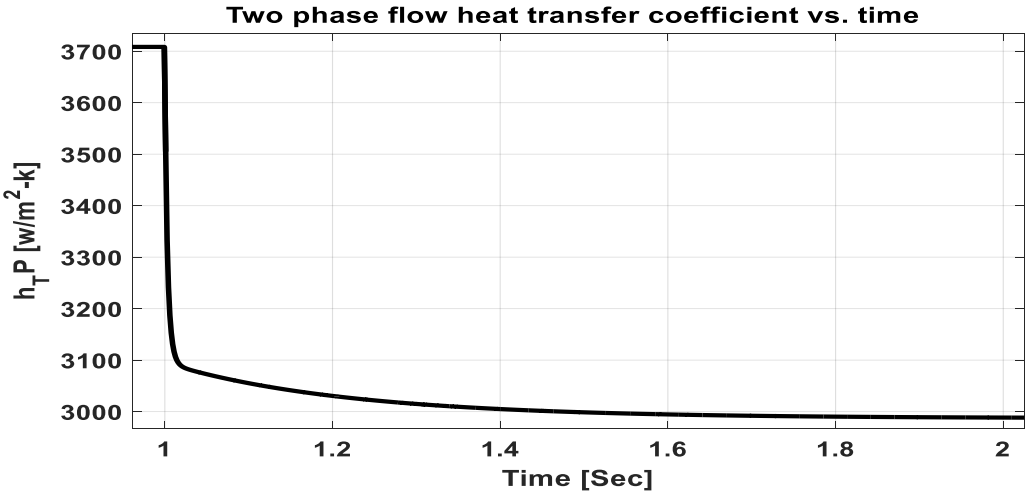


Figure 32: Hot side (R134a) heat transfer coefficient, h_{TP}, with enthalpy step change

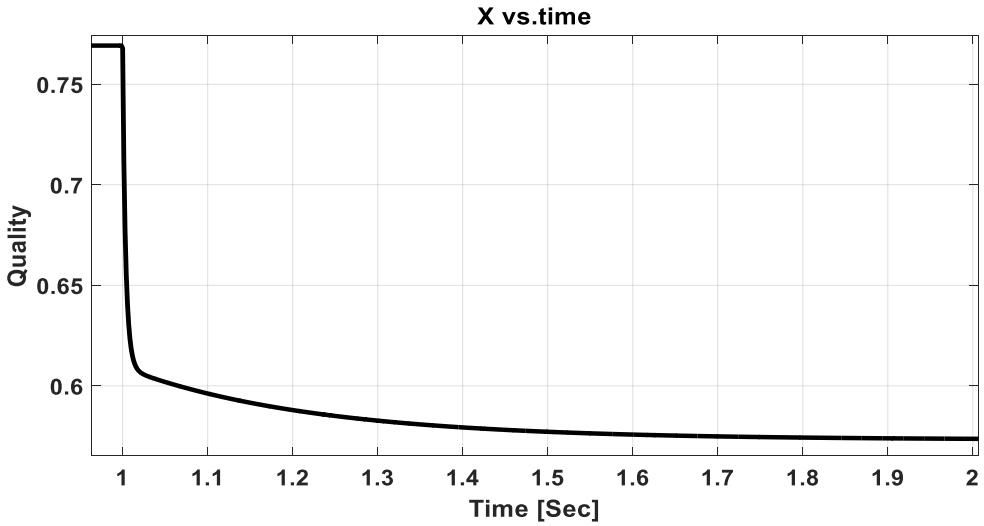


Figure 33: Hot side (R134a) quality with enthalpy step change

Figures 32 and 33 illustrate the convection coefficient and quality response of the hot side refrigerant (R134a). As enthalpy decreases, the quality will decrease. Since h_{TP} is related to the quality, therefore h_{TP} will also decrease. Figure 34 shows the heat transferred to the cold side (Kerosene) Q_c, which is reduced due to the decrease in temperature difference between the heat exchanger and Kerosene. Figure 35 presents the heat transferred from the hot side (R134a) to the heat exchanger during the perturbation. There are two different responses to the enthalpy step change. The step change in enthalpy results in a near step change in the quality, which drives the heat transfer. The heat exchanger thermal mass drives a slower response time. The temperature difference between the R134a and heat exchanger material is increasing because T_h is constant at saturated pressure and T_{HX} decreases. So, the difference is larger which explains Q_h gets larger slowly over time.

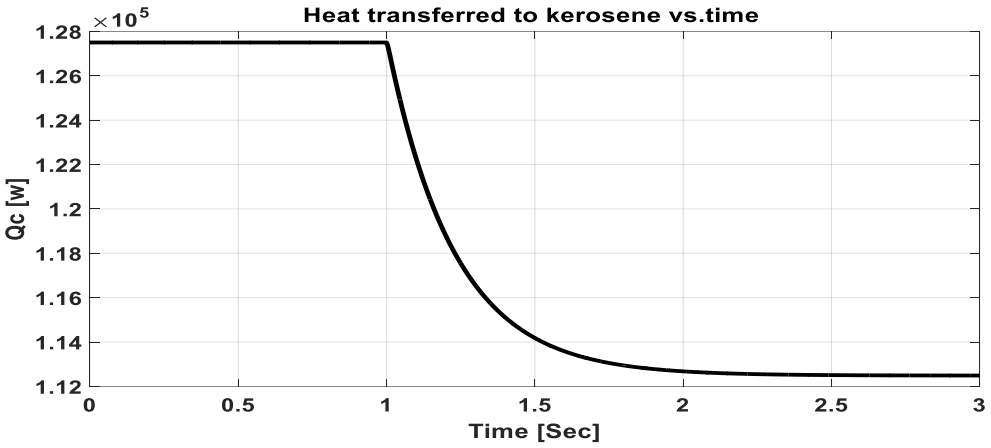


Figure 34: Heat transferred to the cold fluid (Kerosene) with enthalpy step change

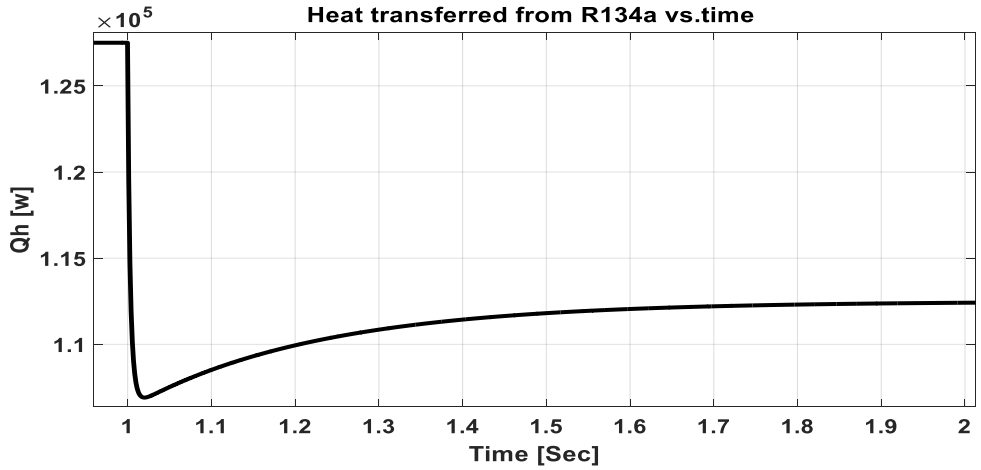


Figure 35: Heat transferred from hot side (R134a) with enthalpy step change

Now, the entropy variation is analyzed. The change in entropy is based on the enthalpy step change. A non- physical instantaneous change in entropy is predicted as shown in Figure 36 at the instant the enthalpy is changed. This happens because of the instantaneous change of a non-dynamic parameter. Table 9 presents the steady- state entropy values.

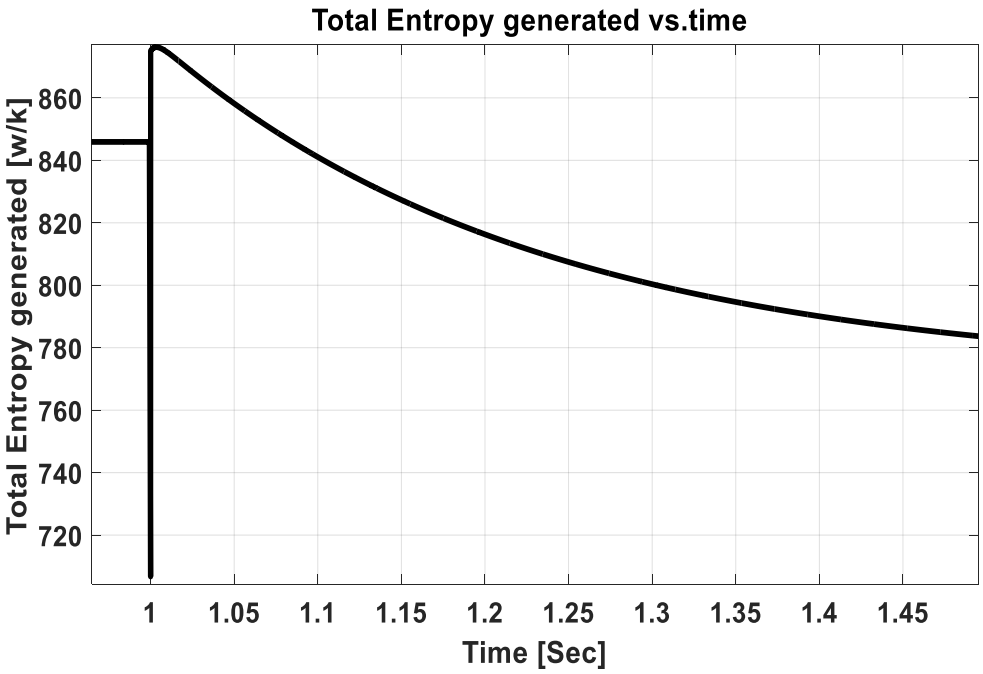


Figure 36: Total entropy generated with enthalpy step change

Table 9: Steady state total entropy generated

ref	Enthalpy (kJ/kg)	Entropy generated (w/k)
1.	350	846
2.	300	771

Realistic Pressure Response

Previously, an ideal step change in pressure was investigated. In this section, a transfer function is employed to emulate the response time of a plenum volume and a valve for perturbing the pressure of the system. The ideal input pressure would be a step change but a first order response is more realistic. The initial pressure is 650 kPa, while the final pressure is 555 kPa. This investigation will provide a more realistic test of the two-phase flow heat exchanger model using different time constants. Finally, a comparison between the transfer function results and the previous pressure step change results can be made. The sample time has been taken as 0.00001sec. To approximate the volume for a given

time constant, the ideal gas equation of state is used for simplicity even though R134a does not behave as an ideal gas.

$$P\,V = m\,R\,T \tag{30}$$

is transformed to the following form

$$P = \frac{m\,R\,T}{V} \tag{31}$$

Taking the derivative of Eqn. (31) with respect to time results in

$$\frac{dP}{dt} = \frac{R\,T}{V}\,\frac{dm}{dt} \tag{32}$$

where the term $\left[\frac{V}{R\,T}\right]$ represents the time constant. To calculate the time constant, the temperature (T) is taken as 298 k and R for R134a is 81.5 $\frac{J}{kg-k}$. Note, that Eqn. (32) assumes temperature is not changing with time. This does introduce some additional error in the approximation of the time constant for a given volume and temperature. Two different time constants were analyzed to evaluate a range of approximate volumes. From Figure 19 in pressure step change section the heat exchanger mass time constant is 0.3 seconds which will be constant for both time constants 0.1 and 0.01 seconds.

Time constant = 0.1 sec.

Figure 37 shows the hot side (R134a) pressure change from 650 kPa to 555 kPa as a first order response with a time constant of 0.1 seconds. In this case, the plenum volume is approximately 2.4 m^3.

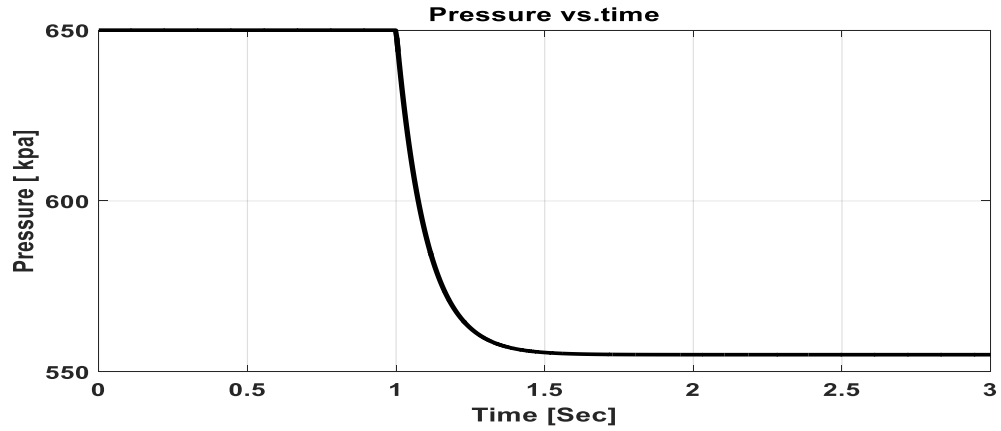


Figure 37: Hot side (R134a) pressure response for a time constant of 0.1sec

Figure 38 plots the hot side (R134a) temperature out during the pressure response for a 0.1 time constant. The hot side (R134a) temperature changes simultaneously with respect to the pressure changes if the hot side (R134a) quality is less than 1. Therefore, $T_{h,out}$ decreases as the pressure decreases from 650 kPa to 555 kPa.

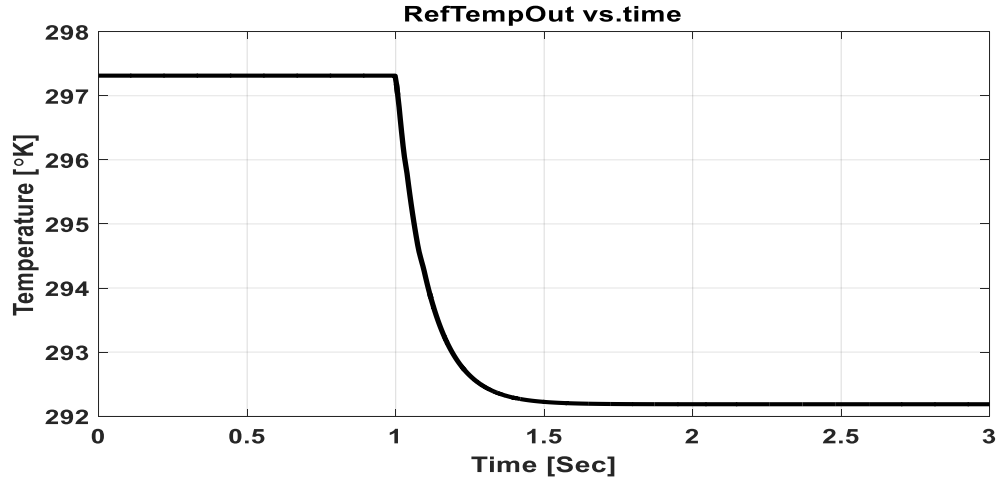


Figure 38: Hot side (R134a) temperature out for a time constant of 0.1sec

Figure 39 shows the hot side (R134a) quality during the pressure response for a 0.1 time constant. Since the pressure decreases, the quality increases as stated earlier. In the time range (1~1.3) seconds, the hot side (R134a) quality increases due to the decrease in the refrigerant pressure. Next, the quality decreases after 1.3 seconds because the hot side

(R134a) heat transferred increases. The change in equilibrium of the fluid is more evident in Figure 41.

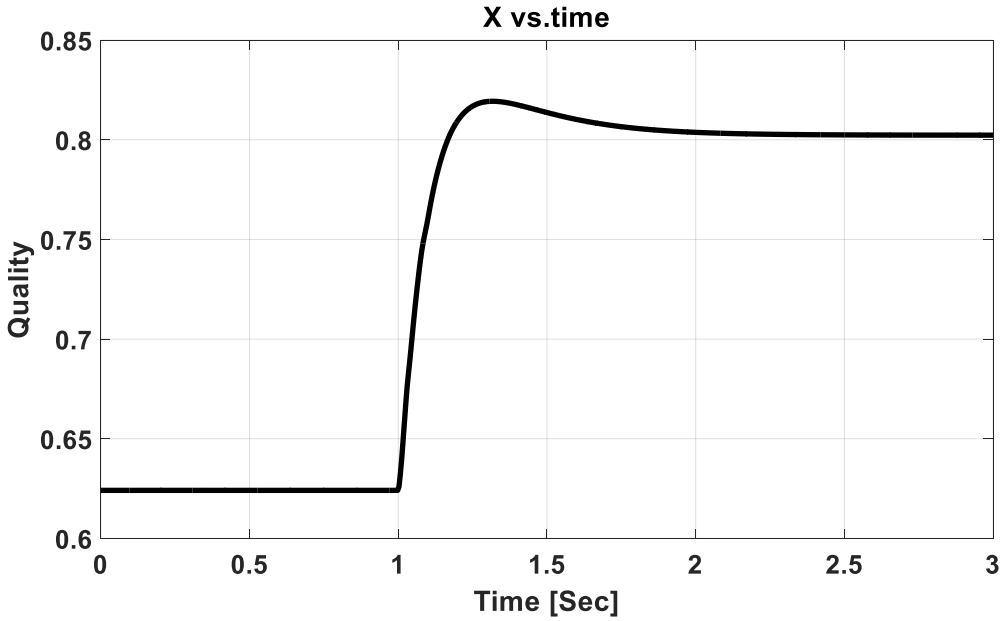


Figure 39: Hot side (R134a) quality for a time constant of 0.1sec

Figure 40 shows the heat transferred from the hot side refrigerant during the pressure response for a transfer function with a 0.1 sec time constant. The heat transfer rate does not have the sudden step change as in the previous section, but still has the undershoot due to the changing quality of the fluid as it reaches a new equilibrium.

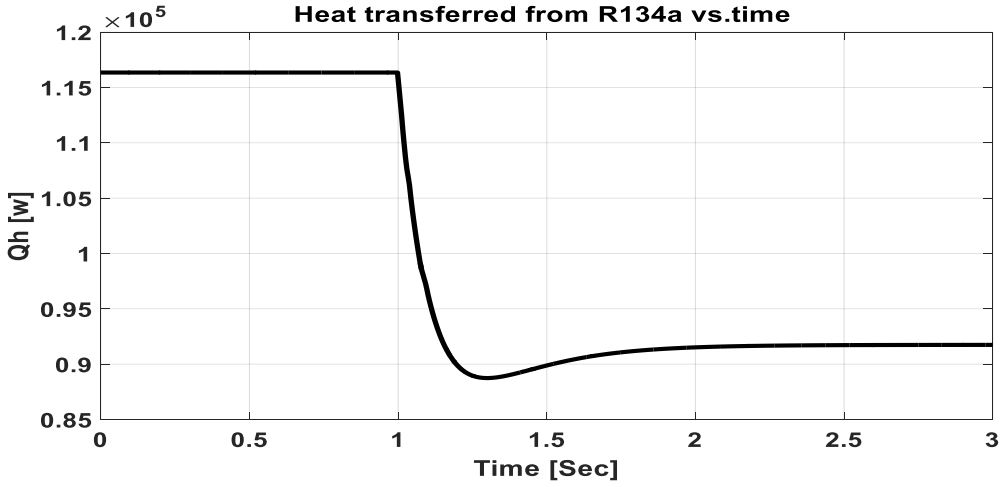


Figure 40: Heat transferred from hot side (R134a) for a time constant of 0.1sec

The temperature difference between the hot side (R134a) temperature and the heat exchanger material temperature decreases by time during the transient region as shown in Figure 41 because $T_{h,out}$ decreases as a result of pressure decreases.

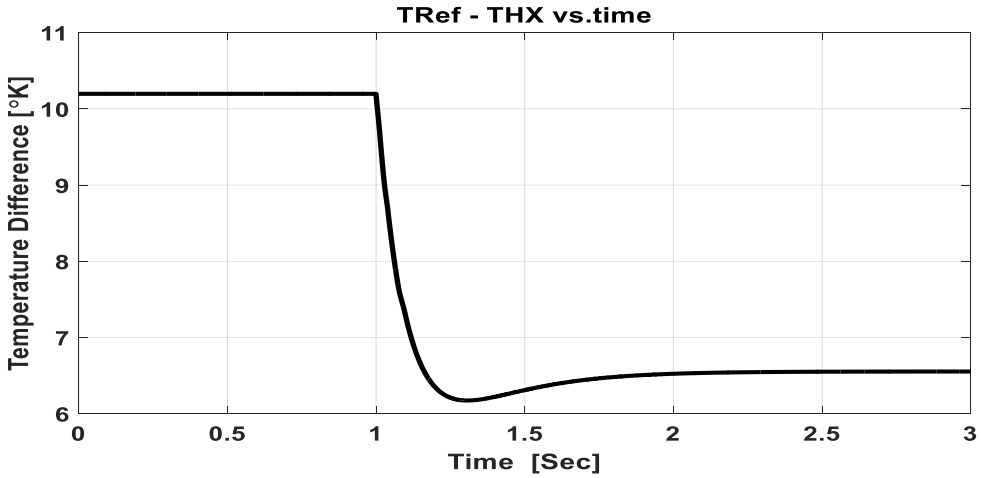


Figure 41: Temperature difference between Hot side (R134a) and the heat exchanger material for a time constant of 0.1sec

Figure 42 shows the total entropy generated during the pressure response for a transfer function with 0.1 sec time constant. Since the total entropy generated is a function of the temperatures and the latter decrease due to the pressure drop, the total entropy generated decreases.

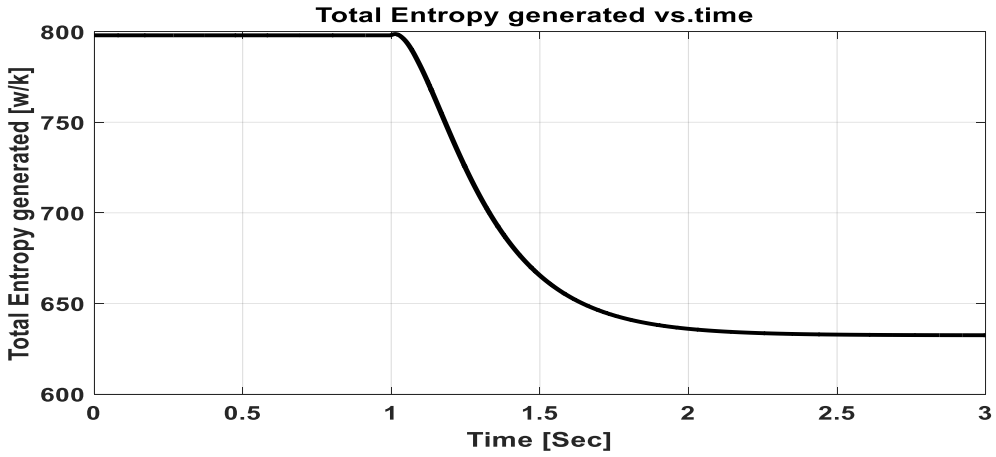


Figure 42: Total entropy generated for a transfer function with a time constant of 0.1sec

After running the model with a specified time constant (0.1 sec), a comparison to the step change results is done. All figures of the results are identical except for pressure, $T_{h,out}$, Q_h, and total entropy generated results. Therefore, a 0.1 second time constant with a first order response does not introduce any issues with the model. The approximate time constant of the heat exchanger temperature is 0.3 seconds. With the plenum volume time constant of 0.1 seconds the heat exchanger-to-plenum volume ratio of time constants is approximately 3-to-1. This ratio of time constants results in normal operation of the heat exchanger, but larger ratios of time constants result in different behavior in heat transfer as is presented in the next section.

Time constant = 0.01 sec.

Figure 43 shows the hot side (R134a) pressure change from 650 kPa to 555 kPa as a first order response with a time constant of 0.01 seconds. The plenum volume for this time constant is approximately 0.24 m^3. The pressure response is relatively faster and equates to a 30- to- 1 ratio for heat exchanger temperature to plenum volume time constant.

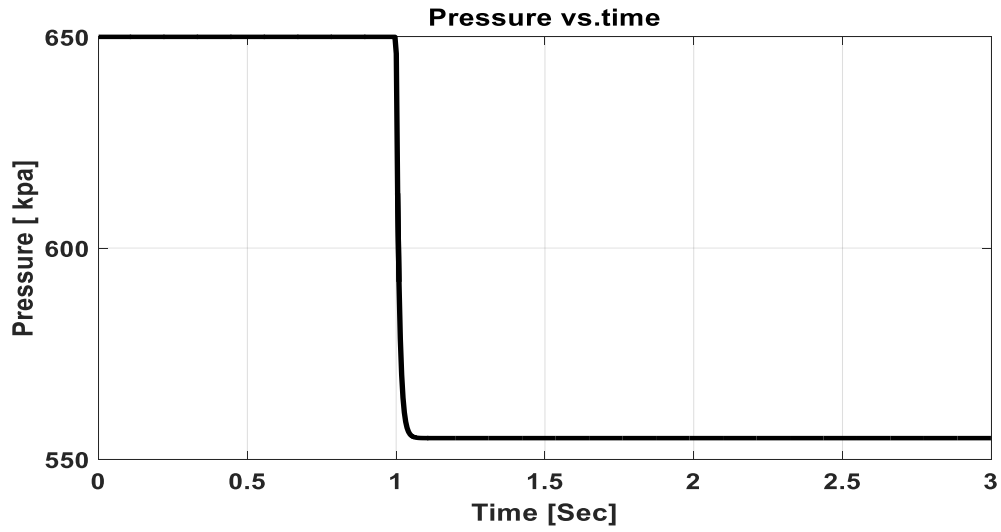


Figure 43: Hot side (R134a) pressure response for a time constant of 0.01sec

Figure 44 shows the hot side (R134a) temperature out during the pressure change for a 0.01 second transfer function. As expected, the hot side (R134a) temperature direction is similar to the hot side (R134a) pressure as long as the refrigerant quality is less than one. The results of the $T_{h,out}$ for 0.1 time constant are identical to the results of the $T_{h,out}$ for 0.01 time constant. Therefore, selecting different time constants will not affect the results.

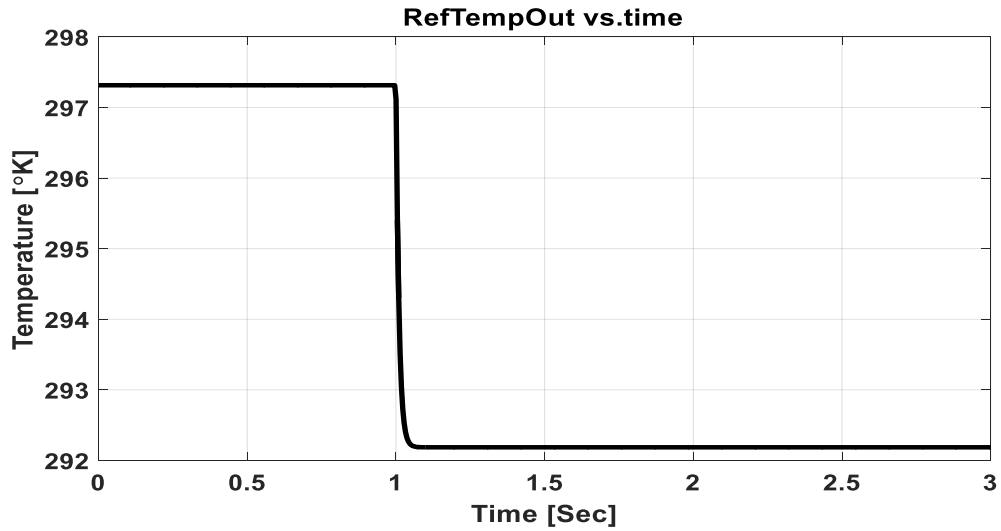


Figure 44: Hot side (R134a) temperature out for a time constant of 0.01sec

Figure 45 shows the temperature difference between the hot side (R134a) and the heat exchanger materials. In the time range (1~ 1.04) seconds, the temperature difference decreases due to the hot side (R134a) temperature decreases because the pressure decreases. After 1.04 seconds, the temperature difference increases due to the decrease in the heat exchanger material temperature.

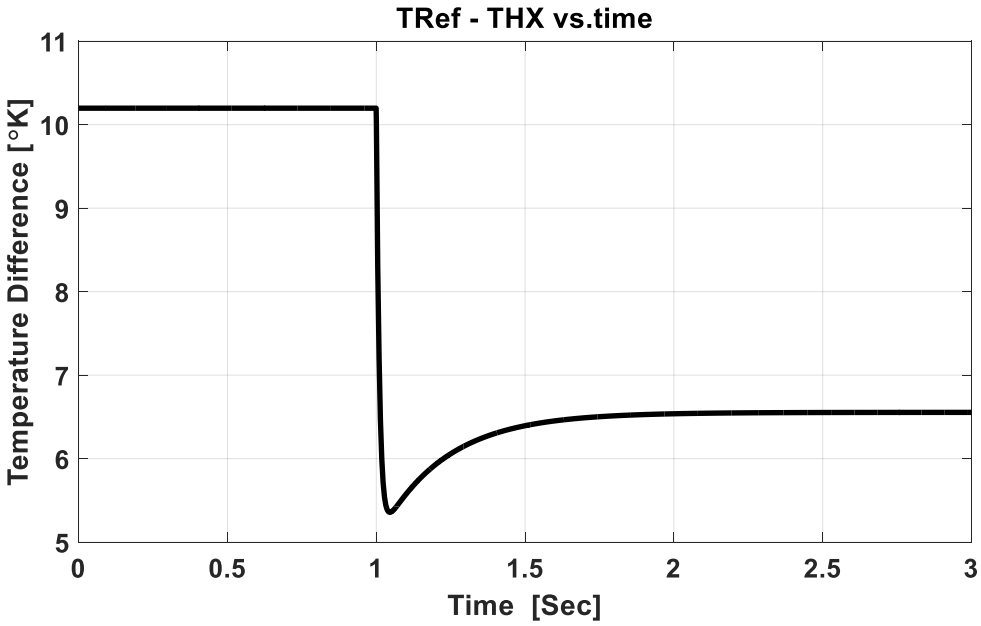


Figure 45: Temperature difference between Hot side (R134a) and the heat exchanger material for a time constant of 0.01sec

Figure 46 shows the hot side (R134a) quality during the pressure response for a 0.01 time constant. Again, once the pressure decreases, the quality increases and vice versa. The response of the hot side (R134a) quality is quicker for this time constant than for 0.1 time constant. Hence, in one second, the refrigerant quality approximately reaches the equilibrium state.

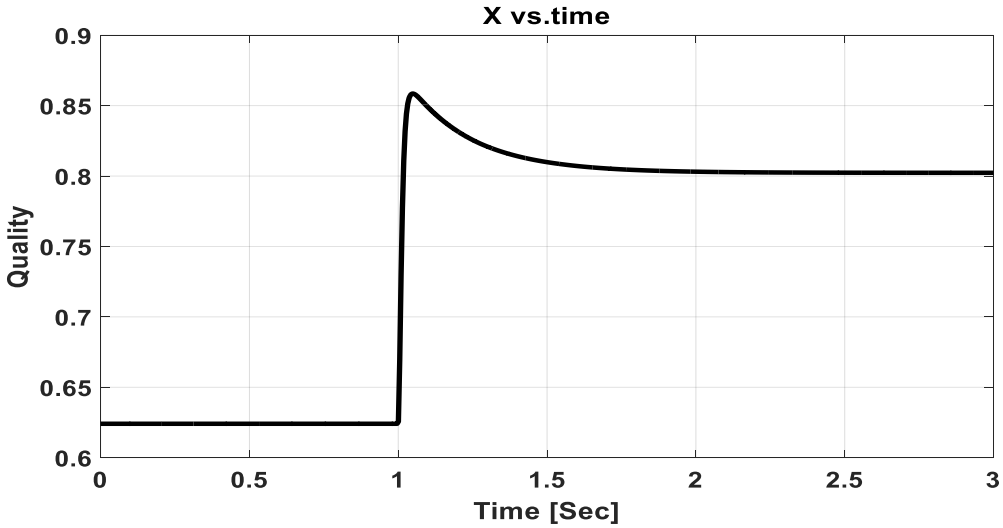


Figure 46: Hot side (R134a) quality for a time constant of 0.01sec

Figure 47 shows the heat transferred from the refrigerant during a 0.01 second time constant transfer function- pressure change. The hot side (R134a) heat transferred in this Figure is similar to hot side (R134a) heat transferred in the pressure step change. Hence, there is a region in which the evaporation is occurring instead of condensation. In the time range (1~ 1.04) seconds, the hot side (R134a) heat transferred decreases as the $T_{h,out}$ decreases due to the pressure decreases Next, after 1.04 seconds, the refrigerant heat transferred increases according to the temperature difference increase in that range of time.

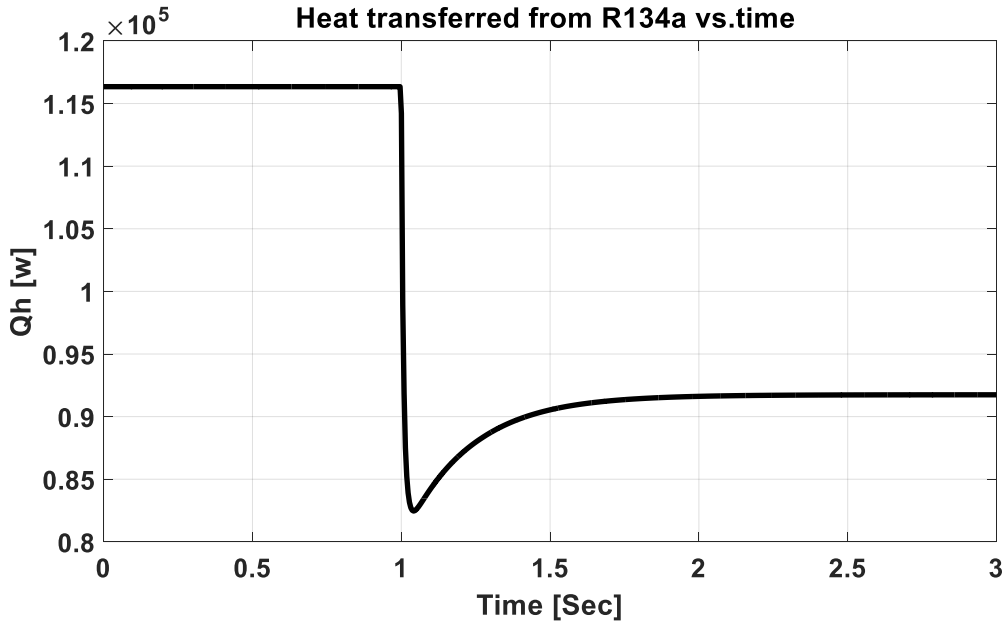


Figure 47: Heat transferred from hot side (R134a) for a time constant of 0.01sec

Figure 48 represents the total entropy generated during time constant of 0.01 seconds first order pressure change.

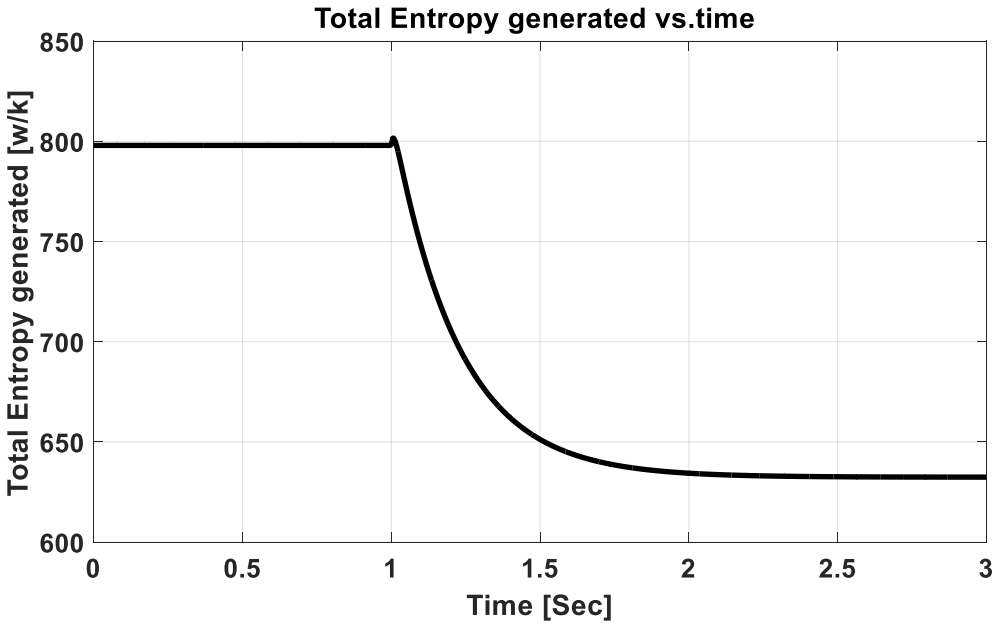


Figure 48: Total entropy generated for a time constant of 0.01sec

CONCLUSION

A stainless-steel two- phase heat exchanger has been modeled transiently. The numerical simulation tool (MATLAB/SIMULINK) was used to study and understand the influence of input perturbations on the operation of the heat exchanger. Two working fluids were selected; Kerosene and R134a.

The heat exchanger model demonstrates its ability to respond quickly to two different perturbations; pressure and enthalpy in very short period of time. Therefore, the results prove manipulation of pressure will dramatically change the heat transfer of R134a. The pressure drop results in a decrease in; kerosene temperature out, R134a temperature out, heat exchanger temperature, kerosene heat transfer coefficient, heat transferred to kerosene, heat transferred from R134a, and the total entropy generated. In turn, two- phase heat transfer coefficient, quality, and R134a enthalpy out increase as a result of the pressure drop. In particular, h_{TP} will be increased approximately twice. There are two time constants of interest. The smaller time constant of the refrigerant fluid effected the fluid quality, enthalpy and heat transfer. Pressure manipulation of a refrigerant provides the opportunity to quickly change the fluid properties and overall heat transfer rate much faster than manipulating the inlet enthalpy or temperature. As shown in the enthalpy perturbation, the fluid temperature does not change. Pressure manipulation provides a rapid capability of manipulating both the heat transfer rate and the temperature of the refrigerant fluid. If a thermal system is designed with the ability to manipulate pressure, then the heat transfer rate may be actively rapidly controlled.

APPENDICES

APPENDIX A

In the Simulink model, it is very necessary to set up many lookup tables. These tables facilitate computing outcomes corresponding to incomes by linear interpolation procedure. There are basically various software/ programs could be used to do this task. However, EES program has been chosen to create these tables.

Table 10: Dynamic viscosity for kerosene versus temperature

Lookup Table

Dynamic viscosity for kerosene

Paste Special	1 Temperature [k]	2 Dynamic viscosity [pa -sec]
Row 1	293	0.004077
Row 2	303	0.003377
Row 3	313	0.002797
Row 4	323	0.002317
Row 5	333	0.001919
Row 6	343	0.001589
Row 7	353	0.001316
Row 8	363	0.00109
Row 9	373	0.0009025
Row 10	383	0.0009025
Row 11	393	0.0009025
Row 12	403	0.0009025
Row 13	413	0.0009025
Row 14	423	0.0009025
Row 15	433	0.0009025
Row 16	443	0.0009025
Row 17	453	0.0009025
Row 18	463	0.0009025
Row 19	473	0.0009025
Row 20	483	0.0009025

Table 11: Dynamic viscosity for R-134a @ different pressure lines versus enthalpy

Lookup Table

Dynamic viscosity for R-134a at different pressure lines

	1 enthalpy [J/kg]	2 mu @ p_1 = 66.19 [pa - s]	3 mu @ p_2 = 132.8 [pa -s]	4 mu @ p_3 = 243.5 [pa - s]	5 mu @ p_4 = 414.9 [pa -s]	6 mu @ p_5 = 665.8 [pa -s]
Row 1	6290	0.0004305	0.0003462	0.0004317	0.0004328	0.0004345
Row 2	27012	0.0003913	0.0003438	0.0003411	0.000342	0.0003433
Row 3	47733	0.0003522	0.0003111	0.0002797	0.0002769	0.000278
Row 4	68455	0.000313	0.0002784	0.0002518	0.0002305	0.0002286
Row 5	89176	0.0002738	0.0002457	0.0002239	0.0002063	0.0001915
Row 6	109898	0.0002347	0.0002129	0.000196	0.0001821	0.0001703
Row 7	130619	0.0001955	0.0001802	0.0001681	0.0001579	0.000149
Row 8	151341	0.0001564	0.0001475	0.0001402	0.0001337	0.0001277
Row 9	172063	0.0001172	0.0001148	0.0001123	0.0001095	0.0001065
Row 10	192784	0.00007803	0.00008212	0.00008439	0.00008534	0.00008522
Row 11	213506	0.00003887	0.00004941	0.00005649	0.00006114	0.00006396
Row 12	234227	0.000009796	0.00001671	0.00002859	0.00003695	0.00004269
Row 13	254949	0.00001082	0.0000109	0.00001104	0.00001275	0.00002143
Row 14	275671	0.00001179	0.00001185	0.00001196	0.00001212	0.00001239
Row 15	296392	0.00001271	0.00001276	0.00001284	0.00001297	0.00001318
Row 16	317114	0.00001359	0.00001363	0.00001369	0.0000138	0.00001398
Row 17	337835	0.00001442	0.00001446	0.00001451	0.00001461	0.00001476
Row 18	358557	0.00001522	0.00001525	0.0000153	0.00001539	0.00001552
Row 19	379278	0.00001599	0.00001602	0.00001607	0.00001614	0.00001626
Row 20	400000	0.00001674	0.00001676	0.0000168	0.00001687	0.00001698

Table 12: Density for R-134a @ different pressure lines versus enthalpy versus enthalpy

Lookup Table

Density for R-134a at different pressure lines

Paste Special	1 enthalpy [J/kg]	2 density @ p_1 = 66.19 [kg/m3]	3 density @ p_2 = 132.8 [kg/m3]	4 density @ p_3 = 243.5 [kg/m3]	5 density @ p_4 = 414.9 [kg/m3]	6 density @ p_5 = 665.8 [kg/m3]
Row 1	6290	1403	1403	1404	1404	1405
Row 2	27012	36.99	560	1355	1356	1357
Row 3	47733	18.73	62.3	555.2	1306	1307
Row 4	68455	12.54	32.99	98.07	639.8	1254
Row 5	89176	9.429	22.43	53.79	146.1	770.6
Row 6	109898	7.553	16.99	37.05	82.47	208.3
Row 7	130619	6.3	13.68	28.26	57.45	120.4
Row 8	151341	5.403	11.44	22.84	44.08	84.68
Row 9	172063	4.73	9.838	19.17	35.76	65.31
Row 10	192784	4.206	8.627	16.51	30.08	53.14
Row 11	213506	3.787	7.682	14.5	25.95	44.8
Row 12	234227	3.413	6.923	12.93	22.83	38.72
Row 13	254949	3.051	6.196	11.59	20.37	34.1
Row 14	275671	2.777	5.621	10.46	18.25	30.33
Row 15	296392	2.56	5.171	9.591	16.64	27.42
Row 16	317114	2.383	4.806	8.893	15.37	25.18
Row 17	337835	2.235	4.504	8.318	14.34	23.39
Row 18	358557	2.109	4.247	7.834	13.47	21.92
Row 19	379278	2.001	4.026	7.419	12.74	20.68
Row 20	400000	1.906	3.833	7.058	12.1	19.61

Table 13: Specific heat for R134a @ different pressure lines versus enthalpy

Lookup Table

Specific heat for R-134a at different pressure lines

Paste Special	1 enthalpy [J/kg]	2 Cp @ p_1 = 66.19 [J/kg - k]	3 Cp @ p_2 = 132.8 [J/kg - k]	4 Cp @ p_3 = 243.5 [J/kg - k]	5 Cp @ p_4 = 414.9 [J/kg - k]	6 Cp@ p_5 = 665.8 [J/kg - k]
Row 1	6290	1264	1293	1263	1262	1262
Row 2	27012	1217	1290	1295	1294	1293
Row 3	47733	1171	1243	1322	1332	1331
Row 4	68455	1124	1197	1276	1364	1376
Row 5	89176	1078	1150	1229	1317	1418
Row 6	109898	1031	1104	1183	1271	1373
Row 7	130619	985.1	1057	1137	1225	1327
Row 8	151341	938.6	1011	1090	1179	1281
Row 9	172063	892.2	964.5	1044	1133	1235
Row 10	192784	845.8	918.1	997.7	1087	1190
Row 11	213506	799.4	871.6	951.3	1041	1144
Row 12	234227	769	825.2	905	994.5	1098
Row 13	254949	802.9	826	869	948.3	1052
Row 14	275671	842.3	857	882.4	925	995.9
Row 15	296392	881.1	891.6	909.3	937.9	982.8
Row 16	317114	918.4	926.2	939.5	960.5	992.7
Row 17	337835	954	960.1	970.3	986.5	1011
Row 18	358557	987.8	992.7	1001	1014	1033
Row 19	379278	1020	1024	1031	1041	1057
Row 20	400000	1051	1054	1060	1069	1082

Table 14: Thermal conductivity for R134a @ different pressure lines versus enthalpy

Lookup Table

Thermal conductivity for R-134a at different pressure lines

	1 enthalpy [J/kg]	2 k @ p_1 = 66.19 [w/m-k]	3 k @ p_2 = 132.8 [w/m-k]	4 k @ p_3 = 243.5 [w/m-k]	5 k @ p_4 = 414.9 [w/m-k]	6 k @ p_5 = 665.8 [w/m-k]
Row 1	6290	0.1083	0.1084	0.1084	0.1085	0.1087
Row 2	27012	0.09905	0.1021	0.1024	0.1025	0.1027
Row 3	47733	0.08978	0.0931	0.09566	0.09606	0.09626
Row 4	68455	0.0805	0.08408	0.08694	0.089	0.08944
Row 5	89176	0.07123	0.07506	0.07822	0.08062	0.08216
Row 6	109898	0.06195	0.06605	0.06951	0.07224	0.07416
Row 7	130619	0.05268	0.05703	0.06079	0.06386	0.06615
Row 8	151341	0.0434	0.04801	0.05207	0.05548	0.05815
Row 9	172063	0.03412	0.039	0.04335	0.0471	0.05014
Row 10	192784	0.02485	0.02998	0.03463	0.03872	0.04214
Row 11	213506	0.01557	0.02096	0.02591	0.03033	0.03413
Row 12	234227	0.009281	0.01195	0.01719	0.02195	0.02613
Row 13	254949	0.01169	0.01194	0.01234	0.01357	0.01813
Row 14	275671	0.01378	0.01396	0.01425	0.01471	0.01539
Row 15	296392	0.01558	0.01572	0.01594	0.0163	0.01682
Row 16	317114	0.01712	0.01723	0.01742	0.0177	0.01812
Row 17	337835	0.01845	0.01854	0.01869	0.01892	0.01927
Row 18	358557	0.01957	0.01964	0.01977	0.01997	0.02026
Row 19	379278	0.02051	0.02057	0.02068	0.02086	0.02111
Row 20	400000	0.02129	0.02135	0.02144	0.02159	0.02182

Table 15: Temperature for R134a @ different pressure lines versus enthalpy

Lookup Table

Temperature for R-134a at different pressure lines

Paste Special	1 enthalpy [J/kg]	2 T @ p_1 = 66.19 [k]	3 T @ p_2 = 132.8 [k]	4 T @ p_3 = 243.5 [k]	5 T @ p_4 = 414.9 [k]	6 T @ p_5 = 665.8 [k]
Row 1	6290	238.1	238.1	238.1	238	238
Row 2	27012	238.2	253.1	254.3	254.3	254.2
Row 3	47733	238.2	253.1	268.1	270	270
Row 4	68455	238.2	253.1	268.1	283.2	285.3
Row 5	89176	238.2	253.1	268.1	283.2	298.2
Row 6	109898	238.2	253.1	268.1	283.2	298.2
Row 7	130619	238.2	253.1	268.1	283.2	298.2
Row 8	151341	238.2	253.1	268.1	283.2	298.2
Row 9	172063	238.2	253.1	268.1	283.2	298.2
Row 10	192784	238.2	253.1	268.1	283.2	298.2
Row 11	213506	238.2	253.1	268.1	283.2	298.2
Row 12	234227	244.9	253.1	268.1	283.2	298.2
Row 13	254949	271.4	273.4	276.7	283.2	298.2
Row 14	275671	296.5	298	300.4	304.1	309.5
Row 15	296392	320.6	321.7	323.6	326.4	330.5
Row 16	317114	343.6	344.5	346	348.2	351.5
Row 17	337835	365.8	366.5	367.7	369.5	372.2
Row 18	358557	387.1	387.7	388.7	390.3	392.5
Row 19	379278	407.7	408.3	409.1	410.4	412.3
Row 20	400000	427.7	428.2	428.9	430.1	431.7

Table 16: Specific volume for R134a vs. internal energy @ different pressure lines

Lookup Table

Specific volume for R134a

Paste Special	1 Internal energy [J/kg]	2 v @ p_1 = 66.19 [m³/kg]	3 v @ p_2 = 132.8 [m³/kg]	4 v @ p_3 = 243.5 [m³/kg]	5 v @ p_4 = 414.9 [m³/kg]	6 v @ p_5 = 665.8 [m³/kg]
Row 1	6250	0.0007127	0.0007126	0.0007125	0.0007124	0.0007122
Row 2	26974	0.02947	0.001937	0.0007382	0.000738	0.0007377
Row 3	47697	0.05823	0.01764	0.001982	0.0007664	0.000766
Row 4	68421	0.08699	0.03334	0.0113	0.001738	0.000798
Row 5	89145	0.1158	0.04904	0.02061	0.007644	0.001456
Row 6	109868	0.1445	0.06475	0.02993	0.01355	0.005405
Row 7	130592	0.1733	0.08045	0.03924	0.01946	0.009354
Row 8	151316	0.202	0.09615	0.04856	0.02536	0.0133
Row 9	172039	0.2308	0.1119	0.05787	0.03127	0.01725
Row 10	192763	0.2596	0.1276	0.06719	0.03717	0.0212
Row 11	213487	0.2903	0.1433	0.07651	0.04308	0.02515
Row 12	234211	0.3294	0.1621	0.08642	0.04899	0.0291
Row 13	254934	0.3653	0.1804	0.09681	0.0554	0.03321
Row 14	275658	0.3987	0.1973	0.1063	0.06125	0.03711
Row 15	296382	0.4302	0.2132	0.1152	0.06666	0.04066
Row 16	317105	0.4601	0.2283	0.1236	0.07173	0.04395
Row 17	337829	0.4887	0.2427	0.1316	0.07652	0.04704
Row 18	358553	0.5161	0.2565	0.1392	0.08108	0.04997
Row 19	379276	0.5425	0.2697	0.1465	0.08546	0.05277
Row 20	400000	0.5681	0.2826	0.1536	0.08967	0.05545

Table 17: Specific heat for R-134a @ liquid phase versus sat. temperature

Lookup Table

Cp for R-134a @ liquid phase

Paste Special	1 Temperature [K]	2 Cp [J/kg-k]
Row 1	238	1263
Row 2	245.1	1277
Row 3	252.2	1291
Row 4	259.3	1307
Row 5	266.4	1324
Row 6	273.5	1342
Row 7	280.6	1363
Row 8	287.7	1385
Row 9	294.8	1411
Row 10	301.9	1441
Row 11	309.1	1476
Row 12	316.2	1517
Row 13	323.3	1567
Row 14	330.4	1630
Row 15	337.5	1714
Row 16	344.6	1831
Row 17	351.7	2013
Row 18	358.8	2346
Row 19	365.9	3218
Row 20	373	14134

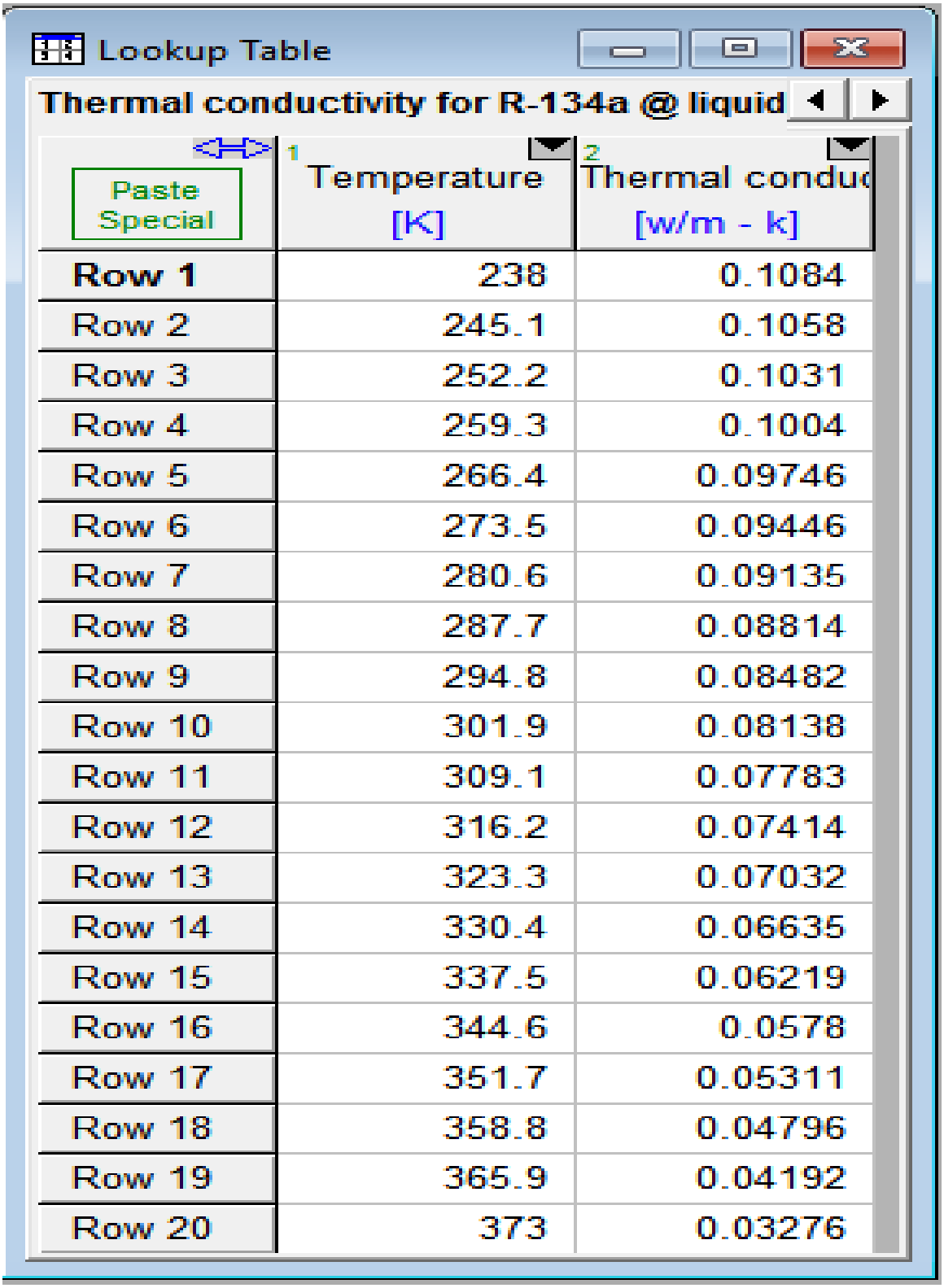
Table 18: Dynamic viscosity for R-134a @ liquid phase versus sat. temperature

Lookup Table

Dynamic viscosity for R-134a @ liquid ph

Paste Special	1 Temperature [K]	2 Dynamic viscos [pa-sec]
Row 1	238	0.0004315
Row 2	245.1	0.0003881
Row 3	252.2	0.0003507
Row 4	259.3	0.0003182
Row 5	266.4	0.0002896
Row 6	273.5	0.0002642
Row 7	280.6	0.0002416
Row 8	287.7	0.0002211
Row 9	294.8	0.0002025
Row 10	301.9	0.0001854
Row 11	309.1	0.0001697
Row 12	316.2	0.0001551
Row 13	323.3	0.0001414
Row 14	330.4	0.0001285
Row 15	337.5	0.0001161
Row 16	344.6	0.0001042
Row 17	351.7	0.00009236
Row 18	358.8	0.00008034
Row 19	365.9	0.00006721
Row 20	373	0.00004845

Table 19: Thermal conductivity for R-134a @ liquid phase versus sat. temperature



Lookup Table

Thermal conductivity for R-134a @ liquid

Paste Special	1 Temperature [K]	2 Thermal conduc [w/m - k]
Row 1	238	0.1084
Row 2	245.1	0.1058
Row 3	252.2	0.1031
Row 4	259.3	0.1004
Row 5	266.4	0.09746
Row 6	273.5	0.09446
Row 7	280.6	0.09135
Row 8	287.7	0.08814
Row 9	294.8	0.08482
Row 10	301.9	0.08138
Row 11	309.1	0.07783
Row 12	316.2	0.07414
Row 13	323.3	0.07032
Row 14	330.4	0.06635
Row 15	337.5	0.06219
Row 16	344.6	0.0578
Row 17	351.7	0.05311
Row 18	358.8	0.04796
Row 19	365.9	0.04192
Row 20	373	0.03276

Table 20: Density for R-134a @ liquid phase versus sat. temperature

Lookup Table

Density for R134a @ liquid sat. line

Paste Special	1 Temperature [k]	2 Density [kg/m3]
Row 1	238	1404
Row 2	248	1374
Row 3	258	1343
Row 4	268	1312
Row 5	278	1279
Row 6	288	1244
Row 7	298	1207
Row 8	308	1168
Row 9	318	1126
Row 10	328	1079
Row 11	338	1027
Row 12	348	965.1
Row 13	358	888.6
Row 14	368	775.6

Table 21: Specific heat for R-134a @ vapor sat. phase versus sat. temperature

Lookup Table

Specific heat for R-134a @ vapor phase

Paste Special	1 Temperature [k]	2 Specific heat [J/kg -k]
Row 1	238	764.1
Row 2	248	787.5
Row 3	258	812.4
Row 4	268	839.2
Row 5	278	868
Row 6	288	899.1
Row 7	298	932.9
Row 8	308	970
Row 9	318	1011
Row 10	328	1057
Row 11	338	1110
Row 12	348	1173
Row 13	358	1248
Row 14	368	1343
Row 15	337.5	1469
Row 16	344.6	1650
Row 17	351.7	1936
Row 18	358.8	2476
Row 19	365.9	3934
Row 20	373	26923

Table 22: Dynamic viscosity for R-134a @ vapor sat. phase versus temperature

Lookup Table

Dynamic viscosity for R-134a @ vapor phase

Paste Special	1 Temperature [k]	2 Dynamic viscosity [pa - sec]
Row 1	238	0.000009525
Row 2	248	0.000009804
Row 3	258	0.00001008
Row 4	268	0.00001036
Row 5	278	0.00001065
Row 6	288	0.00001093
Row 7	298	0.00001122
Row 8	308	0.00001152
Row 9	318	0.00001182
Row 10	328	0.00001214
Row 11	338	0.00001248
Row 12	348	0.00001284
Row 13	358	0.00001325
Row 14	368	0.0000137
Row 15	337.5	0.00001425
Row 16	344.6	0.00001492
Row 17	351.7	0.00001579
Row 18	358.8	0.00001705
Row 19	365.9	0.00001916
Row 20	373	0.00002539

Table 23: Thermal conductivity for R-134a @ vapor sat. phase versus sat. temperature

Lookup Table

Thermal conductivity for R-134a @ vapor phase

Paste Special	1 Temperature [k]	2 Thermal conductivity [w/m-k]
Row 1	238	0.008609
Row 2	248	0.009328
Row 3	258	0.01004
Row 4	268	0.01074
Row 5	278	0.01144
Row 6	288	0.01214
Row 7	298	0.01283
Row 8	308	0.01353
Row 9	318	0.01423
Row 10	328	0.01494
Row 11	338	0.01567
Row 12	348	0.01643
Row 13	358	0.01721
Row 14	368	0.01804
Row 15	337.5	0.01894
Row 16	344.6	0.01993
Row 17	351.7	0.02107
Row 18	358.8	0.02247
Row 19	365.9	0.02441
Row 20	373	0.02974

Table 24: Density for R-134a @ vapor sat. line versus sat. temperature

Lookup Table

Density for R134a @ vapor sat. line

Paste Special	1 Temperature [k]	2 Density [kg/m3]
Row 1	238	3.499
Row 2	248	5.474
Row 3	258	8.244
Row 4	268	12.02
Row 5	278	17.06
Row 6	288	23.66
Row 7	298	32.23
Row 8	308	43.26
Row 9	318	57.46
Row 10	328	75.85
Row 11	338	100.1
Row 12	348	133.1
Row 13	358	181.4
Row 14	368	266.5

Table 25: Enthalpy of evaporation for R-134a. versus sat. temperature

Lookup Table

Temperature vs enthalpy of evaporation for R134a

Paste Special	1 Temperature [k]	2 Enthalpy of evaporation [J/kg]
Row 1	238	222814
Row 2	245.1	218254
Row 3	252.2	213548
Row 4	259.3	208672
Row 5	266.4	203602
Row 6	273.5	198309
Row 7	280.6	192764
Row 8	287.7	186931
Row 9	294.8	180770
Row 10	301.9	174231
Row 11	309.1	167255
Row 12	316.2	159764
Row 13	323.3	151659
Row 14	330.4	142806
Row 15	337.5	133012
Row 16	344.6	121989
Row 17	351.7	109258
Row 18	358.8	93908
Row 19	365.9	73672
Row 20	373	35379

Table 26: Entropy for R134a @ different pressure lines versus enthalpy

Lookup Table

Entropy for R134a @ different pressure lines

Paste Special	1 Enthalpy [J/kg]	2 s @ p_1 = 66.16 [J/kg - k]	3 s @ p_2 = 132.8 [J/kg -k]	4 s @ p_3 = 243.5 [J/kg -k]	5 s @ p_4 = 414.9 [J/kg -k]	6 s @ p_5 = 665.8 [J/kg -k]
Row 1	6290	26.65	26.45	26.12	25.6	24.85
Row 2	27012	113.7	110.6	110.3	109.8	109.1
Row 3	47733	200.7	192.5	189.4	188.9	188.2
Row 4	68455	287.7	274.3	266.7	263.5	262.8
Row 5	89176	374.7	356.2	343.9	336.7	333.6
Row 6	109898	461.7	438	421.2	409.9	403.1
Row 7	130619	548.7	519.9	498.5	483.1	472.6
Row 8	151341	635.7	601.7	575.7	556.2	542.1
Row 9	172063	722.7	683.6	653	629.4	611.6
Row 10	192784	809.7	765.4	730.3	702.6	681.1
Row 11	213506	896.7	847.3	807.6	775.8	750.6
Row 12	234227	983.4	929.2	884.8	848.9	820.1
Row 13	254949	1064	1009	961.7	922.1	889.6
Row 14	275671	1137	1081	1034	992.9	958.4
Row 15	296392	1204	1148	1100	1059	1023
Row 16	317114	1266	1210	1162	1120	1084
Row 17	337835	1325	1268	1220	1178	1141
Row 18	358557	1380	1323	1275	1232	1195
Row 19	379278	1432	1376	1327	1284	1247
Row 20	400000	1482	1425	1376	1333	1296

APPENDIX B

In this appendix, MATLAB files will be presented to set the foundation of creating the Simulink model. These files are doing four major tasks. First, setting the thermodynamic properties (specific heat, density, thermal conductivity, and dynamic viscosity) for fluids used in the heat exchanger model which are kerosene and R-134a. Second, initializing dimensions and size of the heat exchanger. Third, exporting all properties into the Simulink model mask. Therefore, the model will be ready to process all these data and find the results. Finally, plot outcomes with respect to time

The code below is written to set the thermodynamic properties.

```
function [cp ,rho, k, mu, muT] = fluidproperties(val)
%val = input ('give me the name of fluid   ')
 %%% Fluid properties,
    % Specific heat, kJ/kg/K            [in polyval() form]
    % Density, kg/m^3                   [in polyval() form]
    % Thermal conductivity, W/m/K       [in polyval() form]
    % Dynamic viscosity, kg/m/s         [in interp1() form]
    % Temperatures for mu, K            [in interp1() form]
 switch val
     case {'PAO'}
            cp = [3.7749e-3 1.02255];
            rho = [1.5859e-8 -2.6056e-5 1.4797e-2 -4.37867 1346.36];
            k = [5.8823e-5 1.5411e-1];
            mu = [0.90889, 0.11814, 0.030046, 0.011483, 0.0056626,
0.003295, 0.0021441, 0.0015094, 0.0011254, 0.0008761, 0.0008275];
            muT = [220, 240, 260, 280, 300, 320, 340, 360, 380, 400,
405];
```

```
        case {'JP8'}
            cp = [4.43359e-3 6.48908e-1];
            rho = [-7.23225e-1 1.02036e3];
            k = [-1.799e-4 1.67663e-1];
            mu = [0.0137298, 0.00807381, 0.00515649, 0.00360287,
0.00265331, 0.00205681, 0.00162952, 0.00131464, 0.00109445,
0.000937041, 0.000804244, 0.000701833, 0.000617844, 0.000543122,
0.000485191, 0.000435239, 0.000388231, 0.000353682, 0.000321021,
0.000295718];
            muT = [222.778, 232.778, 242.778, 252.778, 262.778,
272.778, 282.778, 292.778, 302.778, 312.778, 322.778, 332.778, 342.778,
352.778, 362.778, 372.778, 382.778, 392.778, 402.778, 412.778];
       case {'AIR'}
            cp = [2.80023e-13 -1.0498e-9 1.38033e-6 -0.000535927
1.06747];
            rho = [2.09244e-16 -7.65474e-13 1.12524e-9 -8.51881e-7
0.000353846 -0.0793459 8.72074];
            k = [-3.6206e-14 9.9793e-11 -1.13283e-7 0.000118727 -
0.00171684];
            mu = [7.06e-6, 0.00001038, 0.00001336, 0.00001606,
0.0000172, 0.00001769, 0.00001853, 0.00001911, 0.00002002, 0.00002081,
0.00002177, 0.00002294, 0.00002682, 0.0000303, 0.00003349, 0.00003643,
0.00003918, 0.00004177];
            muT = [100, 150, 200, 250, 273, 283, 300, 313, 333, 350,
373, 400, 500, 600, 700, 800, 900, 1000];
        case {'H2O'}
            cp = [4.0871e-10 -5.9806e-7 3.37478e-4 -8.56835e-2
1.23546e1];
```

```
            rho = [-2.9681e-8 4.91535e-5 -3.25219e-2 9.18133
8.21792e1];
            k = [-2.98213e-11 5.47109e-8 -4.25128e-5 1.54729e-2 -
1.44067];
            mu = [0.001791, 0.001308, 0.001003, 0.0007977, 0.0006531,
0.0005471, 0.0004668, 0.0004044, 0.0003549, 0.000315, 0.0002822,
0.0001961, 0.0001494, 0.000121, 0.0001015];
            muT = [273.16, 283.15, 293.15, 303.15, 313.15, 323.15,
333.15, 343.15, 353.15, 363.15, 373.15, 413.15, 453.15, 493.15,
533.15];
        case {'Kerosene'}
            cp = [0 2.01];
            rho = [0 820 ];
            k = [0 0.15];
            mu = [0.004077 0.003377 0.002797 0.002317 0.001919 0.001589
0.001316 0.00109 0.0009025 0.0009025 0.0009025 0.0009025 0.0009025
0.0009025 0.0009025 0.0009025 0.0009025 0.0009025 0.0009025 0.0009025
];
            muT = [293 303 313 323 333 343 353 363 373 383 393 403 413
423 433 443 453 463 473 483];
    case {'R134a'}    % these values are exact numbers because we need
to specify the exact number @ each internal energy value NOT in
polynomials format
            cp = [1264 1293 1263 1262 1262; 1217 1290 1295 1294
1293;1171 1243 1322 1332 1331;1124 1197 1276 1364 1376;1078 1150 1229
1317 1418;1031 1104 1183 1271 1373;985.1 1057 1137 1225 1327;938.6 1011
1090 1179 1281;892.2 964.5 1044 1133 1235;845.8 918.1 997.7 1087
1190;799.4 871.6 951.3 1041 1144;769 825.2 905 994.5 1098;802.9  826
869 948.3 1052;842.3 857 882.4 925 995.9;881.1 891.6 909.3 937.9
```

```
982.8;918.4 926.2 939.5 960.5 992.7;954 960.1 970.3 986.5 1011;987.8
992.7 1001 1014 1033;1020 1024 1031 1041 1057;1051 1054 1060 1069
1082].*1e-03;

            rho = [1403 1403 1404 1404 1405;36.99 560 1355 1356
1357;18.73 62.3 555.2 1306 1307;12.54 32.99 98.07 639.8 1254; 9.429
22.43 53.79 146.1 770.6; 7.553 16.99 37.05 82.47 208.3; 6.3 13.68 28.26
57.45 120.4; 5.403 11.44 22.84 44.08 84.68; 4.73 9.838 19.17 35.76
65.31; 4.206 8.627 16.51 30.08 53.14; 3.787 7.682 14.5 25.95 44.8;
3.413 6.923 12.93 22.83 38.72;3.051 6.196 11.59 20.37 34.1;2.777 5.621
10.46 18.25 30.33; 2.56 5.171 9.591 16.64 27.42; 2.383 4.806 8.893
15.37 25.18; 2.235 4.504 8.318 14.34 23.39; 2.109 4.247 7.834 13.47
21.92;2.001 4.026 7.419 12.74 20.68;1.906 3.833 7.058 12.1 19.61 ];

            k = [0.1083 0.1084 0.1084 0.1085 0.1087; 0.09905 0.1021
0.1024 0.1025 0.1027; 0.08978 0.0931 0.09566 0.09606 0.09626; 0.0805
0.08408 0.08694 0.089 0.08944; 0.07123 0.07506 0.07822 0.08062 0.08216;
0.06195 0.06605 0.06951 0.07224 0.07416; 0.05268 0.05703 0.06079
0.06386 0.06615; 0.0434 0.04801 0.05207 0.05548 0.05815; 0.03412 0.039
0.04335 0.0471 0.05014; 0.02485 0.02998 0.03463 0.03872 0.04214;
0.01557 0.02096 0.02591 0.03033 0.03413; 0.009281 0.01195 0.01719
0.02195 0.02613;0.01169 0.01194 0.01234 0.01357 0.01813; 0.01378
0.01396 0.01425 0.01471 0.01539; 0.01558 0.01572 0.01594 0.0163
0.01682; 0.01712 0.01723 0.01742 0.0177 0.01812; 0.01845 0.01854
0.01869 0.01892 0.01927;0.01957 0.01964 0.01977 0.01997 0.02026;
0.02051 0.02057 0.02068 0.02086 0.02111; 0.02129 0.02135 0.02144
0.02159 0.02182];

            mu = [0.0004305 0.0003462 0.0004317 0.0004328
0.0004345;0.0003913 0.0003438 0.0003411 0.000342 0.0003433;0.0003522
0.0003111 0.0002797 0.0002769 0.000278;0.000313 0.0002784 0.0002518
0.0002305 0.0002286;0.0002738 0.0002457 0.0002239 0.0002063 0.0001915;
```

```
0.0002347 0.0002129 0.000196 0.0001821 0.0001703;0.0001955 0.0001802
0.0001681 0.0001579 0.000149;0.0001564 0.0001475 0.0001402 0.0001337
0.0001277;0.0001172 0.0001148 0.0001123 0.0001095 0.0001065;0.00007803
0.00008212 0.00008439 0.00008534 0.00008522;0.00003887 0.00004941
0.00005649 0.00006114 0.00006396;0.000009796 0.00001671 0.00002859
0.00003695 0.00004269;0.00001082 0.0000109 0.00001104 0.00001275
0.00002143;0.00001179 0.00001185 0.00001196 0.00001212
0.00001239;0.00001271 0.00001276 0.00001284 0.00001297
0.00001318;0.00001359 0.00001363 0.00001369 0.0000138
0.00001398;0.00001442 0.00001446 0.00001451 0.00001461
0.00001476;0.00001522 0.00001525 0.0000153 0.00001539
0.00001552;0.00001599 0.00001602 0.00001607 0.00001614
0.00001626;0.00001674 0.00001676 0.0000168 0.00001687 0.00001698];
            muT = [6290 27012 47733 68455 89176 109898 130619 151341
172063 192784 213506 234227 254949 275671 296392 317114 337835 358557
379278 400000];
end
% disp (cp)
% disp (rho)
% disp(k)
% disp(mu)
% disp(muT)
end
```

The following code is written to initialize the dimensions and size of the heat exchanger.

```
function [parameter] = HXsize_v12(params,fluids,params2,material)
params = [1,1, 6.35,0.1, 0.9 ,0.83, 10, 6.35, 0.1, 0.9, 0.07]; %
numbers from simulink model
        %[1 , 0.83, 6.35,0.1, 0.9 ,0.83, 10.6, 6.35, 0.1, 0.9, 0.07]
```

```
params2 = [ 1 , 1.5 ,293, 360.3 , 32, 1 ]; % numbers from simulink
model
%HXsize(s1, h1, tf1, l1, t1, L, s2, h2, tf2, l2, t2) % %For Matlabs
optimization toolbox.
%This function written by Peter Weise determines the weight and
pressure
%drop of a compact heat exchanger based on its physical parameters and
%heat transfer characteristics.  The methodology used is laid out in
the
%book "Compact Heat Exchangers: Selection, Design, and Operation" by
J.E.
%Hesselgreaves.  Also see "Compact Heat Exchangers" by Kays and London.
 %%% Heat exchanger fixed parameters:  The following parameters need to
be
%%% fixed for a given optimization trial.  However, they can be varied.
%%% For example, the heat load varies from one heat exchanger to
another.
%%% Additionally, the fluids vary from one exchanger to another.
s1 = params(1)/1000;       %
h1 = params(2)/1000;       %
l1 = params(3)/1000;       %
tf1 = params(4)/1000;      %
t1 = params(5)/1000;       %
s2 = params(6)/1000;       % Only used for MATLAB optimization toolbox.
h2 = params(7)/1000;       %
l2 = params(8)/1000;       %
tf2 = params(9)/1000;      %
t2 = params(10)/1000;      %
L = params (11);      %
```

```
fluids.ctype = get_param(gcb,'ctype');
fluids.htype = get_param(gcb,'htype');
fluid1 = fluids.ctype;
fluid2 = fluids.htype;
m_dot1 = params2(1);     %(kg/s) cold flow rate
m_dot2 = params2(2);     %(kg/s) hot flow rate
T_in1 = params2(3);     %(K) inlet temperature. In the actual T2T
model, the inlet temperature will be based on downstream conditions and
accepted as an input.
T_in2 = params2(4);      %(K) inlet temperature. See note above
Q_dot = params2(5);      %(kW) heat load
%set_param(gcb, 'mdot1', m_dot1);
% set_param(gcb, 'mdot2', m_dot2);
% set_param(gcb, 'Tin1', T_in1);
% set_param(gcb, 'Tin2', T_in2);
% set_param(gcb, 'load', Q_dot);
eta = 0.7;     %(N) fin efficiency
b1=h1+tf1;
b2=h2+tf2;
%% Material density,
%mat = get_param(gcb,'material');
%material = mat;
%rho_m = (kg/m^3) material density
switch material
    case {'Stainless steel - 316'}
        rho_m = 8027;
    case {'Aluminum'}
        rho_m = 2707;
    case {'Copper'}
```

```
        rho_m = 8954;
end
%%  Solves for the properties of each fluid based on inlet
temperatures, steady state outlet temperatures and fluid property
correlations.
    [f1.cp, f1.rho, f1.k, f1.mu, f1.muT] = fluidproperties(fluid1);
    [f2.cp, f2.rho, f2.k, f2.mu, f2.muT] = fluidproperties(fluid2);
    b = 500 ;    % pressure line
    a = 300000 ;  % enthalpy
    %%% Specific heat
    cp1 = polyval(f1.cp,T_in1);
 %  cp2 = polyval(f2.cp,T_in2);
    X = [66.19 132.8 243.5 414.9 665.8];
    Y = [6290 27012 47733 68455 89176 109898 130619 151341 172063
192784 213506 234227 254949 275671 296392 317114 337835 358557 379278
400000] ;
    Z_cp = [1264 1293 1263 1262 1262; 1217 1290 1295 1294 1293;1171
1243 1322 1332 1331;1124 1197 1276 1364 1376;1078 1150 1229 1317
1418;1031 1104 1183 1271 1373;985.1 1057 1137 1225 1327;938.6 1011 1090
1179 1281;892.2 964.5 1044 1133 1235;845.8 918.1 997.7 1087 1190;799.4
871.6 951.3 1041 1144;769 825.2 905 994.5 1098;802.9 826 869 948.3
1052;842.3 857 882.4 925 995.9;881.1 891.6 909.3 937.9 982.8;918.4
926.2 939.5 960.5 992.7;954 960.1 970.3 986.5 1011;987.8 992.7 1001
1014 1033;1020 1024 1031 1041 1057;1051 1054 1060 1069 1082].*1e-03;
    cp2 = interp2(X,Y,Z_cp,b,a);
    %%% Density
    rho1 = polyval(f1.rho,T_in1);
 %  rho2 = polyval(f2.rho,T_in2);
```

```
    Z_rho= [1403 1403 1404 1404 1405;36.99 560 1355 1356 1357;18.73
62.3 555.2 1306 1307;12.54 32.99 98.07 639.8 1254; 9.429 22.43 53.79
146.1 770.6; 7.553 16.99 37.05 82.47 208.3; 6.3 13.68 28.26 57.45
120.4; 5.403 11.44 22.84 44.08 84.68; 4.73 9.838 19.17 35.76 65.31;
4.206 8.627 16.51 30.08 53.14; 3.787 7.682 14.5 25.95 44.8; 3.413 6.923
12.93 22.83 38.72;3.051 6.196 11.59 20.37 34.1;2.777 5.621 10.46 18.25
30.33; 2.56 5.171 9.591 16.64 27.42; 2.383 4.806 8.893 15.37 25.18;
2.235 4.504 8.318 14.34 23.39; 2.109 4.247 7.834 13.47 21.92;2.001
4.026 7.419 12.74 20.68;1.906 3.833 7.058 12.1 19.61 ];
    rho2 = interp2(X,Y,Z_rho,b,a);
    %%% Thermal Conductivity
    k1 = polyval(f1.k,T_in1);
  % k2 = polyval(f2.k,T_in2);
    Z_k = [0.1083   0.1084 0.1084 0.1085 0.1087; 0.09905 0.1021 0.1024
0.1025 0.1027; 0.08978 0.0931 0.09566 0.09606 0.09626; 0.0805 0.08408
0.08694 0.089 0.08944; 0.07123 0.07506 0.07822 0.08062 0.08216; 0.06195
0.06605 0.06951 0.07224 0.07416; 0.05268 0.05703 0.06079 0.06386
0.06615; 0.0434 0.04801 0.05207 0.05548 0.05815; 0.03412 0.039 0.04335
0.0471 0.05014; 0.02485 0.02998 0.03463 0.03872 0.04214; 0.01557
0.02096 0.02591 0.03033 0.03413; 0.009281 0.01195 0.01719 0.02195
0.02613;0.01169 0.01194 0.01234 0.01357 0.01813; 0.01378 0.01396
0.01425 0.01471 0.01539; 0.01558 0.01572 0.01594 0.0163 0.01682;
0.01712 0.01723 0.01742 0.0177 0.01812; 0.01845 0.01854 0.01869 0.01892
0.01927;0.01957 0.01964 0.01977 0.01997 0.02026; 0.02051 0.02057
0.02068 0.02086 0.02111; 0.02129 0.02135 0.02144 0.02159 0.02182];
    k2 = interp2(X,Y,Z_k,b,a);
    %%% Approximate outlet temperature
    T_out1 = T_in1 + Q_dot/(m_dot1*cp1);
    T_out2 = T_in2 - Q_dot/(m_dot2*cp2);
```

```
    %%% Dynamic Viscocity
    mu1 = interp1(f1.muT,f1.mu,(T_in1+T_out1)/2);
  % mu2 = interp1(f2.muT,f2.mu,(T_in2+T_out2)/2);
    Z_mu = [0.0004305   0.0003462 0.0004317 0.0004328
0.0004345;0.0003913 0.0003438 0.0003411 0.000342 0.0003433;0.0003522
0.0003111 0.0002797 0.0002769 0.000278;0.000313 0.0002784 0.0002518
0.0002305 0.0002286;0.0002738 0.0002457 0.0002239 0.0002063 0.0001915;
0.0002347 0.0002129 0.000196 0.0001821 0.0001703;0.0001955 0.0001802
0.0001681 0.0001579 0.000149;0.0001564 0.0001475 0.0001402 0.0001337
0.0001277;0.0001172 0.0001148 0.0001123 0.0001095 0.0001065;0.00007803
0.00008212 0.00008439 0.00008534 0.00008522;0.00003887 0.00004941
0.00005649 0.00006114 0.00006396;0.000009796 0.00001671 0.00002859
0.00003695 0.00004269;0.00001082 0.0000109 0.00001104 0.00001275
0.00002143;0.00001179 0.00001185 0.00001196 0.00001212
0.00001239;0.00001271 0.00001276 0.00001284 0.00001297
0.00001318;0.00001359 0.00001363 0.00001369 0.0000138
0.00001398;0.00001442 0.00001446 0.00001451 0.00001461
0.00001476;0.00001522 0.00001525 0.0000153 0.00001539
0.00001552;0.00001599 0.00001602 0.00001607 0.00001614
0.00001626;0.00001674 0.00001676 0.0000168 0.00001687 0.00001698];
    mu2 = interp2(X,Y,Z_mu,b,a);
%%  Calculate important heat exchanger physical parameters
    parameter.dh1=(4*s1*h1*l1)/(2*(s1*l1+h1*l1+tf1*h1)+tf1*s1);
    parameter.dh2=(4*s2*h2*l2)/(2*(s2*l2+h2*l2+tf2*h2)+tf2*s2);
    sigma1=s1*(b1-tf1)/(s1+tf1)/(b1+t1);    %porosity of side 1
    sigma2=s2*(b2-tf2)/(s2+tf2)/(b2+t2);    %porosity of side 2
    alpha1=s1/h1;  %various aspect ratios used in calculating j and f
    delta1=tf1/l1;
    gamma1=tf1/s1;
```

```
    alpha2=s2/h2;
    delta2=tf2/l2;
    gamma2=tf2/s2;
   %%% Initialize Ac1 and Ac2.  Correct values will be solved for in an
    %%% iterative process.
    parameter.Ac1_opt = 0.01;
    parameter.Ac2_opt = 0.01;
    difference1 = inf;
    difference2= inf;
    count = 0;
    %% This iterative while loop solves for Ac1 and Ac2.
    if params2(6)>0.5
      while count<=100 && abs(difference1)>=1e-5  &&
abs(difference2)>=1e-5
             %%% parameter.Reynolds' number
            Re1=m_dot1*parameter.dh1/mu1/parameter.Ac1_opt;
            Re2=m_dot2*parameter.dh2/mu2/parameter.Ac2_opt;
             %%% Fanning friction factor
            parameter.f1=9.6243*(Re1^-0.7422)*(alpha1^-
0.1856)*(delta1^0.3053)*(gamma1^-0.2659)*((1+7.669E-
8*(Re1^4.429)*(alpha1^0.92)*(delta1^3.767)*(gamma1^0.236))^0.1);
            parameter.f2=9.6243*(Re2^-0.7422)*(alpha2^-
0.1856)*(delta2^0.3053)*(gamma2^-0.2659)*((1+7.669E-
8*(Re2^4.429)*(alpha2^0.92)*(delta2^3.767)*(gamma2^0.236))^0.1);
            %%% Colburn coefficient
            j1=0.6522*(Re1^-0.5403)*(alpha1^-0.1541)*(delta1^-
0.1409)*(gamma1^-0.0678)*((1+5.269E-
5*(Re1^1.34)*(alpha1^0.504)*(delta1^0.456)*(gamma1^-1.055))^0.1);
```

```
            j2=0.6522*(Re2^-0.5403)*(alpha2^-0.1541)*(delta2^-
0.1409)*(gamma2^-0.0678)*((1+5.269E-
5*(Re2^1.34)*(alpha2^0.504)*(delta2^0.456)*(gamma2^-1.055))^0.1);
             %%% Effectiveness-Ntu method
            T_h1=max(T_in1,T_in2);
            T_c1=min(T_in1,T_in2);
            Cmax=max(m_dot1*cp1,m_dot2*cp2);
            Cmin=min(m_dot1*cp1,m_dot2*cp2);
            Cstar=Cmin/Cmax;
            epsilon = Q_dot/(Cmin*(T_h1-T_c1));
            %if 0>=epsilon || epsilon >1
                %disp('epsilon must be between 0 and 1. Heat load
(Q_dot) is too large or T_h1-T_c1 is too small')
                %break
            %end
             if 0.99<=Cstar<=1.01
                Ntu=epsilon/(1-epsilon);
            else
                Ntu=Log((epsilon-1)/(Cstar*epsilon-1))/(Cstar-1);
            end
             N=2*Ntu/eta;
             %%% Pressure drop
parameter.del_P1=((m_dot1/parameter.Ac1_opt)^2)*4*L*parameter.f1/2/rho1
/parameter.dh1;
parameter.del_P2=((m_dot2/parameter.Ac2_opt)^2)*4*L*parameter.f2/2/rho2
/parameter.dh2;
            %%% Prandtl number
            Pr1=cp1*mu1/k1 * 1000;
            Pr2=cp2*mu2/k2 * 1000 ;
```

```
            %%% Flow velocity
 G1=sqrt(2*rho1*parameter.del_P1*j1/parameter.f1/(Pr1^(2/3))/N);
G2=sqrt(2*rho2*parameter.del_P2*j2/parameter.f2/(Pr2^(2/3))/N);
            %%% Flow area
           Ac1_0=m_dot1/G1;
           Ac2_0=m_dot2/G2;
            difference1=parameter.Ac1_opt-Ac1_0;
           difference2=parameter.Ac2_opt-Ac2_0;
            parameter.Ac1_opt=Ac1_0;
           parameter.Ac2_opt=Ac2_0;
            count = count + 1;
        end
            %parameter.Ac1_opt=parameter.Ac1;
     %parameter.Ac2_opt=parameter.Ac2;
    else
    parameter.Ac1_opt= str2num(get_param(gcb,'Ac1_opt'));
    parameter.Ac2_opt= str2num(get_param(gcb,'Ac2_opt'));
    end
   parameter.Ac1= parameter.Ac1_opt;
   parameter.Ac2= parameter.Ac2_opt;
   parameter.Ac1;
   parameter.Ac2;
   %disp('iterations =')
   %disp(count)
%%% Calculate volume and Weight
    parameter.weight_kg=rho_m*L*(parameter.Ac1/sigma1*(1-
sigma1)+parameter.Ac2/sigma2*(1-sigma2));
    parameter.weight_f1 = (parameter.Ac1*rho1*L);
    parameter.weight_f2 = parameter.Ac2*rho2*L;
```

```
    parameter.vol_HX = L*(parameter.Ac1/sigma1*(1-
sigma1)+parameter.Ac2/sigma2*(1-sigma2));
    parameter.vol_f = (parameter.Ac1+parameter.Ac2)*L;
    Beta = [4*sigma1/parameter.dh1 4*sigma2/parameter.dh2];
    parameter.vol1 = L*parameter.Ac1;
    parameter.vol2 = L*parameter.Ac2;
    parameter.As1 =Beta(1) * (L*(parameter.Ac1/sigma1*(1-sigma1)) +
parameter.Ac1);
    parameter.As2 =Beta(2) * (L*(parameter.Ac2/sigma2*(1-sigma2)) +
parameter.Ac2);
    parameter.L = L;
 %%% Additional outputs,
    parameter.ratios1 = [alpha1 delta1 gamma1];
    parameter.ratios2 = [alpha2 delta2 gamma2];
    parameter.Achx = (sqrt(parameter.Ac1) +
sqrt(parameter.Ac2))/2*parameter.L;
    parameter.Aht1 = parameter.Ac1*(s1 + tf1)/(s1*h1)*L;
    parameter.Aht2 = parameter.Ac2*(s2 + tf2)/(s2*h2)*L;
%    Feb 7, 2017
```

Third, the next code is written to export and feed the Simulink model.

```
number = str2num(get_param(gcb,'number'));
ctype=get_param(gcb,'ctype');
%%%%%%%%%%%%%%%% START COLD FLUID PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Get parameters,
    %ctype = input (' get the cold flow   ')
    %ctype = get_param(gcb,'ctype');
R_c = '0';
```

```
check_c = '0';              % 1=incompressible, 2= ideal Gas, 3= real
Gas, 4= 2-phase
%%% Fluid properties,
    % Specific heat, J/kg/K            [in polyval() form]
    % Density, kg/m^3                  [in polyval() form]
    % Thermal conductivity, W/m/K      [in polyval() form]
    % Dynamic viscosity, kg/m/s        [in interp1() form]
    % Temperatures for mu, K           [in interp1() form]
    switch ctype
        case {'PAO'}
            cp = '[3.7749e-3 1.02255].*1000';
            rho = '[1.5859e-8 -2.6056e-5 1.4797e-2 -4.37867 1346.36]';
            k = '[1.9058e-21, -5.882e-05, 0.1541]';
            mu = '[0.90889, 0.11814, 0.030046, 0.011483, 0.0056626,
0.003295, 0.0021441, 0.0015094, 0.0011254, 0.0008761, 0.0008275,
.00001]';
            muT = '[220, 240, 260, 280, 300, 320, 340, 360, 380, 400,
405, 1000]';
       case {'JP8'}
            cp = '[4.43359e-3 6.48908e-1].*1000';
            rho = '[-7.23225e-1 1.02036e3]';
            k = '[-1.799e-4 1.67663e-1]';
            mu = '[0.0137298, 0.00807381, 0.00515649, 0.00360287,
0.00265331, 0.00205681, 0.00162952, 0.00131464, 0.00109445,
0.000937041, 0.000804244, 0.000701833, 0.000617844, 0.000543122,
0.000485191, 0.000435239, 0.000388231, 0.000353682, 0.000321021,
0.000295718, .00001]';
```

```
            muT = '[222.778, 232.778, 242.778, 252.778, 262.778,
272.778, 282.778, 292.778, 302.778, 312.778, 322.778, 332.778, 342.778,
352.778, 362.778, 372.778, 382.778, 392.778, 402.778, 412.778, 1000]';
        case {'AIR'}
             cp = '[2.80023e-13 -1.0498e-9 1.38033e-6 -0.000535927
1.06747].*1000';
             rho = '[2.09244e-16 -7.65474e-13 1.12524e-9 -8.51881e-7
0.000353846 -0.0793459 8.72074]';
             k = '[-3.6206e-14 9.9793e-11 -1.13283e-7 0.000118727 -
0.00171684]';
             mu = '[7.06e-6, 0.00001038, 0.00001336, 0.00001606,
0.0000172, 0.00001769, 0.00001853, 0.00001911, 0.00002002, 0.00002081,
0.00002177, 0.00002294, 0.00002682, 0.0000303, 0.00003349, 0.00003643,
0.00003918, 0.00004177]';
             muT = '[100, 150, 200, 250, 273, 283, 300, 313, 333, 350,
373, 400, 500, 600, 700, 800, 900, 1000]';
             R_c = '287';
             check_c = '1';
        case {'H2O'}
            cp = '[4.0871e-10 -5.9806e-7 3.37478e-4 -8.56835e-2
1.23546e1].*1000';
            rho = '[-2.9681e-8 4.91535e-5 -3.25219e-2 9.18133
8.21792e1]';
            k = '[-2.98213e-11 5.47109e-8 -4.25128e-5 1.54729e-2 -
1.44067]';
            mu = '[0.001791, 0.001308, 0.001003, 0.0007977, 0.0006531,
0.0005471, 0.0004668, 0.0004044, 0.0003549, 0.000315, 0.0002822,
0.0001961, 0.0001494, 0.000121, 0.0001015, .0001]';
```

```
            muT = '[273.16, 283.15, 293.15, 303.15, 313.15, 323.15,
333.15, 343.15, 353.15, 363.15, 373.15, 413.15, 453.15, 493.15, 533.15,
1000]';
         case {'Kerosene'}
            cp = '[0 2010]';
            rho = '[0 820 ]';
            k = '[0 0.15]';
            mu = '[0.004077 0.003377 0.002797 0.002317 0.001919
0.001589 0.001316 0.00109 0.0009025 0.0009025 0.0009025 0.0009025
0.0009025 0.0009025 0.0009025 0.0009025 0.0009025 0.0009025 0.0009025
0.0009025 ]';
            muT = '[293 303 313 323 333 343 353 363 373 383 393 403 413
423 433 443 453 463 473 483]';
    end
%%% Calculate enthalpy, J/kg
    icp = [str2num(cp)./(length(str2num(cp)):-1:1), 0];
    %%% For polyval,
    hin = strcat('[',num2str(icp),']');
%%% Calculate entropy J/kg-K
    cp1 = str2num(cp);
    n=length(cp1);
    if n==1
        coef_s = [0 cp1];
    else
        kk = (n-1):-1:1;
        coef_s = [cp1(1:(n-1))./kk 0 cp1(n)];
    end
    s_a = ['[', num2str(coef_s(1:(end-1))), ']'];
    s_end  = num2str(coef_s(end));
```

```
%%% Make coefficients available to masked subsystem
    set_param(gcb, 'rho_c', rho);
    set_param(gcb, 'enthalpy_c', hin);
    set_param(gcb, 'cp_c',cp);
    set_param(gcb, 'mu_Input_c', muT);
    set_param(gcb, 'mu_Output_c', mu);
    set_param(gcb, 'k_c', k);
    set_param(gcb, 'entropy_c', s_a);
    set_param(gcb, 'entropy_end_c', s_end);
    set_param(gcb, 'R_c', R_c);
    set_param(gcb, 'fluid_check_c', check_c);
%%%%%%%%%%%%%%%% END COLD FLUID PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%% START HOT FLUID PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Get parameters,
htype=get_param(gcb,'htype');
R_h = '0';
check_h = '0';
%%% Fluid properties,
    % Specific heat, J/kg/K            [in polyval() form]
    % Density, kg/m^3                   [in polyval() form]
    % Thermal conductivity, W/m/K       [in polyval() form]
    % Dynamic viscosity, kg/m/s         [in interp1() form]
    % Temperatures for mu, K            [in interp1() form]
    switch htype
       case {'PAO'}
          cp = '[3.7749e-3 1.02255].*1000';
          rho = '[1.5859e-8 -2.6056e-5 1.4797e-2 -4.37867 1346.36]';
```

```
            k = '[1.9058e-21, -5.882e-05, 0.1541]';
            mu = '[0.90889, 0.11814, 0.030046, 0.011483, 0.0056626,
0.003295, 0.0021441, 0.0015094, 0.0011254, 0.0008761, 0.0008275,
.00001]';
            muT = '[220, 240, 260, 280, 300, 320, 340, 360, 380, 400,
405, 1000]';
        case {'JP8'}
            cp = '[4.43359e-3 6.48908e-1].*1000';
            rho = '[-7.23225e-1 1.02036e3]';
            k = '[-1.799e-4 1.67663e-1]';
            mu = '[0.0137298, 0.00807381, 0.00515649, 0.00360287,
0.00265331, 0.00205681, 0.00162952, 0.00131464, 0.00109445,
0.000937041, 0.000804244, 0.000701833, 0.000617844, 0.000543122,
0.000485191, 0.000435239, 0.000388231, 0.000353682, 0.000321021,
0.000295718, .00001]';
            muT = '[222.778, 232.778, 242.778, 252.778, 262.778, 272.778,
282.778, 292.778, 302.778, 312.778, 322.778, 332.778, 342.778, 352.778,
362.778, 372.778, 382.778, 392.778, 402.778, 412.778, 1000]';
        case {'AIR'}
            cp = '[2.80023e-13 -1.0498e-9 1.38033e-6 -0.000535927
1.06747].*1000';
            rho = '[2.09244e-16 -7.65474e-13 1.12524e-9 -8.51881e-7
0.000353846 -0.0793459 8.72074]';
            k = '[-3.6206e-14 9.9793e-11 -1.13283e-7 0.000118727 -
0.00171684]';
            mu = '[7.06e-6, 0.00001038, 0.00001336, 0.00001606,
0.0000172, 0.00001769, 0.00001853, 0.00001911, 0.00002002, 0.00002081,
0.00002177, 0.00002294, 0.00002682, 0.0000303, 0.00003349, 0.00003643,
0.00003918, 0.00004177]';
```

```
            muT = '[100, 150, 200, 250, 273, 283, 300, 313, 333, 350,
373, 400, 500, 600, 700, 800, 900, 1000]';
            R_h = '287';
            check_h = '1';
        case {'H2O'}
            cp = '[4.0871e-10 -5.9806e-7 3.37478e-4 -8.56835e-2
1.23546e1].*1000';
            rho = '[-2.9681e-8 4.91535e-5 -3.25219e-2 9.18133
8.21792e1]';
            k = '[-2.98213e-11 5.47109e-8 -4.25128e-5 1.54729e-2 -
1.44067]';
            mu = '[0.001791, 0.001308, 0.001003, 0.0007977, 0.0006531,
0.0005471, 0.0004668, 0.0004044, 0.0003549, 0.000315, 0.0002822,
0.0001961, 0.0001494, 0.000121, 0.0001015, .0001]';
            muT = '[273.16, 283.15, 293.15, 303.15, 313.15, 323.15,
333.15, 343.15, 353.15, 363.15, 373.15, 413.15, 453.15, 493.15, 533.15,
1000]';
             case {'R134a'}    % These values are exact numbers because
we need to specify the exact number @ each internal energy value NOT in
polynomials format
             cp = '[1264 1293 1263 1262 1262; 1217 1290 1295 1294
1293;1171 1243 1322 1332 1331;1124 1197 1276 1364 1376;1078 1150 1229
1317 1418;1031 1104 1183 1271 1373;985.1 1057 1137 1225 1327;938.6 1011
1090 1179 1281;892.2 964.5 1044 1133 1235;845.8 918.1 997.7 1087
1190;799.4 871.6 951.3 1041 1144;769 825.2 905 994.5 1098;802.9 826 869
948.3 1052;842.3 857 882.4 925 995.9;881.1 891.6 909.3 937.9
982.8;918.4 926.2 939.5 960.5 992.7;954 960.1 970.3 986.5 1011;987.8
992.7 1001 1014 1033;1020 1024 1031 1041 1057;1051 1054 1060 1069
1082]';
```

```
rho = '[1403 1403 1404 1404 1405;36.99 560 1355 1356
1357;18.73 62.3 555.2 1306 1307;12.54 32.99 98.07 639.8 1254; 9.429
22.43 53.79 146.1 770.6; 7.553 16.99 37.05 82.47 208.3; 6.3 13.68 28.26
57.45 120.4; 5.403 11.44 22.84 44.08 84.68; 4.73 9.838 19.17 35.76
65.31; 4.206 8.627 16.51 30.08 53.14; 3.787 7.682 14.5 25.95 44.8;
3.413 6.923 12.93 22.83 38.72;3.051 6.196 11.59 20.37 34.1;2.777 5.621
10.46 18.25 30.33; 2.56 5.171 9.591 16.64 27.42; 2.383 4.806 8.893
15.37 25.18; 2.235 4.504 8.318 14.34 23.39; 2.109 4.247 7.834 13.47
21.92;2.001 4.026 7.419 12.74 20.68;1.906 3.833 7.058 12.1 19.61 ]';
k = '[0.1083    0.1084 0.1084 0.1085 0.1087; 0.09905 0.1021
0.1024 0.1025 0.1027; 0.08978 0.0931 0.09566 0.09606 0.09626; 0.0805
0.08408 0.08694 0.089 0.08944; 0.07123 0.07506 0.07822 0.08062 0.08216;
0.06195 0.06605 0.06951 0.07224 0.07416; 0.05268 0.05703 0.06079
0.06386 0.06615; 0.0434 0.04801 0.05207 0.05548 0.05815; 0.03412 0.039
0.04335 0.0471 0.05014; 0.02485 0.02998 0.03463 0.03872 0.04214;
0.01557 0.02096 0.02591 0.03033 0.03413; 0.009281 0.01195 0.01719
0.02195 0.02613;0.01169 0.01194 0.01234 0.01357 0.01813; 0.01378
0.01396 0.01425 0.01471 0.01539; 0.01558 0.01572 0.01594 0.0163
0.01682; 0.01712 0.01723 0.01742 0.0177 0.01812; 0.01845 0.01854
0.01869 0.01892 0.01927;0.01957 0.01964 0.01977 0.01997 0.02026;
0.02051 0.02057 0.02068 0.02086 0.02111; 0.02129 0.02135 0.02144
0.02159 0.02182]';
mu = '[0.0004305    0.0003462 0.0004317 0.0004328
0.0004345;0.0003913 0.0003438 0.0003411 0.000342 0.0003433;0.0003522
0.0003111 0.0002797 0.0002769 0.000278;0.000313 0.0002784 0.0002518
0.0002305 0.0002286;0.0002738 0.0002457 0.0002239 0.0002063 0.0001915;
0.0002347 0.0002129 0.000196 0.0001821 0.0001703;0.0001955 0.0001802
0.0001681 0.0001579 0.000149;0.0001564 0.0001475 0.0001402 0.0001337
0.0001277;0.0001172 0.0001148 0.0001123 0.0001095 0.0001065;0.00007803
```

```
0.00008212 0.00008439 0.00008534 0.00008522;0.00003887 0.00004941
0.00005649 0.00006114 0.00006396;0.000009796 0.00001671 0.00002859
0.00003695 0.00004269;0.00001082 0.0000109 0.00001104 0.00001275
0.00002143;0.00001179 0.00001185 0.00001196 0.00001212
0.00001239;0.00001271 0.00001276 0.00001284 0.00001297
0.00001318;0.00001359 0.00001363 0.00001369 0.0000138
0.00001398;0.00001442 0.00001446 0.00001451 0.00001461
0.00001476;0.00001522 0.00001525 0.0000153 0.00001539
0.00001552;0.00001599 0.00001602 0.00001607 0.00001614
0.00001626;0.00001674 0.00001676 0.0000168 0.00001687 0.00001698]';
            muT = '[6290 27012 47733 68455 89176 109898 130619 151341
172063 192784 213506 234227 254949 275671 296392 317114 337835 358557
379278 400000]';
            check_h = '2' ;
    end
     %%% Calculate entropy J/kg-K
    cp1 = str2num(cp);
    n=length(cp1);
    if n==1
        coef_s = [0 cp1];
    else
        kk = (n-1):-1:1;
        coef_s = [cp1(1:(n-1))./kk 0 cp1(n)];
    end
    s_a = ['[', num2str(coef_s(1:(end-1))), ']'];
    s_end  = num2str(coef_s(end));
    %%% Calculate enthalpy, J/kg
if check_h == 0 || check_h == 1
    icp = [str2num(cp)./(length(str2num(cp)):-1:1), 0];
```

```
    % For polyval,
    hin = strcat('[',num2str(icp),']');
    %%% Fluid properties for hot_balance,
   set_param(gcb, 'rho_h', rho);
   set_param(gcb, 'enthalpy_h', hin);
   set_param(gcb, 'cp_h',cp);
   set_param(gcb, 'mu_Input_h', muT);
   set_param(gcb, 'mu_Output_h', mu);
   set_param(gcb, 'k_h', k);
% set_param(gcb, 'entropy_h', s_a);
% set_param(gcb, 'entropy_end_h', s_end);
   set_param(gcb, 'R_h',R_h);
   set_param(gcb, 'fluid_check_h', check_h);
   else
        RG_cp = cp ;
        RG_rho = rho ;
        RG_k = k ;
        RG_mu = mu ;
        RG_muT = muT ;
   set_param(gcb, 'RG_rho_h', RG_rho);
% set_param(gcb, 'enthalpy_h', hin);
   set_param(gcb, 'RG_cp_h',RG_cp);
   set_param(gcb, 'RG_mu_Input_h', RG_muT);
   set_param(gcb, 'RG_mu_Output_h', RG_mu);
   set_param(gcb, 'RG_k_h', RG_k);
% set_param(gcb, 'entropy_h', s_a);
% set_param(gcb, 'entropy_end_h', s_end);
   set_param(gcb, 'RG_R_h',R_h);
   set_param(gcb, 'RG_fluid_check_h', check_h);
```

```
end
%%%%%%%%%%%%%%%%%% END HOT FLUID PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%% START HX MATERAL PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Get parameters,
    material = get_param(gcb,'material');
    %material = get_param(gcb,'material');
%%% Material properties,
    % Specific heat, J/kg/K
    % Density, kg/m^3
    % Thermal conductivity, W/m/K
    switch material
        case {'Stainless steel - 316'}
            cp = '502';
            rho = '8027';
            k = '16.26';
        case {'Aluminum'}
            cp = '896';
            rho = '2707';
            k = '220';
        case {'Copper'}
            cp = '380';
            rho = '8954';
            k = '386';
    end
    %%% HX parameters,
    set_param(gcb, 'cp_m', cp);
    set_param(gcb, 'rho_m', rho);
```

```
    set_param(gcb, 'k_m', k);
%%%%%%%%%%%%%%%%%%%% END HX MATERIAL PROPERTIES
%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Import geometry,
    cgeo = str2num(get_param(gcb,'cgeo'));
    hgeo = str2num(get_param(gcb,'hgeo'));
    length1 = str2num(get_param(gcb,'length'));
    vec = [cgeo hgeo length1];

ctype = get_param(gcb,'ctype');
    htype = get_param(gcb,'htype');
    fluids.ctype = ctype;
    fluids.htype = htype;



mdot1 =evalin('base', get_param(gcb,'mdot1'));
    mdot2 = str2num(get_param(gcb,'mdot2'));
    Tin1  = str2num(get_param(gcb,'Tin1'));
    Tin2  = str2num(get_param(gcb,'Tin2'));
    Q_load  = str2num(get_param(gcb,'load'));
    optimize = strcmp(get_param(gcb,'optimize'),'on');
    vec2 = [mdot1 mdot2 Tin1 Tin2 Q_load optimize];
    %HX_Size_Factor12=evalin('base', get_param(gcb,'HX_Size_Factor'));
    %%% Run HXsize function,
    hx = HXsize_v2(vec,fluids,vec2,material);
    if optimize>0.5
        Ac1_opt=num2str(hx.Ac1_opt);
        Ac2_opt=num2str(hx.Ac2_opt);
```

```
        set_param(gcb, 'Ac1_opt',Ac1_opt);
        set_param(gcb, 'Ac2_opt',Ac2_opt);
    end
%%% Volume of fluid, m^3
    volume_c = num2str(hx.vol1);
    volume_h = num2str(hx.vol2);
    volume_tot = num2str(hx.vol_HX);
    set_param(gcb, 'volume_c', volume_c);
    set_param(gcb, 'volume_h', volume_h);
    set_param(gcb, 'hx_total_volume', volume_tot);
%%% Surface area, m^2
    surf_area_c = hx.As1;
    surf_area_h = hx.As2;
    noverall = 0.9;
    dummy1 = num2str(surf_area_c*noverall/number);
    dummy2 = num2str(surf_area_h*noverall/number);
    set_param(gcb, 'noAs_c', dummy1);
    set_param(gcb, 'noAs_h', dummy2);
%%% Channel length, m
    length1 = linspace(0,length1,number+1);
    length1 = length1(2:end);
    set_param(gcb, 'spacing', mat2str(length1));
%%% Free flow area, m^2
    freeflow_c = num2str(hx.Ac1);
    freeflow_h = num2str(hx.Ac2) ;
    set_param(gcb, 'freeflow_c', freeflow_c);
    set_param(gcb, 'freeflow_h', freeflow_h);
%%% Hydraulic diameter, m
    Dh_c = num2str(hx.dh1);
```

```
    Dh_h = num2str(hx.dh2);
    set_param(gcb, 'Dh_c', Dh_c);
    set_param(gcb, 'Dh_h', Dh_h);
%%% Geometric channel dimension ratios,
    ratio_c = hx.ratios1;
 %   set_param(gcb, 'alpha_c', num2str(ratio_c(1)));
 %   set_param(gcb, 'delta_c', num2str(ratio_c(2)));
 %   set_param(gcb, 'gamma_c', num2str(ratio_c(3)));
    ratio_h = hx.ratios2;
 %   set_param(gcb, 'alpha_h', num2str(ratio_h(1)));
 %   set_param(gcb, 'delta_h', num2str(ratio_h(2)));
 %   set_param(gcb, 'gamma_h', num2str(ratio_h(3)));
%%% Heat exchanger mass, kg
    mass = hx.weight_kg  ;

%   set_param(gcb, 'hx_total_mass', num2str(mass));
 %   set_param(gcb, 'hx_CV_mass', num2str(mass/number));
%%% Heat exchanger area / thickness, m
    Aht = (hx.Aht1 + hx.Aht2)/2;
    t = (cgeo(5) + hgeo(5))/2/1000;
    set_param(gcb, 'Aht_t', num2str(Aht/t));
 %%% Clear workspace,
    clear cytpe cp rho k mu muT icp hin cp1 n s_a s_end htype material
    clear cgeo hgeo length1 vec fluids mdot1 mdot2 Tin1 Tin2 Q_load
vec2
    clear number hx volume_c volume_h surf_area_c surf_area_h noverall
    clear dummy1 dummy2 freeflow_c freeflow_h Dh_c Dh_h ratio_c ratio_h
    clear mass Aht t Ac1_opt Ac2_opt
    %    Feb 7, 2017
```

Finally, this code is specialized to plot results versus time.

```
% Fin plate Heat Exhanger plots
ti= 3; % Start time
whichElements = ti/SampleTime:1:length(time); % Which elements of the
arrays to plot
timeUnit = 'Sec'; % Unit for time axis
timeScale = 1; % Divisor for time axis
theTime = (time(whichElements)-ti)/timeScale; % The scaled time array
xLimits = [0, theTime(end)]; % The limits for time axis
RShift = 459.67; % Difference of Rankin and Fahrenheit
K2R = 9/5; % Conversion from K to R
lbm2kg = 0.4536; % Conversion from lbm to kg
kPa2psi = 14.7/101.325; % Conversion from kPa to psi
 % Set plotting defaults.
set(0,'DefaultFigureWindowStyle','docked'); % Undock all figures.
set(0,'DefaultFigureColor','w'); % Set default figure background color.
set(0,'DefaultLineLineWidth',4); % Set default line size.
set(0,'DefaultAxesFontSize',20); % Set default axes font size.
set(0,'DefaultTextFontSize',20); % Set default text font size.
set(0,'DefaultLineMarkerSize',12); % Set default marker size.
set(0,'DefaultAxesFontWeight','bold'); % Set the default axes font to
bold.
set(0,'DefaultTextFontWeight','bold'); % Set the default text font to
bold..
set(0,'DefaultAxesColorOrder',[0 0 0;0 0 0;0 0 1])
 %%
figure('Name','Pressure')
plot(theTime,Pressure(whichElements))
```

```
ylabel('Pressure [ kpa]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title (' Pressure vs.time ')
grid on
%%
figure('Name','Enthalpy')
plot(theTime,ENTHALPY(whichElements))
ylabel('Enthalpy [ kJ/kg]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title (' Enthalpy vs.time ')
grid on
%%
figure('Name','Kerosene Temperature')
plot(theTime,KerTempOut(whichElements))
ylabel('Temperature [\circK]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title (' KerTempOut vs.time ')
grid on
%%
figure('Name','R134a Temperature ')
plot(theTime,RefTempOut(whichElements))
ylabel('Temperature [\circK]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title (' RefTempOut vs.time ')
grid on
```

```
%%
figure('Name','Heat exchanger temperature')
plot(theTime,Thx(whichElements))
ylabel('Heat exchanger temperature [\circK]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title ('Heat exchanger temperature vs.time ')
grid on
%%
figure('Name','kerosene heat transfer coefficient')
plot(theTime,KerHeatCoef(whichElements))
ylabel('h_c [w/m^2-k]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title (' kerosene heat transfer coefficient vs.time ')
grid on
 %%
figure ('Name' ,'Two phase flow heat transfer coefficient')
plot (theTime, h (whichElements))
ylabel('h_TP [w/m^2-k]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title ('Two phase flow heat transfer coefficient vs. time ')
grid on
%%
figure('Name','X')
plot(theTime,x(whichElements))
ylabel('Quality');
xlabel(['Time [' timeUnit ']']);
```

```
xlim(xLimits);
title (' X vs.time ')
grid on
%%
figure('Name','Qc')
plot(theTime,Qc(whichElements))
ylabel('Qc [w]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title ('Heat transferred to kerosene vs.time ')
grid on
%%
figure('Name','Qh')
plot(theTime,Qh(whichElements))
ylabel('Qh [w]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title ('Heat transferred from R134a vs.time ')
grid on


%%
figure('Name','Total Entropy generated')
plot(theTime,Sgen(whichElements))
ylabel('Total Entropy generated [w/k]');
xlabel(['Time [' timeUnit ']']);
xlim(xLimits);
title ('Total Entropy generated vs.time ')
grid on
```

```
%%
figure('Name','R134a Enthalpy')
plot(theTime,RefHOut(whichElements))
ylabel(' R134a Enthalpy [J/kg]');
xlabel(['Time  [' timeUnit ']']);
xlim(xLimits);
title ('R134a Enthalpy vs.time ')
grid on
%%
figure('Name','R134a Internal Energy')
plot(theTime,RefHOut1(whichElements))
ylabel(' R134a Internal Energy [J/kg]');
xlabel(['Time  [' timeUnit ']']);
xlim(xLimits);
title ('R134a Internal Energy vs.time ')
grid on
```

APPENDIX C

To verify the results of the Simulink model, EES program has been used to check equations employed in the model. Below, EES script shows these equations.

```
$unitsystem SI K pa J Rad mass
$Tabstops 0.5 2 in

                                                    " verifications for the Simulink model "

" Cold balance - friction factor "
alpha = 1.20482
delta = 0.015748
gamma = 0.1
m_dot_c = 25 [ kg/s]                                " mass flow rate "
D = 0.00098833 [m]                                  " hydraulic diameter "
mu = 0.003567 [ pa - sec]                           " dynamic viscosity "
A_c = 0.001 [ m^2 ]                                 " cross sectional area "
Re = m_dot_c * D / (mu * A_c)                       " Reynolds number "
 f = 9.6243 * Re^(- 0.7422) * alpha ^(- 0.1856) * delta ^( 0.3053) * gamma^(-0.2659) * ( 1+ 7.669 *
10^(-8) * Re ^ (4.429) * alpha ^(0.92) * delta^(3.767) * gamma^(0.236))^(0.1)       " friction factor "

" Cold balance - pressure drop "
L =  0.07 [m]                                       " length "
rho = 820 [kg/m^3]                                  " density "
press_drop = f * L / ( 2 * D * rho ) * (m_dot_c /A_c)^2     " pressure drop in (pa) "

" Cold balance - energy equation "
Cp = 2010 [J/kg-k]                                  " specific heat for kerosene "
vol_c = 7.0001e-05 [m^3]                            " volume of kerosene "
```

Q_dot = 14607.48204086 [w] " heat trabsfer "

T_in = 300 [k] " temperature in "

h_in = T_in * Cp " enthalpy in "

T_out = 300.3 [k] " temperature out "

h_out = T_out * Cp " enthalpy out "

dE_dt = Q_dot + m_dot_c * (h_in - h_out) " energy balance "

dT_dt = dE_dt / (Cp * rho * vol_c) " temperature change with respect to time "

" Cold balance - entropy equation "

dS_dt = rho * vol_c * Cp * dT_dt / T_out " entropy change with respect to time "

" Heat transfer coefficient - Cold "

k = 0.15 [w/m-k] " thermal conductivity "

Pr = mu * Cp / k " Prandtl number "

zeta = (0.79 * ln (Re) - 1.64)^(-2) " zeta parameter in Gnielinski correlation "

Z = zeta / 8 " zeta devided by 8"

Nu = (Z* Pr * (Re - 1000)) / ((12.7 * **sqrt**(Z) * (Pr ^(2/3) -1)) +1.07) " nusselt number "

h = Nu * k / D " convective heat transfer coefficient "

" Entropy generation - Cold "

S_gen = dS_dt - m_dot_c * Cp * ln (T_out / T_in) " entropy generated "

{ ======================} { ======================} { ======================}

{" Hot balance - friction factor "

alpha = 0.078302

delta = 0.015748

gamma = 0.12048

m_dot_h = 0.5 [kg/sec] " mass flow rate "

D = 0.0015099 [m] " hydraulic diameter "

mu = 1.30327e-5 [pa -sec] " dynamic viscosity for R-134a"

A_c = 0.0018471 [m^2] " cross sectional area "

Re = m_dot_h * D / (mu * A_c) " Reynolds number "

f = 9.6243 * Re^(- 0.7422) * alpha ^(- 0.1856) * delta ^(0.3053) * gamma^(-0.2659) * (1+ 7.669 *

10^(-8) * Re ^ (4.429) * alpha ^(0.92) * delta^(3.767) * gamma^(0.236))^(0.1) "

friction factor "

" Hot balance - pressure drop "

L = 0.07 [m] " length "

rho = 22.98 [kg/m^3] " density of R-134a "

press_drop = f * L / (2 * D * rho) * (m_dot_h/A_c)^2 " pressure drop in pa "

" Hot balance - energy equation "

Qdot = 14607.48 [w] " heat transfer "

h_in = 300000 [J/kg] " enthalpy in "

h_out = 270785.0359 [J/kg] " enthalpy out "

p_out = 500 [kpa] " pressure out "

v = 0.05184 [m^3 / kg] " specific volume "

dE_dt = -Qdot +(m_dot_h * (h_in - h_out)) " energy balance "

m = 0.002971 [kg] " mass of R-134a "

```
du_dt  =   dE_dt /  m                                   " internal energy change with
respect to time "
T_out = 331.5 [K]                                       " temperature out "
u = 270759.118 [ J/kg]                                  " internal energy "
h_check = u + p_out * v                                 " h_check should equal to
h_out "

" Hot balance - entropy equation "
dS_dt = m * du_dt / T_out                               " entropy change with respect
to time "

{ ======================} { ======================} { ======================}

" Heat transfer coefficient - Hot "

" liquid phase "
mu_l = 0.0001857 [ pa -sec]                             " dynamic viscosity @ liquid
phase "
Cp_l = 1440 [ J/kg-k]                                   " specific heat @ liquid phase "
k_l = 0.08144 [ w/m-k]                                  " thermal conductivity @ liquid
phase "
Pr_l = mu_l * Cp_l /k_l                                 " Prandtl number @ liquid
phase "
Re_l = m_dot_h * D / ( mu_l * A_c)                      " Reynolds number @ liquid
phase "
zeta_l = (0.79 * ln (Re_l) - 1.64)^(-2)                 " zeta parameter in Gnielinski
correlation "
Z_l = zeta_l / 8                                        " zeta devided by 8"
```

```
Nu_l = ( Z_l* Pr_l * (Re_l - 1000)) / ((12.7 * sqrt( Z_l) * (Pr_l ^(2/3) -1)) +1.07)  " Nusselt number
@ liquid phase "
h_l = Nu_l * k_l / D                                    " heat transfer coefficient @
liquid phase "

" vapor phase "
mu_v = 1.21342e-05                                      " dynamic viscosity @ vapor
phase "
Cp_v = 1056 [ J/kg-k]                                   " specific heat @ vapor phase
"
k_v = 0.01493 [w/m-k]                                   " thermal conductivity @ vapor
phase "
Pr_v = mu_v * Cp_v / k_v                                " Prandtl number @ vapor
phase "
Re_v = m_dot_h * D / ( mu_v * A_c)                      " Reynolds number @ vapor
phase "
zeta_v = (0.79 * ln (Re_v) - 1.64)^(-2)                 " zeta parameter in Gnielinski
correlation "
Z_v = zeta_v / 8                                        " zeta devided by 8"
Nu_v = ( Z_v* Pr_v * (Re_v - 1000)) / ((12.7 * sqrt( Z_v) * (Pr_v ^(2/3) -1)) +1.07)     " Nusselt
number @ vapor phase "
h_v = Nu_v * k_v / D                                    " heat transfer coefficient @
vapor phase "

" two phase "
F_fl = 1.63                                             " fluid dependant parameter "
x = 0.9949                                              " quality "
rho_g = 38.3  [ kg/m^3]                                 " vapor density  "
```

```
rho_l =  1182 [kg/m^3]                                   " liquid density "
Co = (rho_g / rho_l)^(0.5) * ((1-x)/x)^(0.8)             " convective number "
q = 4023 [w/m^2]                                         " heat transfer per unit area "
u_lg = 174349.4568 [ J/kg]                               " enthalpy for evaporation "
G = m_dot_h /A_c                                         " mass flux"
Bo = q / ( G * u_lg)                                     " boiling number "
h_r =  (1.183744 * Co^(-0.3) + 225.5474 * BO^(2.8) * F_fl) * (1 - x)^(0.003) * h_l     " heat transfer
coefficient for two phase "

" Entropy generation - Hot "
s_in = 1057 [ J/kg-k]                                    " entropy in "
s_out = 964.7 [ J/kg-k]                                  " entropy out "
S_gen  = dS_dt - m_dot_h * (s_in - s_out)                " entropy generated "}
```

Solution

Main

Unit Settings: SI K Pa J mass rad

α = 0.0783	A_c = 0.001847 [m^2]	Bo = 0.00008524	Co = 0.002649	Cp_l = 1440 [J/kg-k]
Cp_v = 1056 [J/kg-k]	D = 0.00151 [m]	δ = 0.01575	dE_{dt} = 0.00205	dS_{dt} = 0.000006184
du_{dt} = 0.69	f = 0.01054	F_{fl} = 1.63	G = 270.7	γ = 0.1205
h_{check} = 270785	h_{in} = 300000 [J/kg]	h_l = 588.2	h_{out} = 270785 [J/kg]	h_r = 4064
h_v = 793.8	k_l = 0.08144 [w/m-k]	k_v = 0.01493 [W/m-K]	L = 0.07 [m]	m = 0.002971 [kg]
μ = 0.00001303 [pa-sec]	μ_l = 0.0001857 [pa-sec]	μ_v = 0.00001213	$\dot{m}_h$ = 0.5 [kg/sec]	ν_l = 10.9
ν_v = 80.28	$press_{drop}$ = 779	Pr_l = 3.283	Pr_v = 0.8583	p_{out} = 500 [kPa]
q = 4023 [W/m^2]	Qdot = 14607 [W]	Re = 31361	Re_l = 2201	Re_v = 33683
ρ = 22.98 [kg/m^3]	ρ_g = 38.3 [kg/m^3]	ρ_l = 1182 [kg/m^3]	S_{gen} = -46.15	s_{in} = 1057 [J/kg-k]
s_{out} = 964.7 [J/kg-k]	T_{out} = 331.5 [K]	u = 270759 [J/kg]	u_{lg} = 174349	v = 0.05184 [m^3/kg]
x = 0.9949	ζ_l = 0.05072	ζ_v = 0.02299	Z_l = 0.00634	Z_v = 0.002873

Figure 49: Results of hot flow (R-134a)

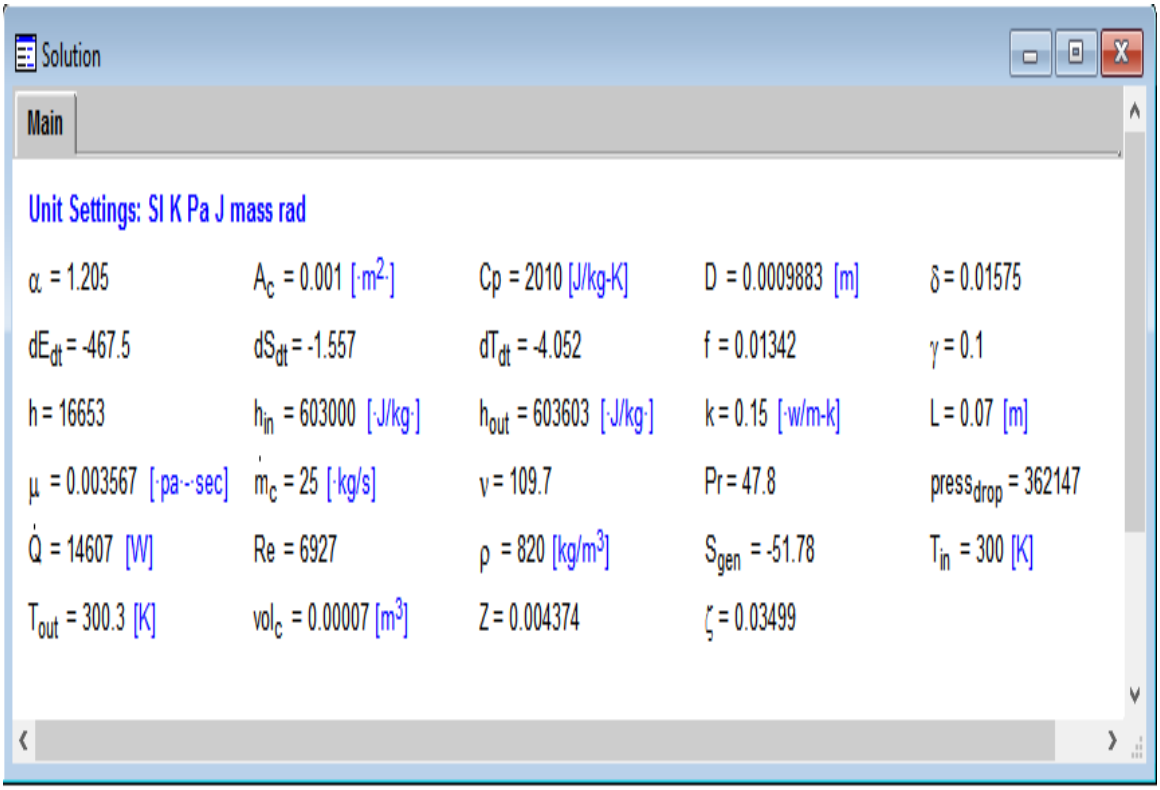


Figure 50: Results of cold flow (kerosene)

REFERENCES

[1] S. S. Garba, “Cranfield University,” *Applied Sciences*. .

[2] H. D. Kim, J. L. Felder, M. T. Tong, and M. J. Armstrong, “Revolutionary Aeropropulsion Concept for Sustainable Aviation: Turboelectric Distributed Propulsion,” *21st Int. Symp. Air Breath. Engines*, pp. 1–12, 2013.

[3] H. Kim, “Distributed Propulsion Vehicles,” *Int. Congr. Aeronaut. Sci.*, pp. 1–11, 2010.

[4] R. Radebaugh, “Cryocoolers for aircraft superconducting generators and motors,” *AIP Conf. Proc.*, vol. 1434, no. 57, pp. 171–182, 2012.

[5] J. Felder, H. Kim, and G. Brown, “Turboelectric Distributed Propulsion Engine Cycle Analysis for Hybrid-Wing-Body Aircraft,” *47th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo.*, pp. 1–25, 2009.

[6] E. Jones, D. Doroni-dawes, and D. Larkin, “NASA N3-X Preliminary Design Study : Final Technical Report,” vol. 1, no. June, pp. 1–92, 2016.

[7] C. M. Lewandowski, “Turboelectric Distributed Propulsion in a Hybrid Wing Body Aircraft,” *NASA Tech. Rep. 1 ISABE-20*, pp. 1–20, 2011.

[8] G. Brown, “Weights and Efficiencies of Electric Components of a Turboelectric Aircraft Propulsion System,” *49th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo.*, no. January, 2011.

[9] J. Palmer and E. Shehab, “Cryogenic Systems Study for Turbo-Electric Distributed Propulsion Aircraft Solution.”

[10] F. Berg, J. Palmer, L. Bertola, P. Miller, and G. Dodds, “Cryogenic system options for a superconducting aircraft propulsion system,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 101, no. December 2015, p. 12085, 2015.

[11] E. Jones, D. Doroni-dawes, and D. Larkin, “NASA N3-X Preliminary Design Study : Final Technical Report,” 2016.

[12] Boeing, “Airplane characteristics for airport planning - 747,” *Boeing Commerical Airplanes*, no. November, p. 126, 2011.

[13] “Enhanced ECS Generator Models in an Integrated Air Vehicle Platform Final.” .

[14] G. J. Michna, A. M. Jacobi, and R. L. Burton, “Friction Factor and Heat Transfer Performance of an Offset-Strip Fin Array at Air-Side Reynolds Numbers to 100,000,” *Int. Refrig. Air Cond. Conf.*, vol. 129, no. September 2007, pp. 0–8, 2006.

[15] Y.-Y. Yan and T.-F. Lin, “Evaporation heat transfer and pressure drop of refrigerant R-134a in a plate heat exchanger,” *J. Heat Transf.*, vol. 121, no. 1, pp. 118–127, 1999.

[16] M. Teruel, “Rectangular offset strip-fin heat exchanger lumped parameters dynamic model,” *Brazilian Symp. Aerosp. Eng. Appl.*, 2009.

[17] P. Yuan, G. B. Jiang, Y. L. He, and W. Q. Tao, “Performance simulation of a two-phase flow distributor for plate-fin heat exchanger,” *Appl. Therm. Eng.*, vol. 99,

pp. 1236–1245, 2016.

[18] S. Ben Saad, P. Clément, J. F. Fourmigué, C. Gentric, and J. P. Leclerc, “Single phase pressure drop and two-phase distribution in an offset strip fin compact heat exchanger,” *Appl. Therm. Eng.*, vol. 49, pp. 99–105, 2012.

[19] J. H. Park and Y. S. Kim, “Evaporation heat transfer and pressure drop characteristics of R-134a in the oblong shell and plate heat exchanger,” *KSME Int. J.*, vol. 18, no. 12, pp. 2284–2293, 2004.

[20] M. A. X. Agitators and K. Benefits-, “Process Solutions International,” no. 866, pp. 4–7.

[21] P. Talukdar, “Plate-Fin Heat Exchanger.”

[22] F.Mayinger, “Classification nad Applications of Two - Phase flow Heat Exchangers.pdf.”

[23] S. RK and D. Sekuli´c, *Fundamentals of heat exchanger design*. 2003.

[24] V. Donowski and S. Kandlikar, “Correlating evaporation heat transfer coefficient of refrigerant R-134a in a plate heat exchanger,” *Proc. Boil. 2000 Phenom. …*, pp. 1–18, 2000.

[25] “an improved correlation for predicting two phase flow.pdf.” .

[26] S. G. Kandlikar, “Boiling heat transfer with binary mixtures: Part II - Flow boiling in plain tubes,” *Journal of Heat Transfer-Transactions of the Asme*, vol. 120. pp. 388–394, 1998.

[27] R. Roberts and J. Doty, “Implementation of a transient exergy analysis for a plate–

fin heat exchanger," *Int. J. Exergy*, vol. 16, no. 1, pp. 109–126, 2015.

[28] K. Pottler, C. M. Sippel, A. Beck, and J. Fricke, "Heat Transfer and Pressure Drop Correlations for Offest Strip Fins Usable for Solar Air Heating Collectors," no. June, 2017.

[29] R. Cicchitti, A. Lombaradi, C. Silversti, M. Soldaini, G., and Zavattarlli, "Two-Phase Cooling Experiments- Pressure Drop, Heat Transfer, and Burnout Measurement," vol. 7, no. 6, pp. 407–425, 1960.