Thermal Management System of a Drone Pod System

Author: Charlie Nitschelm

# **Introduction**

The Reaper fitted with a Seaspray 7500E Drone Pod needs an efficient thermal management system to maintain a temperature of 20 °C during its entire flight. The pod will experience outside environments as cold as -20 °C during the peak of its flight, and 50 °C during takeoff. Specifically, given these conditions, what is the best-known refrigerant to use in the drone pods thermal management system that is most efficient for the conditionals it will experience, its environmental impact, safety handling and cost considerations.

# **Refrigerant Selection**

To narrow down the selection of the desired refrigerant to three top choices, a performance analysis needs to happen on several widely used refrigerants. Once an array of refrigerants are selected for their particular performance in the Seaspray Drone Pod, other considerations can be made regarding to its environmental impact, safety handling and overall cost to determine the most desired refrigerant for the drone pod thermal management system.

## **Performance Analysis**

MATLAB and CoolProps is utilized to numerically calculate the coefficient of performance (COP) of any refrigerant that could be used as the working fluid of the drone pod thermal management system. The figure below details the differences in COP of various types of refrigerants used today. The graph tests five different types of refrigerants: R-Refrigerants like R410a, water and heavy water, isobutane and isobutene, MDM and cyclohexane, and lastly acetone and ammonia.

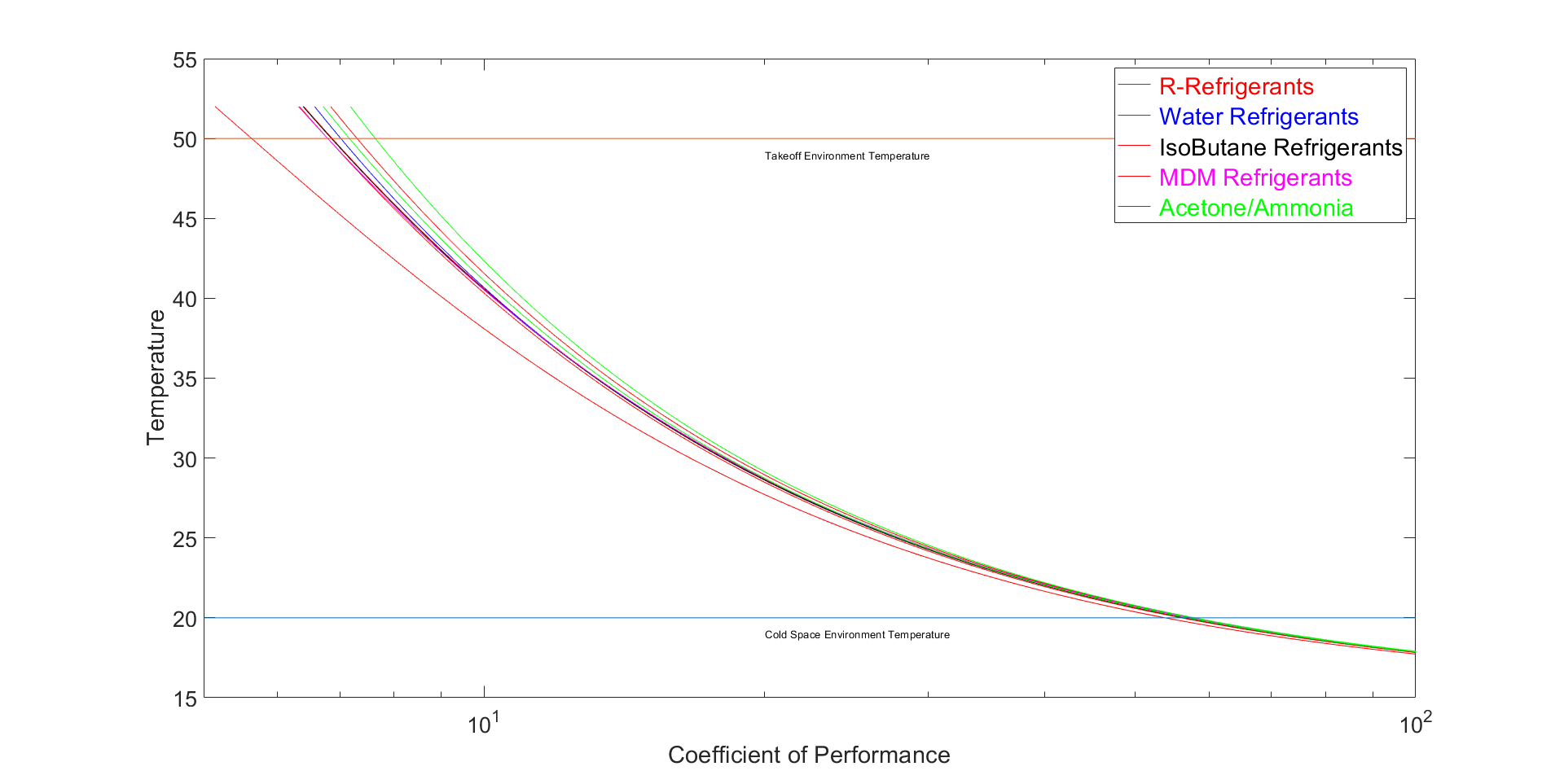


Figure 1- Coefficient of Performance of Several Refrigerants

The top four refrigerants with the highest COP with respect to the thermal management system of the drone pod are acetone with a COP of 7.88, R123 with a COP of 7.55, ammonia with a COP of 7.40, and lastly R290 with a COP of 6.87.

## **Environment, Safety and Cost Considerations**

The analysis above provides us with the best refrigerants for the thermal management system of a drone pod in respect to its COP. When deciding on the desired refrigerant, environment, safety handling, and overall cost must be considered. Very low to zero environmental harm is desired for the refrigerant. The use of the refrigerant must be safe when the refrigerant is handled correctly. The cost of the refrigerant must be considered, as the price of these drone pods are important to their final development and upkeep. Acetone, the best refrigerant with respect to its COP, has risks that need to be considered before accepting it as the desired refrigerant overall. Although it occurs naturally in the environment through decomposing vegetation and volcanic gases, the majority of the chemical released into the atmosphere is created from industrial purposes. It is flammable at room temperature (20 °C) and is a risk to the oxygen depletion in aquatic systems due to microbial consumption. It also has health risks to people when it is not handled perfectly [1]. R123, the second most efficient refrigerant analyzed, has little to zero effect on the environment but can be toxic to people when applied to the body by accident [2]. Ammonia is the third most efficient refrigerant for the system, but it has huge risks to the environment and people that is well understood. R290 is the fourth most efficient refrigerant, and has little to zero impact on the environment and is non-toxic to people when handling [4]. The required mass flow rate of the working fluid can also be predicted to understand the amount of the refrigerant that needs to move through the system per second. The figure below shows that amount of mass flow rate the thermal management system requires for a given outside temperature.

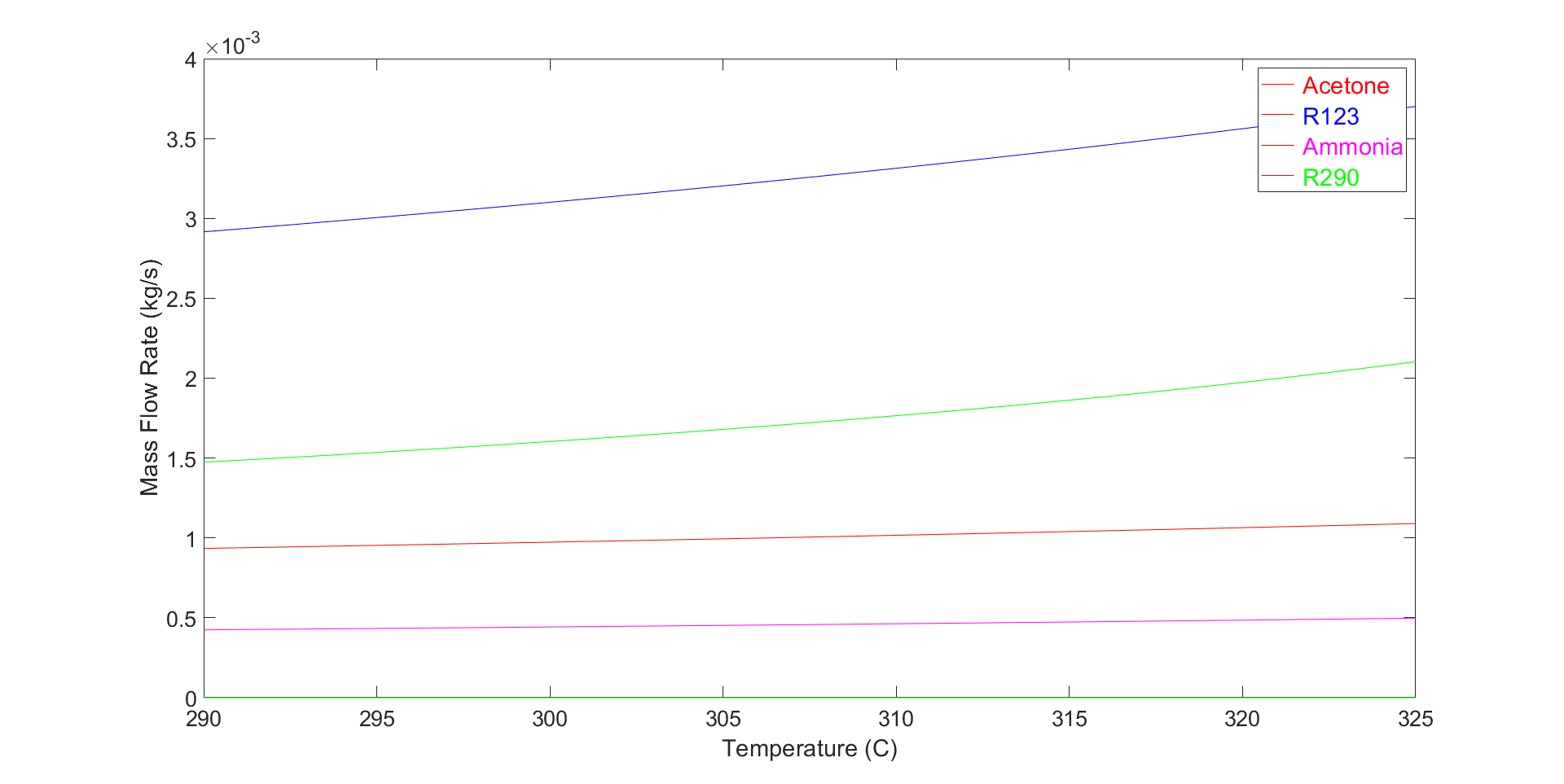


Figure 2- Mass Flow Rate of Most Efficient Refrigerants

The final consideration before a desired refrigerant is the monetary cost of the working fluid. Acetone, R123, ammonia and R290 has a price per kilogram of $20.00, $3.00, $8.00 and $7.00, respectively [4] [5] [6]. This information on environmental impact, safety handling and cost considerations yields a desired refrigerant for the thermal management system of the drone pod. Acetone, the most efficient refrigerant, has huge risks to people and the environment while also being the most expensive. R123, the second most efficient refrigerant, has nearly zero impact to the environment but can be toxic to people when handling. The third most efficient refrigerant, ammonia, has huge risk to the environment and people. With the considerations of efficiency, mass flow rate, environmental impact, safety handling and cost, R290 is clearly the most desired refrigerant for its environmental impact, safety handling and cost while also having a high COP and low mass flow rate compared to many other refrigerants.

# **Various Environmental Temperatures on the Refrigeration Cycle**

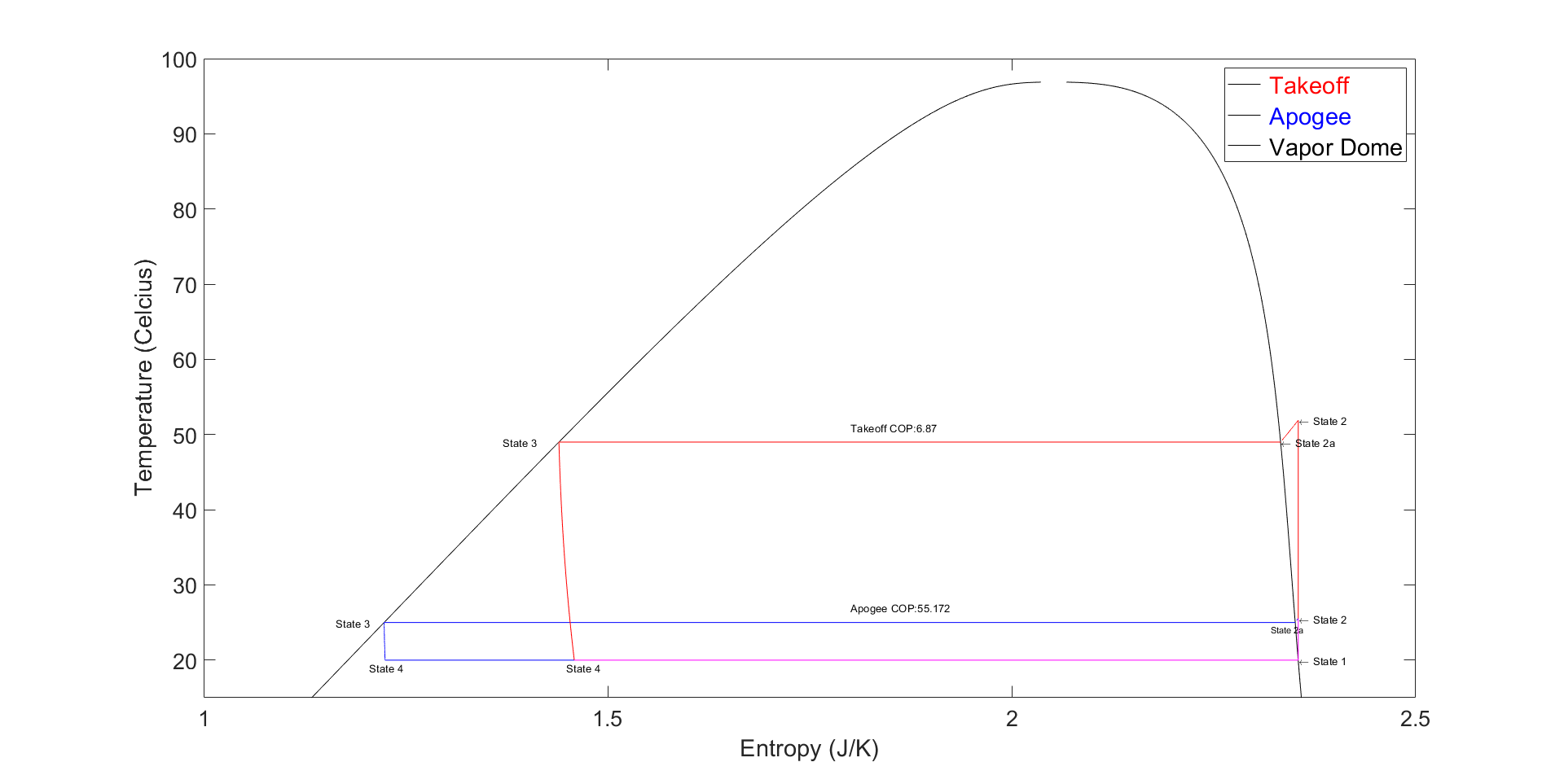
R290, the fourth most efficient refrigerant and third least required mass flow rate, is non-toxic to people, thermally stable, and has little to zero effect on the environment. Now that the desired refrigerant, R290, has been chosen, a more in depth look of the cycle can be brought forward. Below are the T-s, P-h and P-v diagrams of the R290 refrigeration cycle with the expected takeoff and apogee outside temperatures. 

Figure 3- T-s Refrigeration Cycle of R290

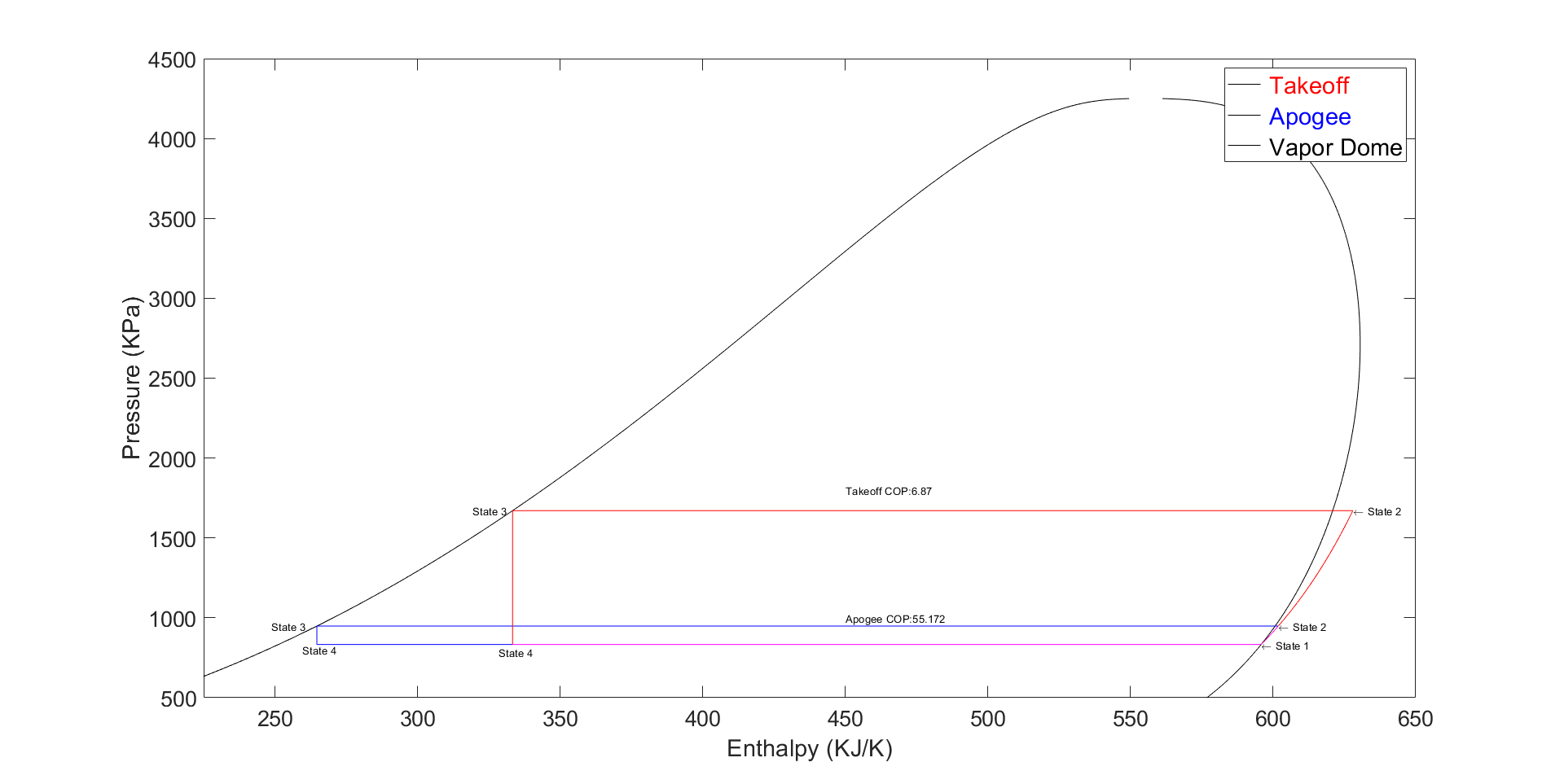


Figure 4- P-h Refrigeration Cycle of R290

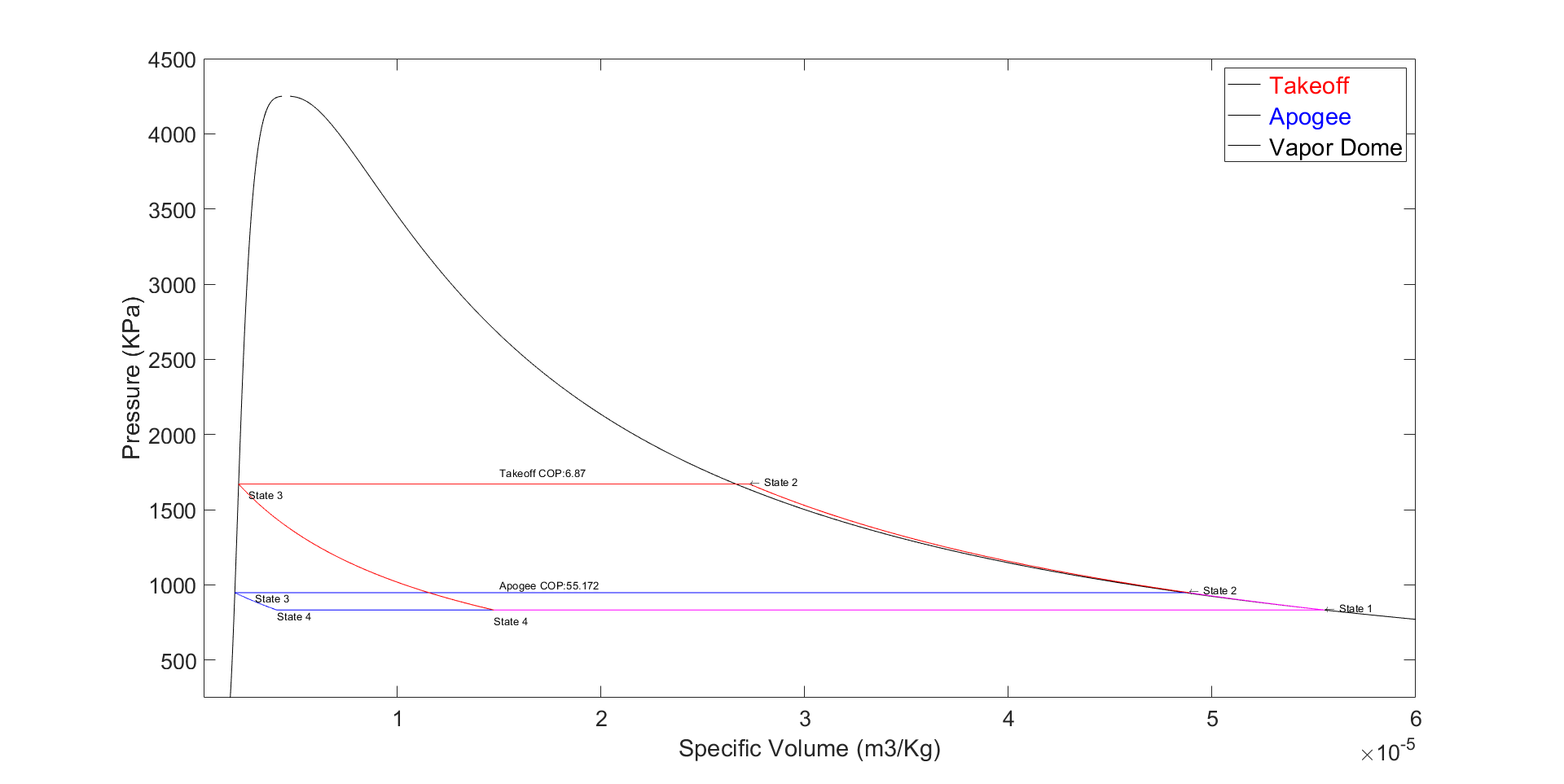


Figure 5- P-v Refrigeration Cycle of R290

As the Drone Pod is taking off, it can encounter an outside temperature of almost 50 degrees Celsius. With that huge difference in outside temperature to a desired temperature of 20 degrees Celsius, a COP is calculated to be 6.67. However, when the Drone Pod rises to an altitude that causes the outside temperature to fall below the desired temperature inside the Drone Pod, the Drone Pod is assumed to be perfectly insulated. This then creates the outside temperature to just be the change in temperature for the condenser, which was selected to be 5 Kelvin. That COP is much higher at 55 because of the small temperature variation of the entire refrigeration cycle. If the Drone Pod was considered not perfectly insulated, a thermal management system must be designed to switch to a heat pump when the outside temperature falls below the desired internal temperature of the pod.

# **Final Recommendation**

This report investigated the effect of different refrigerants applied to a thermal management system of the Seaspray Drone Pod with respect to its calculated COP. The best refrigerants for the system could then be investigated further for its environmental impact, safety handling and cost. The refrigerant is desired to have little to zero impact to the environment, poses little risk to overall handling, and minimal material cost. R290, the fourth most efficient refrigerant analyzed, was then selected as the desired refrigerant for the system as it is very efficient comparatively, low mass flow rate, safe to the environment, poses little to no threat to people, and has a very low material cost. If the requirements and preferences of the selected refrigerant changes, then another analysis could be required to determine the best refrigerant for the given conditions.

# **References**

[1] “Risks and Benefits.” *Acetone*, badacetone.weebly.com/risks-and-benefits.html.

[2] Refrigerants, National. “R123 Safety Data Sheet.” *SDS*, www.refrigerants.com/pdf/SDS%20R123.pdf.

[3] “R290 (CARE 40) Propane.” *Linde Gas*, www.linde-gas.com/en/products\_and\_supply/refrigerants/natural\_refrigerants/r290\_propane/index.html.

[4] “Acetone | Price | per Kg | USD.” *PharmaCompass.com*, www.pharmacompass.com/price/acetone.

[5] “Buy Chemicals 99.9% Purity Cyclohexane 110-82-7 - Buy Cyclohexane,Cyclohexane Price,Cas:110-82-7 Product on Alibaba.com.” *Www.alibaba.com*, www.alibaba.com/product-detail/buy-chemicals-99-9-purity-Cyclohexane\_60579895295.html?spm=a2700.7724838.2017115.51.4424711fe06nk3.

[6] “Refrigerant Gas Cyclohexane R134a - Buy Refrigerant Hfc 134a Good,Hfc R134a Refrigerant,Hfc 134a Refrigerant Product on Alibaba.com.” *Www.alibaba.com*, www.alibaba.com/product-detail/refrigerant-gas-cyclohexane-R134a\_1384971567.html.

# **Appendix**

## **MATLAB Code**

The code below utilized a single script with desired for loops to create the information necessary to write this report. Functions could have been used, but it would not have made debugging as easy, and was not necessary for the structure of this code.

clear all; close all

clc;

%%%%%%%%%%%%%

% COP vs Outside Temperature for 3 Different Refrigerants

%%%%%%%%%%%%%

Working\_Fluid\_1 = {'R410a'};

Working\_Fluid\_2 = {'R290'};

Working\_Fluid\_3 = {'Ammonia'};

Working\_Fluid\_4 = {'Water'};

Working\_Fluid\_5 = {'R22'};

Working\_Fluid\_6 = {'R114'};

Working\_Fluid\_7 = {'IsoButane'};

Working\_Fluid\_8 = {'MDM'};

Working\_Fluid\_9 = {'R12'};

Working\_Fluid\_10 = {'R143a'};

Working\_Fluid\_11 = {'R123'};

Working\_Fluid\_12 = {'Acetone'};

Working\_Fluid\_13 = {'R404a'};

Working\_Fluid\_14 = {'R125'};

Working\_Fluid\_15 = {'R134a'};

Working\_Fluids = [Working\_Fluid\_1,Working\_Fluid\_2,Working\_Fluid\_3,Working\_Fluid\_4,Working\_Fluid\_5,Working\_Fluid\_6,Working\_Fluid\_7,...

Working\_Fluid\_8,Working\_Fluid\_9,Working\_Fluid\_10,Working\_Fluid\_11,Working\_Fluid\_12,Working\_Fluid\_13,Working\_Fluid\_14,Working\_Fluid\_15];

Outside\_Temperature = linspace(273 + 18 - 1, 273 + 50 + 2, 1000);

COP\_R410a = zeros(1000);

COP\_R290 = zeros(1000);

COP\_Ammonia = zeros(1000);

COP\_Water = zeros(1000);

COP\_R22 = zeros(1000);

COP\_R114 = zeros(1000);

COP\_IsoButane = zeros(1000);

COP\_MDM = zeros(1000);

COP\_R12 = zeros(1000);

COP\_R143a = zeros(1000);

COP\_R123 = zeros(1000);

COP\_Acetone = zeros(1000);

COP\_R404a = zeros(1000);

COP\_R125 = zeros(1000);

COP\_R134a = zeros(1000);

MFR\_R410a = zeros(1000);

MFR\_R290 = zeros(1000);

MFR\_Ammonia = zeros(1000);

MFR\_Water = zeros(1000);

MFR\_R22 = zeros(1000);

MFR\_R114 = zeros(1000);

MFR\_IsoButane = zeros(1000);

MFR\_MDM = zeros(1000);

MFR\_R12 = zeros(1000);

MFR\_R143a = zeros(1000);

MFR\_R123 = zeros(1000);

MFR\_Acetone = zeros(1000);

MFR\_R404a = zeros(1000);

MFR\_R125 = zeros(1000);

MFR\_R134a = zeros(1000);

j=1;

for i = Working\_Fluids

k=1;

for temps = Outside\_Temperature

Working\_Fluid\_1 = Working\_Fluids{j};

%%%%%%%%%%%%%

% Constants for the Refridgeration Cycle for Durham, NH

%%%%%%%%%%%%%

Cold\_Space = 273+20; % Kelvin - Temperature of Cold Space of Pod

%%%%%%%%%%%%%

% Location 1 - Between Evaporator and Compressor

%%%%%%%%%%%%%

% Finding all values of the fluid at location 1

Q\_1 = 1; % Saturated Vapor

T\_1 = Cold\_Space; % Temperature from 4 to 1

P\_1 = CoolProp.PropsSI('P', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

H\_1 = CoolProp.PropsSI('H', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

U\_1 = CoolProp.PropsSI('U', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

S\_1 = CoolProp.PropsSI('S', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

V\_1 = 1/CoolProp.PropsSI('D', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 3 - Between Condensor and Expansion Valve - Saturated Liquid

%%%%%%%%%%%%%

% Finding all values of the fluid at location 3

Q\_3 = 0; % Saturated Liquid

T\_3\_Summer = temps + 5; % Temperature from 4 to 1

P\_3\_Summer = CoolProp.PropsSI('P', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

H\_3\_Summer = CoolProp.PropsSI('H', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

U\_3\_Summer = CoolProp.PropsSI('U', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

S\_3\_Summer = CoolProp.PropsSI('S', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

V\_3\_Summer = 1/CoolProp.PropsSI('D', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 2 - Between Condensor and Expansion Valve Super-Heated Vapor

%%%%%%%%%%%%%

% Finding all values of the fluid at location 2

S\_2\_Summer = S\_1; % Constant Entropy

P\_2\_Summer = P\_3\_Summer; % Constant Pressure Isobar

T\_2\_Summer = CoolProp.PropsSI('T', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

H\_2\_Summer = CoolProp.PropsSI('H', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

U\_2\_Summer = CoolProp.PropsSI('U', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

V\_2\_Summer = 1/CoolProp.PropsSI('D', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 2a - On Vapor Dome During the process 2 to 3 in condensor

%%%%%%%%%%%%%

% Finding all values of the fluid at location 2a

Q\_2a=1; % On Vapor Dome as Saturdated Vapor

P\_2a\_Summer = P\_2\_Summer; % Isobar = Constant Pressure

T\_2a\_Summer = T\_3\_Summer; % Constant Temperature from T3

S\_2a\_Summer = CoolProp.PropsSI('S', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

H\_2a\_Summer = CoolProp.PropsSI('H', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

U\_2a\_Summer = CoolProp.PropsSI('U', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

V\_2a\_Summer = 1/CoolProp.PropsSI('D', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 4 - Mixture between the expansion valve and evaporator

%%%%%%%%%%%%%

% Finding all values of the fluid at location 4

T\_4 = T\_1; % Constant Temperature from 4 to 1 Process

H\_4\_Summer = H\_3\_Summer; % Expansion Valve is ~ constant enthalpy process

P\_4\_Summer = P\_1; % Isobaric Process

U\_4\_Summer = CoolProp.PropsSI('U', 'P', P\_4\_Summer, 'H', H\_4\_Summer, Working\_Fluid\_1);

S\_4\_Summer = CoolProp.PropsSI('S', 'P', P\_4\_Summer, 'H', H\_4\_Summer, Working\_Fluid\_1);

V\_4\_Summer = 1/CoolProp.PropsSI('D', 'P', P\_4\_Summer, 'H', H\_4\_Summer, Working\_Fluid\_1);

% Co-Efficient of Performance where COP = (QL/Wnet) = ((h1-h4)/(h2-h1))

COP\_Summer = (H\_1 - H\_4\_Summer)/(H\_2\_Summer - H\_1);

% Mass Flow Rate where flow rate = Cooling Capacity / Q\_L

dH\_Evaporator\_Summer = H\_1 - H\_4\_Summer;

Cooling\_Capacity = 500; % Watts

Mass\_Flow\_Rate\_Summer = Cooling\_Capacity / dH\_Evaporator\_Summer;

% Compressor Power

Compressor\_Power\_Summer = Mass\_Flow\_Rate\_Summer \* (H\_2\_Summer - H\_1); % [J/s]

if j == 1

COP\_R410a(k) = COP\_Summer;

MFR\_R410a(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 2

COP\_R290(k) = COP\_Summer;

MFR\_R290(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 3

COP\_Ammonia(k) = COP\_Summer;

MFR\_Ammonia(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 4

COP\_Water(k) = COP\_Summer;

MFR\_Water(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 5

COP\_R22(k) = COP\_Summer;

MFR\_R22(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 6

COP\_R114(k) = COP\_Summer;

MFR\_R114(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 7

COP\_IsoButane(k) = COP\_Summer;

MFR\_IsoButane(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 8

COP\_MDM(k) = COP\_Summer;

MFR\_MDM(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 9

COP\_R12(k) = COP\_Summer;

MFR\_R12(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 10

COP\_R143a(k) = COP\_Summer;

MFR\_R143a(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 11

COP\_R123(k) = COP\_Summer;

MFR\_R123(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 12

COP\_Acetone(k) = COP\_Summer;

MFR\_Acetone(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 13

COP\_R404a(k) = COP\_Summer;

MFR\_R404a(k) = Mass\_Flow\_Rate\_Summer;

elseif j == 14

COP\_R125(k) = COP\_Summer;

MFR\_R125(k) = Mass\_Flow\_Rate\_Summer;

else

COP\_R134a(k) = COP\_Summer;

MFR\_R134a(k) = Mass\_Flow\_Rate\_Summer;

end

k=k+1;

end

j=j+1;

end

for pos = 1:1000

COP\_R410a(pos) = abs(COP\_R410a(pos));

COP\_R290(pos) = abs(COP\_R290(pos));

COP\_Ammonia(pos) = abs(COP\_Ammonia(pos));

COP\_Water(pos) = abs(COP\_Water(pos));

COP\_R22(pos) = abs(COP\_R22(pos));

COP\_R114(pos) = abs(COP\_R114(pos));

COP\_IsoButane(pos) = abs(COP\_IsoButane(pos));

COP\_MDM(pos) = abs(COP\_MDM(pos));

COP\_R12(pos) = abs(COP\_R12(pos));

COP\_R143a(pos) = abs(COP\_R143a(pos));

COP\_R123(pos) = abs(COP\_R123(pos));

COP\_Acetone(pos) = abs(COP\_Acetone(pos));

COP\_R404a(pos) = abs(COP\_R404a(pos));

COP\_R125(pos) = abs(COP\_R125(pos));

COP\_R134a(pos) = abs(COP\_R134a(pos));

end

% Plotting COP vs Outside Temperature

figure(1)

% R Refrigerants

semilogx(COP\_R410a,Outside\_Temperature-273,'r',COP\_R290,Outside\_Temperature-273,'r')

semilogx(COP\_R404a,Outside\_Temperature-273,'r',COP\_R125,Outside\_Temperature-273,'r',COP\_R134a,Outside\_Temperature-273,'r',COP\_R22,Outside\_Temperature-273,'r')

semilogx(COP\_R123,Outside\_Temperature-273,'r',COP\_R12,Outside\_Temperature-273,'r',COP\_R143a,Outside\_Temperature-273,'r',COP\_R114,Outside\_Temperature-273,'r')

hold on

semilogx(COP\_Water,Outside\_Temperature-273,'b')

semilogx(COP\_IsoButane,Outside\_Temperature-273,'k')

semilogx(COP\_MDM,Outside\_Temperature-273,'m')

semilogx(COP\_Acetone,Outside\_Temperature-273,'g',COP\_Ammonia,Outside\_Temperature-273,'g')

plot([5,100],[20,20],[5,150],[50,50])

hold off

text(20,49,'Takeoff Environment Temperature')

text(20,19,'Cold Space Environment Temperature')

% Plot Syntax

%title('COP of a Thermal Mangement System with Varying Outside Temperature','FontSize',18)

xlabel('Coefficient of Performance','FontSize',22)

set(gca,'fontsize',20)

ylabel('Temperature','FontSize',22)

set(gca,'fontsize',20)

xlim([5 100])

ylim([15 55])

lgd = legend('\color{red} R-Refrigerants','\color{blue} Water Refrigerants','\color{black} IsoButane Refrigerants','\color{magenta} MDM Refrigerants','\color{green} Acetone/Ammonia');

lgd.FontSize = 22;

hold off

% Create a list from best to worst refrigerants in respect to its COP

COP\_R410a\_Max = COP\_R410a(915); % The index selection for Takeoff Temperature

COP\_R290\_Max = COP\_R290(915);

COP\_Ammonia\_Max = COP\_Ammonia(915);

COP\_Water\_Max = COP\_Water(915);

COP\_R22\_Max = COP\_R22(915);

COP\_R114\_Max = COP\_R114(915);

COP\_IsoButane\_Max = COP\_IsoButane(915);

COP\_MDM\_Max = COP\_MDM(915);

COP\_R12\_Max = COP\_R12(915);

COP\_R143a\_Max = COP\_R143a(915);

COP\_R123\_Max = COP\_R123(915);

COP\_Acetone\_Max = COP\_Acetone(915);

COP\_R404a\_Max = COP\_R404a(915);

COP\_R125\_Max = COP\_R125(915);

COP\_R134a\_Max = COP\_R134a(915);

COP\_TakeOff\_Sorted =sortrows({COP\_R410a\_Max,'R410a';COP\_R290\_Max,'R290';COP\_Ammonia\_Max,'Ammonia';COP\_Water\_Max,'Water';COP\_R22\_Max,'R22';COP\_R114\_Max,'R114';...

COP\_IsoButane\_Max,'IsoButane';COP\_MDM\_Max,'MDM';COP\_R12\_Max,'R12';COP\_R143a\_Max,'R143a';COP\_R123\_Max,'R123';COP\_Acetone\_Max,'Acetone';COP\_R404a\_Max,'R404a';COP\_R125\_Max,'R125';COP\_R134a\_Max,'R125'},1);

figure(5)

plot(Outside\_Temperature,MFR\_Acetone,'r',Outside\_Temperature,MFR\_R123,'b',Outside\_Temperature,MFR\_Ammonia,'m',Outside\_Temperature,MFR\_R290,'g')

xlabel('Temperature (C)','FontSize',22)

set(gca,'fontsize',20)

ylabel('Mass Flow Rate (kg/s)','FontSize',22)

set(gca,'fontsize',20)

lgd = legend('\color{red} Acetone','\color{blue} R123','\color{magenta} Ammonia','\color{green} R290');

lgd.FontSize = 22;

hold off

%Once the best COPs are chosen from the above analysis, the final full analysis of the desired working fluid is calculated and plotted

Working\_Fluid\_1 = {'R290'};

Working\_Fluid\_1 = Working\_Fluid\_1{1};

Max\_Pressure = 4250000;

%%%%%%%%%%%%%

% Constants for the Refridgeration Cycle for Durham, NH

%%%%%%%%%%%%%

Cold\_Space = 273+20; % Kelvin - Temperature of Cold Space of Pod

Hot\_Space\_Liftoff = 273 + 49; % Hottest Outside Environment

Cold\_Space\_Apogee = 273 - 18; % Coldest Outside Environment

## Summer is used to refer to Liftoff, and Winter is for Apogee %%%

%%%%%%%%%%%%%

% Location 1 - Between Evaporator and Compressor

%%%%%%%%%%%%%

% Finding all values of the fluid at location 1

Q\_1 = 1; % Saturated Vapor

T\_1 = Cold\_Space; % Temperature from 4 to 1

P\_1 = CoolProp.PropsSI('P', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

H\_1 = CoolProp.PropsSI('H', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

U\_1 = CoolProp.PropsSI('U', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

S\_1 = CoolProp.PropsSI('S', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

V\_1 = 1/CoolProp.PropsSI('D', 'T', T\_1, 'Q', Q\_1, Working\_Fluid\_1);

Q\_1\_Apogee = 1; % Saturated Vapor

T\_1\_Apogee = Cold\_Space; % Temperature from 4 to 1

P\_1\_Apogee = CoolProp.PropsSI('P', 'T', T\_1\_Apogee, 'Q', Q\_1, Working\_Fluid\_1);

H\_1\_Apogee = CoolProp.PropsSI('H', 'T', T\_1\_Apogee, 'Q', Q\_1, Working\_Fluid\_1);

U\_1\_Apogee = CoolProp.PropsSI('U', 'T', T\_1\_Apogee, 'Q', Q\_1, Working\_Fluid\_1);

S\_1\_Apogee = CoolProp.PropsSI('S', 'T', T\_1\_Apogee, 'Q', Q\_1, Working\_Fluid\_1);

V\_1\_Apogee = 1/CoolProp.PropsSI('D', 'T', T\_1\_Apogee, 'Q', Q\_1, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 3 - Between Condensor and Expansion Valve - Saturated Liquid

%%%%%%%%%%%%%

% Finding all values of the fluid at location 3

Q\_3 = 0; % Saturated Liquid

T\_3\_Summer = Hot\_Space\_Liftoff; % Temperature from 4 to 1

P\_3\_Summer = CoolProp.PropsSI('P', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

H\_3\_Summer = CoolProp.PropsSI('H', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

U\_3\_Summer = CoolProp.PropsSI('U', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

S\_3\_Summer = CoolProp.PropsSI('S', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

V\_3\_Summer = 1/CoolProp.PropsSI('D', 'T', T\_3\_Summer, 'Q', Q\_3, Working\_Fluid\_1);

T\_3\_Winter = Cold\_Space + 5; % Temperature from 4 to 1

P\_3\_Winter = CoolProp.PropsSI('P', 'T', T\_3\_Winter, 'Q', Q\_3, Working\_Fluid\_1);

H\_3\_Winter = CoolProp.PropsSI('H', 'T', T\_3\_Winter, 'Q', Q\_3, Working\_Fluid\_1);

U\_3\_Winter = CoolProp.PropsSI('U', 'T', T\_3\_Winter, 'Q', Q\_3, Working\_Fluid\_1);

S\_3\_Winter = CoolProp.PropsSI('S', 'T', T\_3\_Winter, 'Q', Q\_3, Working\_Fluid\_1);

V\_3\_Winter = 1/CoolProp.PropsSI('D', 'T', T\_3\_Winter, 'Q', Q\_3, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 2 - Between Condensor and Expansion Valve Super-Heated Vapor

%%%%%%%%%%%%%

% Finding all values of the fluid at location 2

S\_2\_Summer = S\_1; % Constant Entropy

P\_2\_Summer = P\_3\_Summer; % Constant Pressure Isobar

T\_2\_Summer = CoolProp.PropsSI('T', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

H\_2\_Summer = CoolProp.PropsSI('H', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

U\_2\_Summer = CoolProp.PropsSI('U', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

V\_2\_Summer = 1/CoolProp.PropsSI('D', 'P', P\_2\_Summer, 'S', S\_2\_Summer, Working\_Fluid\_1);

P\_2\_Winter = P\_3\_Winter; % Constant Pressure Isobar

S\_2\_Winter = S\_1\_Apogee; % Constant Entropy

T\_2\_Winter = CoolProp.PropsSI('T', 'P', P\_2\_Winter, 'S', S\_2\_Winter, Working\_Fluid\_1);

H\_2\_Winter = CoolProp.PropsSI('H', 'P', P\_2\_Winter, 'S', S\_2\_Winter, Working\_Fluid\_1);

U\_2\_Winter = CoolProp.PropsSI('U', 'P', P\_2\_Winter, 'S', S\_2\_Winter, Working\_Fluid\_1);

V\_2\_Winter = 1/CoolProp.PropsSI('D', 'P', P\_2\_Winter, 'S', S\_2\_Winter, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 2a - On Vapor Dome During the process 2 to 3 in condensor

%%%%%%%%%%%%%

% Finding all values of the fluid at location 2a

Q\_2a=1; % On Vapor Dome as Saturdated Vapor

P\_2a\_Summer = P\_2\_Summer; % Isobar = Constant Pressure

T\_2a\_Summer = T\_3\_Summer; % Constant Temperature from T3

S\_2a\_Summer = CoolProp.PropsSI('S', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

H\_2a\_Summer = CoolProp.PropsSI('H', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

U\_2a\_Summer = CoolProp.PropsSI('U', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

V\_2a\_Summer = 1/CoolProp.PropsSI('D', 'P', P\_2a\_Summer, 'Q', Q\_2a, Working\_Fluid\_1);

P\_2a\_Winter = P\_2\_Winter; % Isobar = Constant Pressure

T\_2a\_Winter = T\_3\_Winter; % Constant Temperature from T3

S\_2a\_Winter = CoolProp.PropsSI('S', 'P', P\_2a\_Winter, 'Q', Q\_2a, Working\_Fluid\_1);

H\_2a\_Winter = CoolProp.PropsSI('H', 'P', P\_2a\_Winter, 'Q', Q\_2a, Working\_Fluid\_1);

U\_2a\_Winter = CoolProp.PropsSI('U', 'P', P\_2a\_Winter, 'Q', Q\_2a, Working\_Fluid\_1);

V\_2a\_Winter = 1/CoolProp.PropsSI('D', 'P', P\_2a\_Winter, 'Q', Q\_2a, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Location 4 - Mixture between the expansion valve and evaporator

%%%%%%%%%%%%%

% Finding all values of the fluid at location 4

T\_4 = T\_1; % Constant Temperature from 4 to 1 Process

H\_4\_Summer = H\_3\_Summer; % Expansion Valve is ~ constant enthalpy process

P\_4\_Summer = P\_1; % Isobaric Process

U\_4\_Summer = CoolProp.PropsSI('U', 'P', P\_4\_Summer, 'H', H\_4\_Summer, Working\_Fluid\_1);

S\_4\_Summer = CoolProp.PropsSI('S', 'P', P\_4\_Summer, 'H', H\_4\_Summer, Working\_Fluid\_1);

V\_4\_Summer = 1/CoolProp.PropsSI('D', 'P', P\_4\_Summer, 'H', H\_4\_Summer, Working\_Fluid\_1);

T\_4\_Winter = T\_1\_Apogee; % Constant Temperature from 4 to 1 Process

H\_4\_Winter = H\_3\_Winter; % Expansion Valve is ~ constant enthalpy process

P\_4\_Winter = P\_1\_Apogee; % Isobaric Process

U\_4\_Winter = CoolProp.PropsSI('U', 'P', P\_4\_Winter, 'H', H\_4\_Winter, Working\_Fluid\_1);

S\_4\_Winter = CoolProp.PropsSI('S', 'P', P\_4\_Winter, 'H', H\_4\_Winter, Working\_Fluid\_1);

V\_4\_Winter = 1/CoolProp.PropsSI('D', 'P', P\_4\_Winter, 'H', H\_4\_Winter, Working\_Fluid\_1);

%%%%%%%%%%%%%

% Calculations of the Vapor Dome for the Refridgeration Cycle

% Will be plotted as a T-s and P-h Diagram

% Two curves will be plotted for each graph, one for the Saturated Liquid and one for Saturdated Vapor Sections

%%%%%%%%%%%%%

% Constant Variables and Pressure Array to Calculate Graph Values

Q\_SL = 0; % Saturated Liquid

Q\_SV = 1; % Saturated Vapor

P\_SL\_SV = linspace(100000,Max\_Pressure,1000); % Pressures for the Saturated Liquid Curve

T\_SL = zeros(length(P\_SL\_SV));

S\_SL = zeros(length(P\_SL\_SV));

H\_SL = zeros(length(P\_SL\_SV));

V\_SL = zeros(length(P\_SL\_SV));

T\_SV = zeros(length(P\_SL\_SV));

S\_SV = zeros(length(P\_SL\_SV));

H\_SV = zeros(length(P\_SL\_SV));

V\_SV = zeros(length(P\_SL\_SV));

% Looping 1000 times to provide values for the Vapor Dome Curves for T, s, h and v. P array will be graphed with them

for index=1:1000

T\_SL(index) = CoolProp.PropsSI('T', 'P', P\_SL\_SV(index), 'Q', Q\_SL, Working\_Fluid\_1) -273;

S\_SL(index) = CoolProp.PropsSI('S', 'P', P\_SL\_SV(index), 'Q', Q\_SL, Working\_Fluid\_1);

H\_SL(index) = CoolProp.PropsSI('H', 'P', P\_SL\_SV(index), 'Q', Q\_SL, Working\_Fluid\_1);

V\_SL(index) = 1/CoolProp.PropsSI('D','P',P\_SL\_SV(index), 'Q', Q\_SL, Working\_Fluid\_1);

T\_SV(index) = CoolProp.PropsSI('T', 'P', P\_SL\_SV(index), 'Q', Q\_SV, Working\_Fluid\_1) -273;

S\_SV(index) = CoolProp.PropsSI('S', 'P', P\_SL\_SV(index), 'Q', Q\_SV, Working\_Fluid\_1);

H\_SV(index) = CoolProp.PropsSI('H', 'P', P\_SL\_SV(index), 'Q', Q\_SV, Working\_Fluid\_1);

V\_SV(index) = 1/CoolProp.PropsSI('D','P',P\_SL\_SV(index), 'Q', Q\_SV, Working\_Fluid\_1);

end

%%%%%%%%%%%%%

% Calculations of Every Point during the Refridgeration Cycle

%%%%%%%%%%%%%

% Compressor Points - Points 1 to 2 - Constant Entropy

S\_Compressor\_Liftoff = S\_1; % Constant Entropy Process

S\_Compressor\_Apogee = S\_1\_Apogee; % Constant Entropy Process

P\_Compressor\_Summer = linspace(P\_1,P\_2\_Summer,1000);

P\_Compressor\_Winter = linspace(P\_1\_Apogee,P\_2\_Winter,1000);

T\_Compressor\_Summer = zeros(length(P\_Compressor\_Summer));

T\_Compressor\_Winter = zeros(length(P\_Compressor\_Summer));

S\_Compressor\_Summer = zeros(length(P\_Compressor\_Summer));

S\_Compressor\_Winter = zeros(length(P\_Compressor\_Summer));

H\_Compressor\_Summer = zeros(length(P\_Compressor\_Summer));

H\_Compressor\_Winter = zeros(length(P\_Compressor\_Summer));

V\_Compressor\_Summer = zeros(length(P\_Compressor\_Summer));

V\_Compressor\_Winter = zeros(length(P\_Compressor\_Summer));

for index = 1:1000

T\_Compressor\_Summer(index) = CoolProp.PropsSI('T', 'P', P\_Compressor\_Summer(index), 'S', S\_Compressor\_Liftoff, Working\_Fluid\_1) - 273;

T\_Compressor\_Winter(index) = CoolProp.PropsSI('T', 'P', P\_Compressor\_Winter(index), 'S', S\_Compressor\_Apogee, Working\_Fluid\_1) - 273;

S\_Compressor\_Summer(index) = CoolProp.PropsSI('S', 'P', P\_Compressor\_Summer(index), 'S', S\_Compressor\_Liftoff, Working\_Fluid\_1);

S\_Compressor\_Winter(index) = CoolProp.PropsSI('S', 'P', P\_Compressor\_Winter(index), 'S', S\_Compressor\_Apogee, Working\_Fluid\_1);

H\_Compressor\_Summer(index) = CoolProp.PropsSI('H', 'P', P\_Compressor\_Summer(index), 'S', S\_Compressor\_Liftoff, Working\_Fluid\_1);

H\_Compressor\_Winter(index) = CoolProp.PropsSI('H', 'P', P\_Compressor\_Winter(index), 'S', S\_Compressor\_Apogee, Working\_Fluid\_1);

V\_Compressor\_Summer(index) = 1/CoolProp.PropsSI('D', 'P', P\_Compressor\_Summer(index), 'S', S\_Compressor\_Liftoff, Working\_Fluid\_1);

V\_Compressor\_Winter(index) = 1/CoolProp.PropsSI('D', 'P', P\_Compressor\_Winter(index), 'S', S\_Compressor\_Apogee, Working\_Fluid\_1);

end

% Condensor - Points 2 to 2a - Constant Pressure

P\_Condensor\_Summer = P\_2\_Summer; % Constant Pressure Process

P\_Condensor\_Winter = P\_2\_Winter; % Constant Pressure Process

T\_Condensor\_Summer = linspace(T\_2\_Summer,T\_2a\_Summer+.25,1000); % Added constant .25 so it does not go to the mixture

T\_Condensor\_Winter = linspace(T\_2\_Winter,T\_2a\_Winter+.25,1000); % Added constant .25 so it does not go to the mixture

S\_Condensor\_Summer = zeros(length(P\_Condensor\_Summer));

S\_Condensor\_Winter = zeros(length(P\_Condensor\_Summer));

H\_Condensor\_Summer = zeros(length(P\_Condensor\_Summer));

H\_Condensor\_Winter = zeros(length(P\_Condensor\_Summer));

V\_Condensor\_Summer = zeros(length(P\_Condensor\_Summer));

V\_Condensor\_Winter = zeros(length(P\_Condensor\_Summer));

for index = 1:1000

S\_Condensor\_Summer(index) = CoolProp.PropsSI('S', 'T', T\_Condensor\_Summer(index), 'P', P\_Condensor\_Summer, Working\_Fluid\_1);

S\_Condensor\_Winter(index) = CoolProp.PropsSI('S', 'T', T\_Condensor\_Winter(index), 'P', P\_Condensor\_Winter, Working\_Fluid\_1);

H\_Condensor\_Summer(index) = CoolProp.PropsSI('H', 'T', T\_Condensor\_Summer(index), 'P', P\_Condensor\_Summer, Working\_Fluid\_1);

H\_Condensor\_Winter(index) = CoolProp.PropsSI('H', 'T', T\_Condensor\_Winter(index), 'P', P\_Condensor\_Winter, Working\_Fluid\_1);

V\_Condensor\_Summer(index) = 1/CoolProp.PropsSI('D', 'T', T\_Condensor\_Summer(index), 'P', P\_Condensor\_Summer, Working\_Fluid\_1);

V\_Condensor\_Winter(index) = 1/CoolProp.PropsSI('D', 'T', T\_Condensor\_Winter(index), 'P', P\_Condensor\_Winter, Working\_Fluid\_1);

end

% Condensor - Points 2a to 3 - Constant Pressure - Straight Line inside Vapor Dome

T\_Condensora\_Summer = [T\_2a\_Summer,T\_3\_Summer]; % Constant Temperature

T\_Condensora\_Winter = [T\_2a\_Winter,T\_3\_Winter]; % Constant Temperature

P\_Condensora\_Summer = [P\_2a\_Summer,P\_3\_Summer];

P\_Condensora\_Winter = [P\_2a\_Winter,P\_3\_Winter];

S\_Condensora\_Summer = [S\_2a\_Summer,S\_3\_Summer];

S\_Condensora\_Winter = [S\_2a\_Winter,S\_3\_Winter];

H\_Condensora\_Summer = [H\_2a\_Summer,H\_3\_Summer];

H\_Condensora\_Winter = [H\_2a\_Winter,H\_3\_Winter];

V\_Condensora\_Summer = [V\_2a\_Summer,V\_3\_Summer];

V\_Condensora\_Winter = [V\_2a\_Winter,V\_3\_Winter];

% Expansion Valve - Points 3 to 4 - Constant Enthalpy

H\_Valve\_Summer = H\_3\_Summer; % Constant Enthlpy Process

H\_Valve\_Winter = H\_3\_Winter; % Constant Enthlpy Process

P\_Valve\_Summer = linspace(P\_3\_Summer,P\_4\_Summer,1000); % Added constant .5 so it does not go to the mixture

P\_Valve\_Winter = linspace(P\_3\_Winter,P\_4\_Winter,1000); % Added constant .5 so it does not go to the mixture

S\_Valve\_Summer = zeros(length(P\_Valve\_Summer));

S\_Valve\_Winter = zeros(length(P\_Valve\_Summer));

T\_Valve\_Summer = zeros(length(P\_Valve\_Summer));

T\_Valve\_Winter = zeros(length(P\_Valve\_Summer));

V\_Valve\_Summer = zeros(length(P\_Valve\_Summer));

V\_Valve\_Winter = zeros(length(P\_Valve\_Summer));

for index = 1:1000

S\_Valve\_Summer(index) = CoolProp.PropsSI('S', 'H', H\_Valve\_Summer, 'P', P\_Valve\_Summer(index), Working\_Fluid\_1);

S\_Valve\_Winter(index) = CoolProp.PropsSI('S', 'H', H\_Valve\_Winter, 'P', P\_Valve\_Winter(index), Working\_Fluid\_1);

T\_Valve\_Summer(index) = CoolProp.PropsSI('T', 'H', H\_Valve\_Summer, 'P', P\_Valve\_Summer(index), Working\_Fluid\_1);

T\_Valve\_Winter(index) = CoolProp.PropsSI('T', 'H', H\_Valve\_Winter, 'P', P\_Valve\_Winter(index), Working\_Fluid\_1);

V\_Valve\_Summer(index) = 1/CoolProp.PropsSI('D', 'H', H\_Valve\_Summer, 'P', P\_Valve\_Summer(index), Working\_Fluid\_1);

V\_Valve\_Winter(index) = 1/CoolProp.PropsSI('D', 'H', H\_Valve\_Winter, 'P', P\_Valve\_Winter(index), Working\_Fluid\_1);

end

% Evaporator - Points 4 to 1 - Constant Temperature - Straight Line inside Vapor Dome

T\_Evaporator\_Summer = [T\_4,T\_1]; % Constant Temperature

T\_Evaporator\_Winter = [T\_4\_Winter,T\_1\_Apogee]; % Constant Temperature

P\_Evaporator\_Summer = [P\_4\_Summer,P\_1];

P\_Evaporator\_Winter = [P\_4\_Winter,P\_1\_Apogee];

S\_Evaporator\_Summer = [S\_4\_Summer,S\_1];

S\_Evaporator\_Winter = [S\_4\_Winter,S\_1\_Apogee];

H\_Evaporator\_Summer = [H\_4\_Summer,H\_1];

H\_Evaporator\_Winter = [H\_4\_Winter,H\_1\_Apogee];

V\_Evaporator\_Summer = [V\_4\_Summer,V\_1];

V\_Evaporator\_Winter = [V\_4\_Winter,V\_1\_Apogee];

%%%%%%%%%%%%%

% Calculation COP for each temperature

%%%%%%%%%%%%%

COP\_Summer = (H\_1 - H\_4\_Summer)/(H\_2\_Summer - H\_1);

COP\_Winter = (H\_1\_Apogee - H\_4\_Winter)/(H\_2\_Winter - H\_1\_Apogee);

%%%%%%%%%%%%%

% Graphing the T-s and P-h and P-vGraphs with Vapor Dome and Labels

%%%%%%%%%%%%%

% T-s Graph for the Summer

figure(2)

% Vapor Dome

plot(S\_SL/1000,T\_SL,'k',S\_SV/1000,T\_SV,'k')

hold on

% Processes

% Winter

plot(S\_Compressor\_Winter/1000,T\_Compressor\_Winter,'b',S\_Condensor\_Winter/1000, T\_Condensor\_Winter-273,'b',S\_Condensora\_Winter/1000, T\_Condensora\_Winter-273,'b',S\_Valve\_Winter/1000,T\_Valve\_Winter-273,'b',S\_Evaporator\_Winter/1000, T\_Evaporator\_Winter-273,'b')

text(S\_1\_Apogee/1000,T\_1\_Apogee-273, '\leftarrow State 1')

text(S\_2\_Winter/1000,T\_2\_Winter-273, '\leftarrow State 2')

text(S\_2a\_Winter/1000-.03,T\_2a\_Winter-273-1, 'State 2a','FontSize', 8)

text(S\_3\_Winter/1000-.06,T\_3\_Winter-273, 'State 3')

text(S\_4\_Winter/1000-.02,T\_4\_Winter-273-1, 'State 4')

text(1.8,27, strcat('Apogee COP: ' , num2str(COP\_Winter)))

% Summer

plot(S\_Compressor\_Summer/1000,T\_Compressor\_Summer,'r',S\_Condensor\_Summer/1000, T\_Condensor\_Summer-273,'r',S\_Condensora\_Summer/1000, T\_Condensora\_Summer-273,'r',S\_Valve\_Summer/1000,T\_Valve\_Summer-273,'r',S\_Evaporator\_Summer/1000, T\_Evaporator\_Summer-273,'m')

plot(S\_Compressor\_Winter/1000,T\_Compressor\_Winter,'m')

text(S\_2\_Summer/1000,T\_2\_Summer-273, '\leftarrow State 2')

text(S\_2a\_Summer/1000,T\_2a\_Summer-273, '\leftarrow State 2a')

text(S\_3\_Summer/1000-.07,T\_3\_Summer-273, 'State 3')

text(S\_4\_Summer/1000-.01,T\_4-273-1, 'State 4')

text(1.8,51, strcat('Takeoff COP: ' , '6.87'))

% Syntax

%title('Refridgeration Cycle during the Winter in Durham, NH','FontSize',20)

xlabel('Entropy (J/K)','FontSize',22)

set(gca,'fontsize',20)

ylabel('Temperature (Celcius)','FontSize',22)

set(gca,'fontsize',20)

lgd = legend('\color{red} Takeoff','\color{blue} Apogee','\color{black} Vapor Dome');

lgd.FontSize = 22;

xlim([1 2.5])

ylim([15 100])

hold off

% P-h Graph for the Summer

figure(3)

% Vapor Dome

plot(H\_SL/1000,P\_SL\_SV/1000,'k',H\_SV/1000,P\_SL\_SV/1000,'k')

hold on

% Processes

%Winter

P\_Condensor\_Winter = [P\_2\_Winter,P\_2a\_Winter];

H\_Condensor\_Winter = [H\_2\_Winter,H\_2a\_Winter];

P\_Valve\_Winter\_Ph = [P\_3\_Winter,P\_4\_Winter];

H\_Valve\_Winter\_Ph = [H\_3\_Winter,H\_4\_Winter];

plot(H\_Compressor\_Winter/1000,P\_Compressor\_Winter/1000,'b',H\_Condensor\_Winter/1000, P\_Condensor\_Winter/1000,'b',H\_Condensora\_Winter/1000, P\_Condensora\_Winter/1000,'b',H\_Valve\_Winter\_Ph/1000,P\_Valve\_Winter\_Ph/1000,'b',H\_Evaporator\_Winter/1000, P\_Evaporator\_Winter/1000,'b')

text(H\_1\_Apogee/1000,P\_1\_Apogee/1000, '\leftarrow State 1')

text(H\_2\_Winter/1000,P\_2\_Winter/1000, '\leftarrow State 2')

text(H\_3\_Winter/1000-16,P\_3\_Winter/1000, 'State 3')

text(H\_4\_Winter/1000-5,P\_4\_Winter/1000-35, 'State 4')

text(450,1000, strcat('Apogee COP: ' , num2str(COP\_Winter)))

% Summer

P\_Condensor\_Summer = [P\_2\_Summer,P\_2a\_Summer];

H\_Condensor\_Summer = [H\_2\_Summer,H\_2a\_Summer];

P\_Valve\_Summer\_Ph = [P\_3\_Summer,P\_4\_Summer];

H\_Valve\_Summer\_Ph = [H\_3\_Summer,H\_4\_Summer];

plot(H\_Compressor\_Summer/1000,P\_Compressor\_Summer/1000,'r',H\_Condensor\_Summer/1000, P\_Condensor\_Summer/1000,'r',H\_Condensora\_Summer/1000, P\_Condensora\_Summer/1000,'r',H\_Valve\_Summer\_Ph/1000,P\_Valve\_Summer\_Ph/1000,'r',H\_Evaporator\_Summer/1000, P\_Evaporator\_Summer/1000,'m')

plot(H\_Compressor\_Winter/1000,P\_Compressor\_Winter/1000,'m')

text(H\_2\_Summer/1000,P\_2\_Summer/1000, '\leftarrow State 2')

text(H\_3\_Summer/1000-14,P\_3\_Summer/1000, 'State 3')

text(H\_4\_Summer/1000-5,P\_4\_Summer/1000-50, 'State 4')

text(450,1800, strcat('Takeoff COP: ' , '6.87'))

% Plot Syntax

%title('Refrigeration Cycle during the Summer and Winter in Durham, NH','FontSize',20)

xlabel('Enthalpy (KJ/K)','FontSize',22)

set(gca,'fontsize',20)

ylabel('Pressure (KPa)','FontSize',22)

set(gca,'fontsize',20)

lgd = legend('\color{red} Takeoff','\color{blue} Apogee','\color{black} Vapor Dome');

lgd.FontSize = 22;

xlim([225 650])

ylim([500 4500])

hold off

% P-h Graph for the Summer

figure(4)

% Vapor Dome

plot(V\_SL/1000,P\_SL\_SV/1000,'k',V\_SV/1000,P\_SL\_SV/1000,'k')

hold on

% Processes

%Winter

P\_Condensor\_Winter = [P\_2\_Winter,P\_2a\_Winter];

V\_Condensor\_Winter = [V\_2\_Winter,V\_2a\_Winter];

%P\_Valve\_Winter = [P\_3\_Winter,P\_4\_Winter];

%V\_Valve\_Winter = [V\_3\_Winter,V\_4\_Winter];

plot(V\_Compressor\_Winter/1000,P\_Compressor\_Winter/1000,'b',V\_Condensor\_Winter/1000, P\_Condensor\_Winter/1000,'b',V\_Condensora\_Winter/1000, P\_Condensora\_Winter/1000,'b',V\_Valve\_Winter/1000,P\_Valve\_Winter/1000,'b',V\_Evaporator\_Winter/1000, P\_Evaporator\_Winter/1000,'b')

text(V\_1/1000,P\_1/1000+20, '\leftarrow State 1')

text(V\_2\_Winter/1000,P\_2\_Winter/1000+20, '\leftarrow State 2')

text(V\_3\_Winter/1000+.000001,P\_3\_Winter/1000-35, 'State 3')

text(V\_4\_Winter/1000,P\_4\_Winter/1000-35, 'State 4')

text(.000015,1000, strcat('Apogee COP: ' , num2str(COP\_Winter)))

% Summer

P\_Condensor\_Summer = [P\_2\_Summer,P\_2a\_Summer];

V\_Condensor\_Summer = [V\_2\_Summer,V\_2a\_Summer];

%P\_Valve\_Summer = [P\_3\_Summer,P\_4\_Summer];

%V\_Valve\_Summer = [V\_3\_Summer,V\_4\_Summer];

plot(V\_Compressor\_Summer/1000,P\_Compressor\_Summer/1000,'r',V\_Condensor\_Summer/1000, P\_Condensor\_Summer/1000,'r',V\_Condensora\_Summer/1000, P\_Condensora\_Summer/1000,'r',V\_Valve\_Summer/1000,P\_Valve\_Summer/1000,'r',V\_Evaporator\_Summer/1000, P\_Evaporator\_Summer/1000,'m')

plot(V\_Compressor\_Winter/1000,P\_Compressor\_Winter/1000,'m')

text(V\_2\_Summer/1000,P\_2\_Summer/1000+20, '\leftarrow State 2')

text(V\_3\_Summer/1000+.0000005,P\_3\_Summer/1000-65, 'State 3')

text(V\_4\_Summer/1000,P\_4\_Summer/1000-70, 'State 4')

text(.000015,1750, strcat('Takeoff COP: ' , '6.87'))

% Plot Syntax

%title('Refrigeration Cycle during the Summer and Winter in Durham, NH','FontSize',20)

xlabel('Specific Volume (m3/Kg)','FontSize',22)

set(gca,'fontsize',20)

ylabel('Pressure (KPa)','FontSize',22)

set(gca,'fontsize',20)

lgd = legend('\color{red} Takeoff','\color{blue} Apogee','\color{black} Vapor Dome');

lgd.FontSize = 22;

xlim([.0000005 .00006])

ylim([250 4500])

hold off