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.

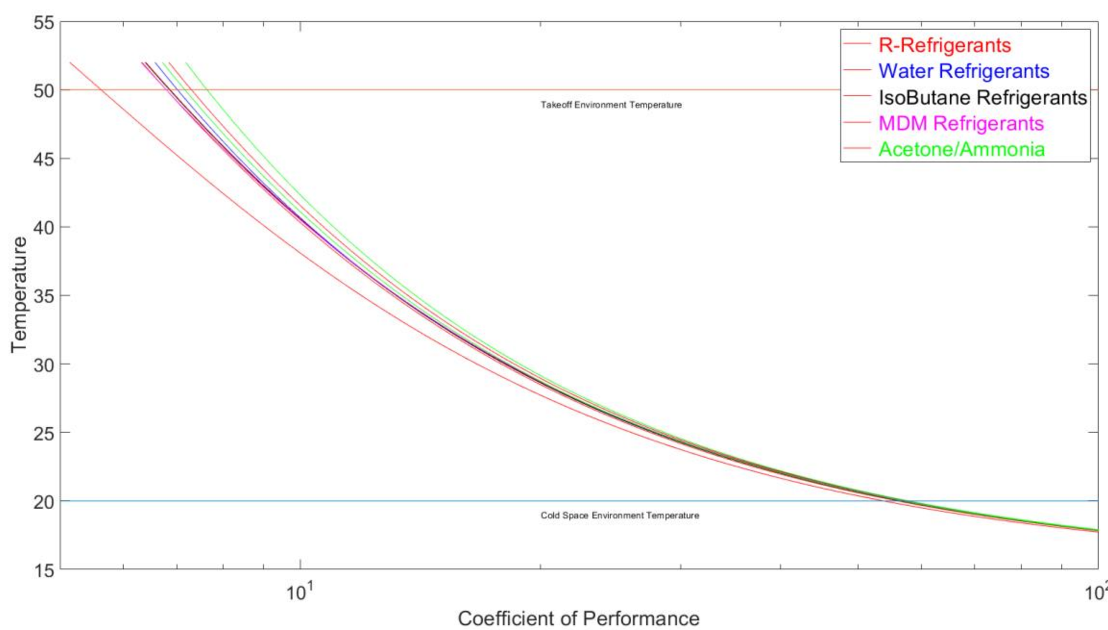

Figure 1- Coefficient of Performance of Several Refrigerants

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.

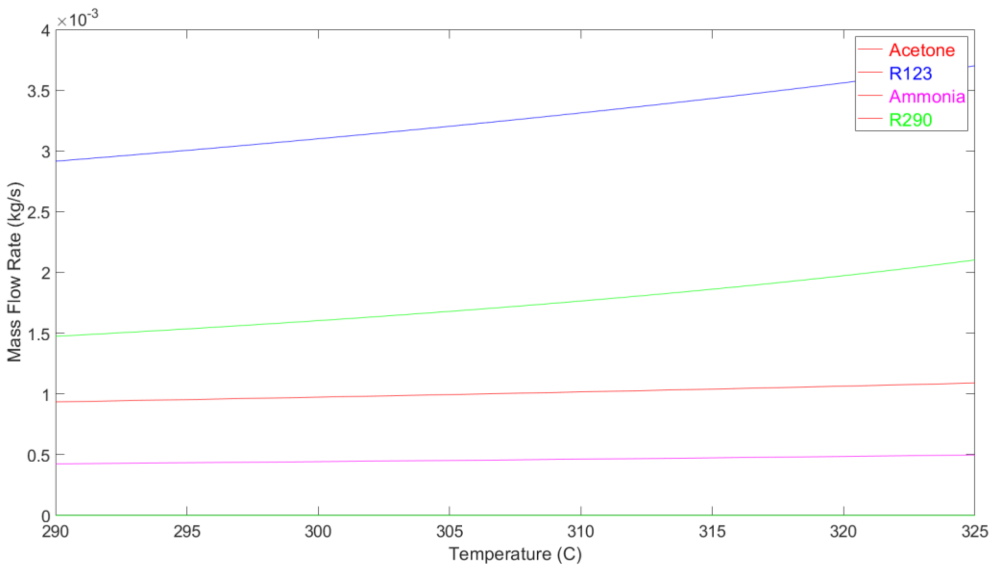

Figure 2- Mass Flow Rate of Most Efficient Refrigerants

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.

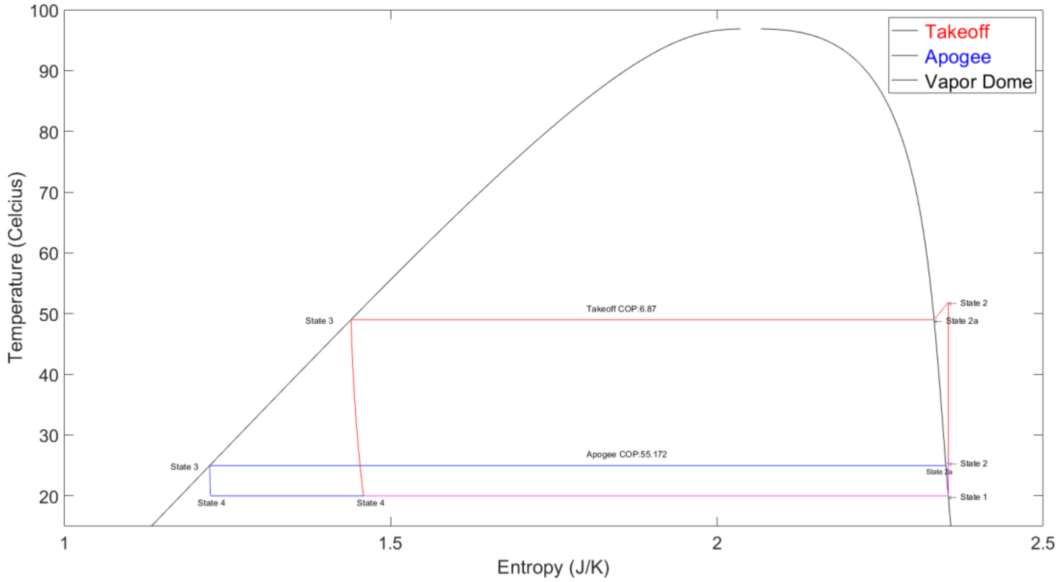

Figure 3- T-s Refrigeration Cycle of R290

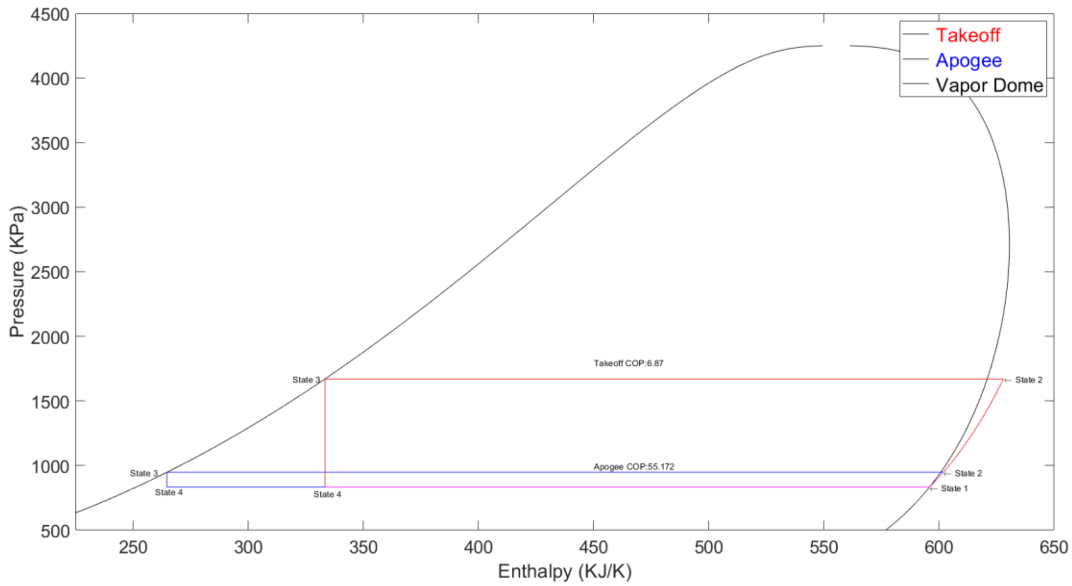

Figure 4- P-h Refrigeration Cycle of R290

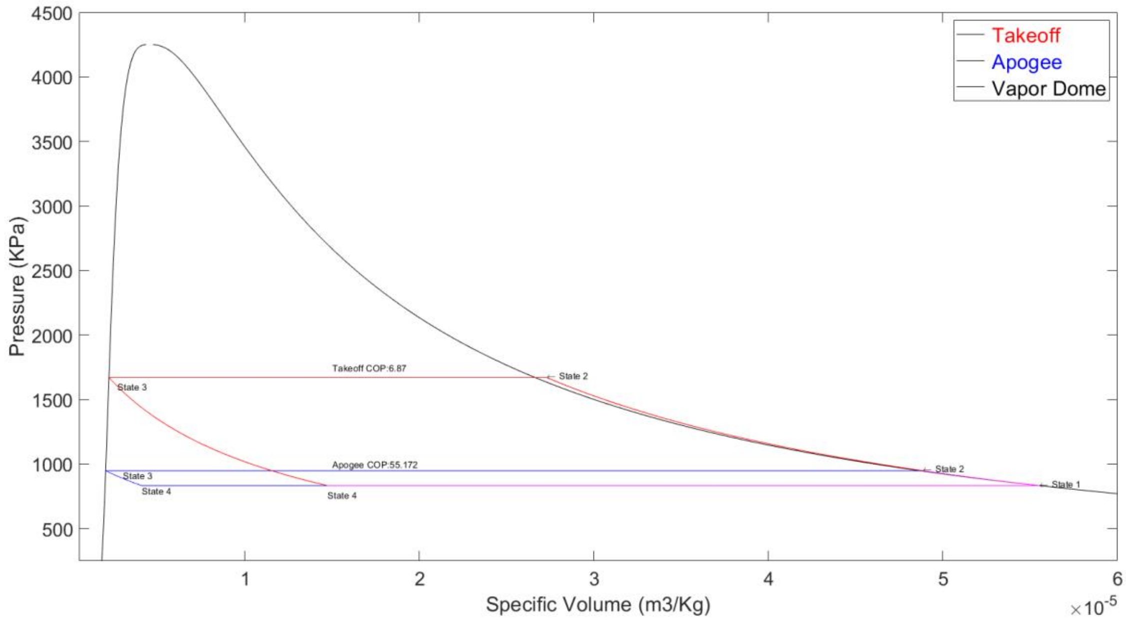

Figure 5- P-v Refrigeration Cycle of R290

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_Fl
uid_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_Ma
x,'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_R1
34a_Max,'R125'},1);
figure(5)
plot(Outside_Temperature,MFR_Acetone,'r',Outside_Temperature,MFR_R123,'b',Outside_Temperatu
re,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_Winte
r/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_Summe
r/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
```