# ABSTRACT

NICKELS, JOHN THOMAS. Optimization of Turbine and Heat Exchanger Stages For the Expansion Process of a Compressed Air Energy Storage System (Fortran Computer Modeling and Simulation). (Under the direction of Dr. Stephen Terry, PE).

Electricity production is one of the pillars supporting our modern world. Through careful planning and quick action, power producers keep the steady stream of electrons flowing through a seemingly endless network of transmission and distribution lines. Somewhat unpredictable, grid loads are supplied by base, intermediate, and peak generation plants. In order to supply all system loads reliably, energy producers may produce more energy than necessary, or may use quicker, more expensive electricity generation methods. Energy storage is seldom incorporated into an electricity grid's framework, particularly on a utility scale. Storage allows for, among other things, more predictable generation, less wasted resources, reserve capacity, electricity time shift, frequency regulation and intermittency remediation for renewable generation. The energy storage currently implemented generally takes the form of pumped hydropower, a geology-dependent technology, or as electrochemical battery storage, which can be very expensive with shorter longevity. New technologies are being studied and new projects are being constructed. An old technology that has been both studied and implemented is compressed air energy storage (CAES). Only a handful of facilities exist around the world, with the two oldest using the stored compressed air for combustion turbine-based electricity generation. The computer model written and run for the purpose of this Master's thesis simulates a compressed air energy storage system. The CAES system is considered as an energy storage system only, with input energy from excess production and output energy from expansion of stored air. The system incorporates intercooled compression, insulated storage, and expansion via turbines with heat exchangers to provide preheat. Simulations over certain parameter variations were tested in order to demonstrate system performance given excess power data. The resulting performance characteristics were used to optimize the variable inputs for the given data. The optimized system resulted in a net negative energy storage, mainly due to the temperature of the stored air, the compression heat rejected, the need for air preheat to avoid condensation in the expansion system, and the lack of thermal

storage. More study should be devoted to CAES and thermal energy storage, but the simple system configuration as simulated does not have promise as an energy storage system.

Optimization of Turbine and Heat Exchanger Stages for the Expansion Process of a Compressed Air Energy Storage System (Fortran Computer Modeling and Simulation)

by
John Thomas Nickels

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Mechanical Engineering

Raleigh, North Carolina

2015

APPROVED BY:

| | |
|---|---|
| Dr. Stephen Terry, PE<br>Committee Chair | Dr. Herbert Eckerlin, PE |

Dr. Alexei Saveliev

# DEDICATION

For Jude.

# BIOGRAPHY

John was born on September 29, 1976 to Marilyn and Kenneth Nickels in our Nation's Capital, the District of Columbia. His older brother RJ was ecstatic to have a brand new punching bag, as his old one was worn and full of tiny fist holes. By the time John was four, he had already deduced that he was likely the milkman's son, as he looked nothing like any of his relatives and was not anywhere as smart as them. I mean, seriously, he could not even tie his own shoes. Thankfully for little Johnny, Velcro was finally applied to shoes, sparing him from endless embarrassment. Due to his inherent intolerance of anything dairy-related, he was forced to give up his lifelong dream of following in his biological father's footsteps of dairy logistics, and chose instead to study the lucrative field of environmental biology. This education decision landed him in the ever expanding and high stakes field of water and soil testing, a career that is much sought after by peanut butter and jelly connoisseurs and ramen enthusiasts alike. It was only because he was making too much money that he decided to take an early retirement to explore his one true love, Mechanical Engineering. Sorry honey. He had given up his fleeting flirtations with the field during his childhood in Maryland over the lack of engineering schools that did not have some silly turtle for a mascot. Because seriously, a turtle? Why not a rock? It would probably be faster. Not that a Wolfpack is any better for a mascot. That is like calling a team a "swarm" instead of "the bees." Sure, both are scary, but the bees are what sting you. Anyway, it was only when John moved to North Carolina that he realized there were engineering schools that had somewhat logical mascots. It was at this point of recognition that John decided to throw his cares to the wind, move to Raleigh, and study to be a <u>Mechanical</u> Engineer……with cool robot arms and computer chip brain parts!

Yes, is joke. Maybe one day I tell you about chicken that is crossing road.

# ACKNOWLEDGMENTS

For my family, I would first like to acknowledge and thank my lovely wife, Anna, for her overwhelming patience and support throughout my extended return to education; you are truly the backbone of our happy little family. I want to thank my son Jude, not only for being **THE** BBE, but for reminding me of the reasons to stay on my path; thanks little dude! I would also like to acknowledge my parents, Ken and Marilyn, for their unwavering commitment to and belief in education; I could not have survived without your experience, advice and support. To that, I must mention my late grandfather and grandmother, Frank and Rita Wenzke; without their commitment to education and family, none of this endeavor would be possible. Finally, I would like to acknowledge my older brother RJ. You started this thing, and have been my model for success; you are truly the smartest person I have ever met.

Next, I would like to acknowledge Jackson Wooten, Daniel Poprocki, Kiran Thiruman, Corey Misenheimer, Taylor Atkins, Gopal Chaudry, Alex Viola, Harsha Ramakrishna, Kevin Martin, Josh Poole and Panth Naik. You guys put up with me for far too long; thanks for not kicking me out of the IAC office, even though I know you wanted to. Thanks also to Mike Breedlove, for giving me a shot in the shop, Dr. Paul Ro, for giving me a spot in your lab, and to Annie White for taking SO MUCH time to help; you are all awesome. Thanks to Dr. James Kribs, for your sanity checks and derivations; I'm glad you came back to State just to help me! Ha. Ha. Ha.

I would also like to acknowledge the Idaho National Laboratory for the opportunity to work on the energy storage project that initialized the research for this thesis. It is a noble task to push progress forward, and it is only with the determination to recognize and realize innovative ideas that future generations might thrive.

Finally, I would like to acknowledge my committee members, Drs. Stephen Terry, PE, Herbert Eckerlin, PE, and Alexei Saveliev. Thank you, Dr. Terry, for your willingness to take me on

board as a research assistant and your patient support throughout this endeavor; you are worth more than you know, and you know more than most. Thank you, Dr. Eckerlin, for your support, help, and unbelievable wealth of knowledge; you have spilled more engineering concepts on your tie than we can ever learn. Thank you, Dr. Saveliev for your commitment to the thermal sciences and to teaching; your humor, inquisitive nature, and capacity for understanding have been models for me during my studies and my research. I value your perspectives because you see things so completely. All three of you are exceptional teachers and remarkable people; I hold you all in the highest esteem and am so honored for you to be part of this.

# TABLE OF CONTENTS


LIST OF TABLES ..... viii
LIST OF FIGURES ..... ix
CHAPTER 1 ..... 1
Introduction ..... 1
Background ..... 4
1.1 Energy Production, Consumption and Demand ..... 4
1.2 Electricity Production and Power Cycles ..... 9
1.3 Finite Resources, Renewables, and Conservation ..... 11
1.4 System Loads and Energy Generation ..... 12
1.5 Energy Storage and Capacity ..... 14
1.6 The Current State of Compressed Air Storage ..... 16
1.7 Turbomachinery and Receivers ..... 19
1.7.1 Compressors ..... 19
1.7.2 Expansion Turbines ..... 25
1.7.3 Air Receivers ..... 27
CHAPTER 2 ..... 30
Relevant Thermodynamic Theory ..... 30
2.1 System Overview ..... 30
2.2 The Properties of Dry Air and Ideal Gas Behavior ..... 32
2.3 Intercooled Air Compression ..... 35
2.3.1 Compression Methods ..... 35
2.3.2 Staged Intercooling ..... 38
2.3.3 Isentropic Compressor Efficiency ..... 42
2.4 Psychrometrics ..... 42
2.5 Mass Storage ..... 45
2.6 Heated Air Expansion ..... 47
CHAPTER 3 ..... 52
Computer Modeling ..... 52
3.1 Fortran and Excel ..... 52
3.2 Fortran Model Structure ..... 53
3.2.1 User Input (Main Program) ..... 56

3.2.2 CAES Main (Main Subroutine) ..... 57
3.2.3 Specific Heat (Branch Subroutine) ..... 60
3.2.4 Compressor Work (Branch Subroutine) ..... 61
3.2.5 Excess Energy File Check (Branch Subroutine) ..... 62
3.2.6 Receiver Volume (Branch Subroutine) ..... 63
3.2.7 Humidity Properties (Branch Subroutine) ..... 64
3.2.8 Receiver Temperature (Branch Subroutine) ..... 65
3.2.9 Expander/Heat Exchanger (Branch Subroutine) ..... 68
3.2.9a Expander/Heat Exchanger [$T_{h\_out}$ Defined] (Branch Subroutine) ..... 68
3.2.9b Expander/Heat Exchanger [$kJ_{psreq}$ Defined] (Branch Subroutine) ..... 70
3.2.10 Runtime (Branch Subroutine) ..... 72
3.2.11 Turbine System (Branch Subroutine) ..... 75
CHAPTER 4 ..... 76
System Analysis and Results ..... 76
4.1 Excess Energy ..... 79
4.2 Receiver Conditions ..... 81
4.3 Turbine Stage / Heat Exchanger Number and Reheat ..... 87
4.4 Turbine Inlet Pressure ..... 97
4.5 Storage (Compressor Outlet) Pressure ..... 102
4.6 Storage and Turbine Inlet Pressures ..... 107
4.7 Turbine Mass Flow Rate ..... 112
4.8 Inlet Dew Point Temperature ..... 117
4.9 Final Model Plan ..... 119
CHAPTER 5 ..... 121
Final Simulation, Discussion and Conclusions ..... 121
5.1 Final Simulation ..... 121
5.2 Discussion ..... 126
5.3 Conclusions ..... 128
5.4 Future Work ..... 129
REFERENCES ..... 131
APPENDICES ..... 134
Appendix A ..... 135
Appendix B ..... 136

# LIST OF TABLES

**Table 1: Air Compression Variable Values Used for All Simulations ..... 76**
**Table 2: Turbine Stage Number and Temperature Gradient Variations ..... 88**
**Table 3: Turbine Inlet Pressure Variations ..... 98**
**Table 4: Storage Pressure Variations ..... 103**
**Table 5: Storage and Turbine Input Pressure Variations ..... 107**
**Table 6: Turbine Mass Flow Rate Variations ..... 113**
**Table 7: Dew Point Temperature Variations ..... 118**
**Table 8: Compressor Variable Values ..... 122**
**Table 9: Expansion System Variable Values ..... 122**

# LIST OF FIGURES

**Figure 1: Electricity Generation Allocation for System Demand (Summer Day) [Load Curve Data [1]]....... 2**
**Figure 2: System Demand by Customer Class [Load Data [1]] ....... 6**
**Figure 3: Electricity Generation by Fuel Source 2013 (EIA)[5]....... 7**
**Figure 4: Electricity Generation by Fuel Source (EIA)[5] ....... 8**
**Figure 5: Rankine (Steam) Cycle, T-s Diagram[8]....... 9**
**Figure 6: Open Brayton (Gas) Cycle[8]....... 10**
**Figure 7: Full Generation and Demand-Driven Generation [Load Data [1]]....... 13**
**Figure 8: Pumped Hydro Storage Cycle[15]....... 15**
**Figure 9: Positive Displacement Compressor Types....... 20**
**Figure 10: Reciprocating Piston-Cylinder Compression[24] ....... 21**
**Figure 11: Helical Screw Compression (Dark Areas Are Air)[25] ....... 22**
**Figure 12: Siemens STC-SX Axial Compressor[26]....... 23**
**Figure 13: GE Centrifugal Compressors (Closed and Open Impeller Designs)[27] ....... 24**
**Figure 14: OPRA 016 Radial Gas Turbine (With Centrifugal Compressor)[28]....... 25**
**Figure 15: Centrax CX400 Axial Gas Turbine (With Axial Compressor)[29] ....... 26**
**Figure 16: Basic Compressed Air Energy Storage System Components and Layout....... 31**
**Figure 17: P-v Diagram of Isothermal, Polytropic and Isentropic Compression[7] ....... 36**
**Figure 18: Staged Intercooled Compression System with Aftercooling....... 39**
**Figure 19: Staged Compression with Intercooling (Polytropic Processes Shown)[7] ....... 41**
**Figure 20: Insulated Storage Receiver Diagram ....... 46**
**Figure 21: Staged Heated Expansion System (Turbines & Heat Exchangers)....... 48**
**Figure 22: Compressed Air Energy Storage Sizing and Runtime Program ....... 55**
**Figure 23: Excess Power vs. Time (Peak Day)....... 80**
**Figure 24: Yearly Excess Power vs. Time (Day 145-152) ....... 81**
**Figure 25: Receiver Mass In/Out & Total vs. Time (Day 0-7)....... 83**
**Figure 26: Receiver Mass In/Out & Total vs. Time (Day 179-186)....... 84**
**Figure 27: Receiver Temperature & Pressure vs. Time (Day 0-7)....... 85**
**Figure 28: Receiver Temperature & Pressure vs. Time (Day 179-186)....... 86**

**Figure 29: Receiver Temperature & Pressure vs. Time (Day 182-183)........................................87**
**Figure 30: Reheat Energy and Receiver Temperature vs. Time (Day 182-183) ($T_{out}$=300 K).....89**
**Figure 31: Reheat Energy and Receiver Temperature vs. Time (Day 179-186) ($T_{out}$=$T_{dp}$) .........90**
**Figure 32: Reheat Energy and Receiver Temperature vs. Time (Day 179-186) (MAX Reheat..91**
**Figure 33: $T_{in}$, $T_{out}$, % Input Return and Δh vs. Stages (SIZE MODEL) (Full Reheat)..............92**
**Figure 34: $T_{in}$, $T_{out}$ = 300 K, % Input Return and Δh vs. Stages (SIZE MODEL) .......................93**
**Figure 35: $T_{in}$, $T_{out}$ = $T_{dp}$, % Return and Δh vs. Stages (SIZE MODEL).......................................94**
**Figure 36: Energy Payback, Reheater Size & Power Output vs. Stages (Full Reheat.................95**
**Figure 37: Energy Payback, Reheater Size & Power Output vs. Stages (Reheat for $T_{out}$=300 K) ........................................................................................................................................96**
**Figure 38: Energy Payback, Reheater Size & Power Output vs. Stages (Reheat for $T_{out}$=$T_{dp}$ ) .97**
**Figure 39: Minimum and Required Volume vs. Turbine Inlet Pressure ($T_{out}$=300 K)................99**
**Figure 40: Receiver and Turbine Temperature Limits vs. Turbine Inlet Pressure ($T_{out}$=300 K) ......................................................................................................................................101**
**Figure 41: Input Return and Power Out vs. Turbine Inlet Pressure ($T_{out}$=300 K) ....................102**
**Figure 42: Minimum and Required Receiver Volumes vs. Storage Pressure............................104**
**Figure 43: $\dot{m}_{max}$ and Total Mass (SIZE MODEL) vs. Storage Pressure.....................................105**
**Figure 44: Input Return and Power Out vs. Storage Pressure ...................................................106**
**Figure 45: Minimum and Required Receiver Volumes vs. Storage Pressure (ΔP 10 bar) ........108**
**Figure 46: Max Receiver Temperature and Pressure vs. Storage Pressure (ΔP 10 bar)...........109**
**Figure 47: $\dot{m}_{max}$ and M (SIZE MODEL) vs. Storage Pressure (ΔP 10 bar) ...............................110**
**Figure 48: Input Return and Power Out vs. Inlet Pressure (ΔP 10 bar).....................................111**
**Figure 49: Turbine Inlet and Outlet Temperatures vs. Storage Pressure (ΔP 10 bar).............112**
**Figure 50: Stored Energy vs. Mass Flow Rate..............................................................................114**
**Figure 51: Reheat Required vs. Mass Flow Rate..........................................................................115**
**Figure 52: Reheat Required vs. Mass Flow Rate (7.125 kg/s) ....................................................116**
**Figure 53: Stored Energy vs. Mass Flow Rate (7.125 kg/s) ........................................................117**
**Figure 54: Water Removed vs. Inlet Dew Point Temperature....................................................119**
**Figure 55: $T_{in}$ and $T_{out}$ vs. Stages ($T_{out}$=$T_{dp}$).......................................................................123**
**Figure 56: Expansion System Output and Reheat Power vs. Stages ..........................................124**
**Figure 57: Overall Efficiency and Energy Input Return vs. Stages............................................125**
**Figure 58: Energy Input Return vs. Stages....................................................................................126**

**Figure 59: Fraction of Input (Neglecting Reheat Energy)............................................................ 128**

# CHAPTER 1

## Introduction

Electricity is a mainstay of our technologically advanced civilization. Economies and lives depend on the steady flow of electrons through a seemingly endless matrix of metal strands. Many of the Earth's six billion occupants take this flow for granted, to power anything from heavy machinery on the factory floor to life support systems in an emergency room; few of them may recognize the lifetimes of technological developments and tremendous amounts of effort that go into the generation, transmission, and distribution of this essential utility. Electricity use and demand drives electricity production. Power must be produced and supplied when it is required, instantaneously and without fail. This requires major infrastructure which includes, among other things, generation plants, miles of power lines for transmission and distribution, transformers, metering and monitoring equipment, and occasionally, energy storage reserves.

Energy producers use various methods and technologies to meet the ever-changing power demand of its customer base, as shown in Figure 1. These methods include: base load generators, such as coal and nuclear plants, that run at steady rates to supply power that is always needed; intermediate load generators, such as combined cycle gas generators, that can be cycled on a daily basis to provide the somewhat variable loads ("between the shoulders" of the load curve) over the course of the day; and peak generator plants, such as gas turbines, that can be up and running in minutes to supply the peak system loads. Power companies may also purchase energy from other producers, which can be sold by contract or by bid, to meet demand.

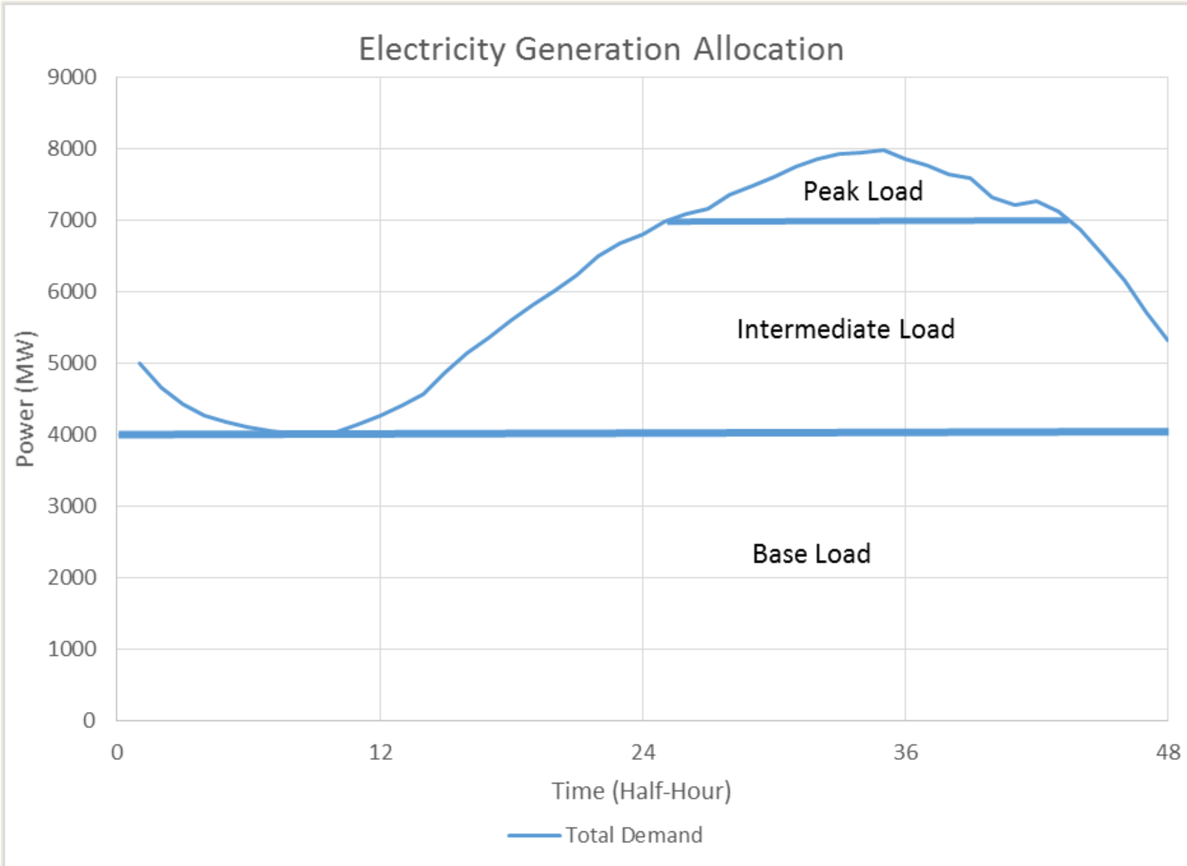


*Figure 1: Electricity Generation Allocation for System Demand (Summer Day) [Load Curve Data [1]]*

It is uncommon to have power grids that incorporate energy storage. Off-peak surplus energy is sold at a fraction of the on-peak price; a perfect example of the economic principle of supply and demand. However, immediate demand must be met with immediate production. This is handled, on the part of electric utilities, through careful planning and consideration, coupled with overproduction of power and quick action to buy or produce more. Large scale energy storage of excess production would ameliorate many supply side issues, particularly regarding peak loading, as well as conserve valuable fuel resources. Conversely, energy storage on a smaller scale for isolated power systems with a single generation source, such as for a remote campus or base that is off a main power grid, would allow the generator to cycle less often. Regardless of the specific implementation, the general idea is this: excess energy would be stored during off-peak times and extracted during on-peak times. The focus of this project is to design and analyze a compressed air energy storage (CAES) system model, and to optimize the number of turbine-heat exchanger couple sets for the expansion process.

The general outline of this thesis paper is:

- Chapter 1: Introduction and Background

  The first chapter will set the stage for the rest of the paper. It explains the reasons for creating the model, the current state of the art for compressed air energy storage and the physical components considered in the model.

- Chapter 2: Relevant Thermodynamic Theory

  The second chapter is devoted to theory, and explains the relevant scientific principles regarding air compression, storage and expansion. These are the thermodynamic foundations of the model.

- Chapter 3: Model Description

  Chapter three outlines and explains all of the components, key variables, and concepts of the computer model. This section details the overall construction and flow of the program, as well as the individual subroutines.

- Chapter 4: System Analysis and Results

  The purpose of chapter four is to explain the way the model was used, the specific variable changes that were made, and their output. The effect on number of turbine/heat exchanger couples is the focus. Overall results as well as those attributed to individual component (variable) augmentation are explored and analyzed.

- Chapter 5: Final Simulation, Discussion and Conclusions

  Chapter five is devoted to analyzing a final set of optimized system configurations. Conclusions based on the analysis performed in the previous chapter as well as on the final simulation are discussed. Recommendations for future work are covered in this final chapter.

## Background

### 1.1 Energy Production, Consumption and Demand

Two important concepts relating to energy use and production are consumption and demand. The primary "take-away" concept is that consumption is an amount, while demand is a rate. This difference can be explained with an analogy of a water fountain versus a fire hose. Both devices require water, but they require it at different rates. To deliver water to the end-user, a fire hose requires significantly more water every second it operates than a water fountain does; this is the concept of DEMAND. If you send the amount of water intended for the fountain to the fire hose, you will never put out your fire (unless it is very small). If you reverse the scenario, you may just blow your drinking fountain off the wall. Regardless of the arrangement, the water utility delivering the water to the two customers must be able to supply ALL of the system's customers at the TOTAL COMBINED DEMAND RATE at the time it is needed. If the system is below capacity, fires may continue to burn as hoses trickle.

The concept of consumption is much simpler. It is merely the AMOUNT used during a certain time period. It is a count, not a rate. Given sufficient time, the water fountain can consume as much or more water than the fire hose. For example, if a fire hose that requires 5 gallons per second to operate (DEMAND) is left fully open for one hour (3600 seconds), it will CONSUME an amount of 18,000 gallons. If a water fountain that requires 0.05 gallons per second to operate is left on for 100 hours (360,000 seconds), it will also consume an amount of 18,000 gallons. The periods over which the water is used are quite different, but the total consumption is the same.

The confusion begins for many people when they see the units used for energy demand and consumption. Demand is measured in units of power (kilowatts, kW) and consumption is measured in units of energy (kilowatt-hours, kWh). Don't worry, the unit for demand did not just become a count and the unit for energy did not just become an obscure unit of time! It is merely a bookkeeping practice to make dealing with power and energy easier. Since kilowatts are really just kilojoules per second:

$$kW = \frac{kJ}{s} = \left[\frac{ENERGY}{TIME}\right]$$

and kilowatt-hours are really just the number of kilojoules per second (kW) used for a period of time in hours (3600 seconds per hour):

$$kWh = {}^{kJ}\!/_{s} \times hours = \left[\frac{ENERGY}{TIME} \times TIME = ENERGY\right]$$

The units match up. Demand (power, kW) is what drives the scale of production capacity (system infrastructure). Consumption (energy, kWh) is the sum of the actual individual system demands over a period of time, which drives the degree to which a system is used (fuel use). Now that we understand demand and consumption, we can tackle system loads.

System loads are almost never steady. Individual users use energy differently at various times of the day and night, as shown in Figure 2. Industrial customers with 24 hour production cycles may hit their highest demand peak at 4:00 AM, while most residential customers are asleep at home. Seasonal variations change the way people use electricity as does location; not many North American residences are running air conditioners in January, and just as few may be running them in July if they are located in Maine. As stated in the introduction, system loads are split into base, intermediate and peak loads. The base load power is supplied by lower per-unit cost generation which is less likely to be cycled or run at below-optimum levels. The intermediate load is met with generation plants that may be more costly to operate than the base generation, but that can be cycled up and down on a daily basis. Finally, the peak load is usually met with power produced by gas turbine plants. These plants can cycle on and off quickly, but are less efficient and more costly to operate.

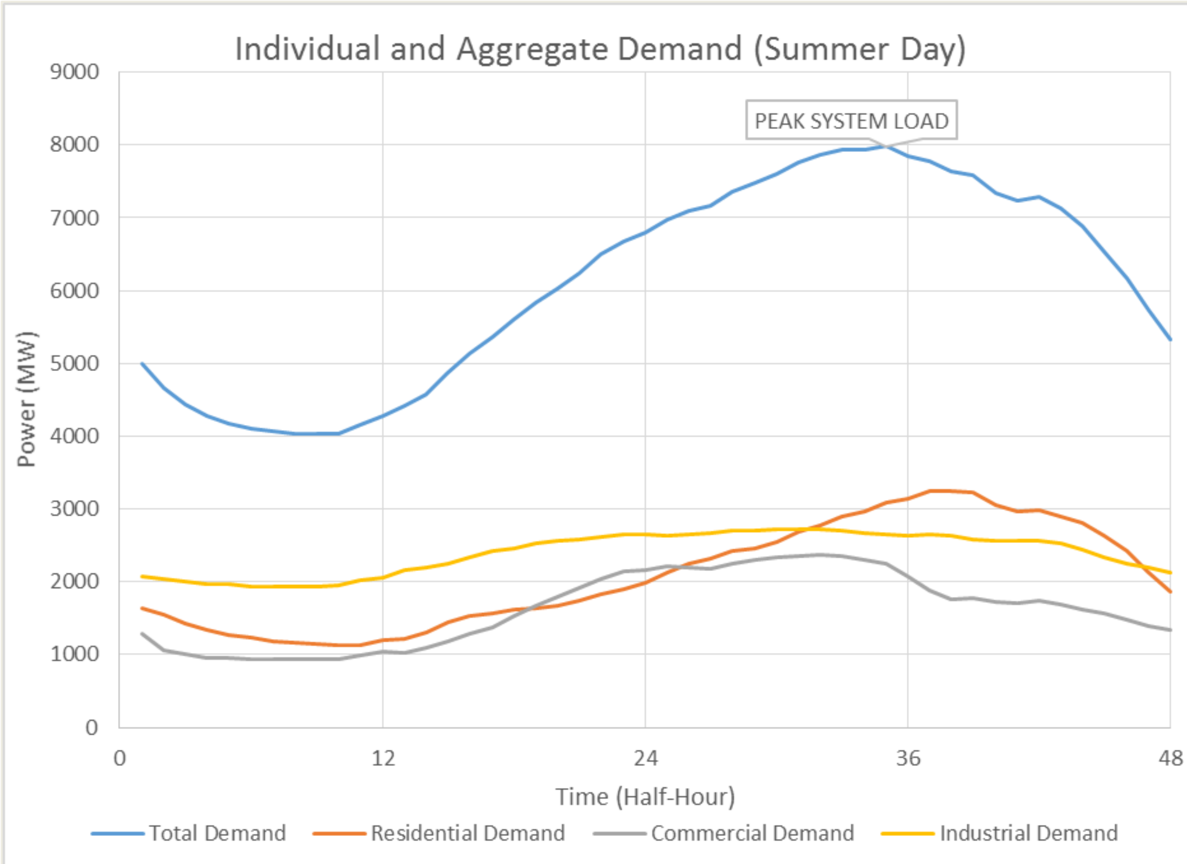


*Figure 2: System Demand by Customer Class [Load Data [1]]*

Renewable sources such as wind, solar and hydro play a smaller part in generation capacity overall as shown in Figure 3 and Figure 4, and their uses and effects are spread throughout the loading spectrum. Solar photovoltaic systems only produce during the day, and have sunlight-dependent levels of production. Solar thermal generation plants can produce power after dark if they have sufficient thermal storage. Wind turbines can produce power any time the wind is blowing sufficiently, day or night. The largest, most reliable source of renewable energy is hydro power, though geographical limitations and ecological considerations limit their implementation and continued use[2]. Efforts for and against hydro are being made: one side wants to increase power production from existing installations to maximize clean energy sources; the other side wants to remove the obstructions to healthy streams and a thriving ecosystem[3][4]. Time will tell the outcome, though reliable, zero-emissions and low cost electricity is a difficult position to argue against.

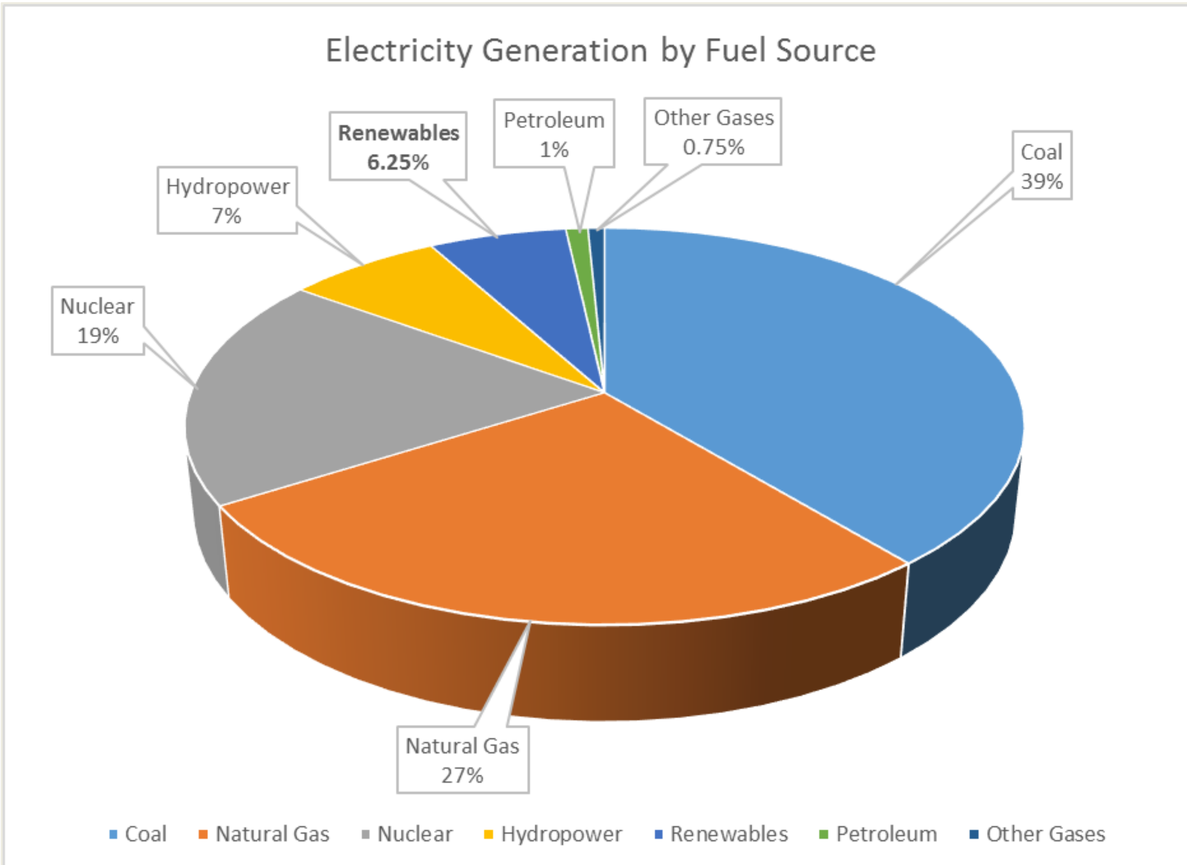


*Figure 3: Electricity Generation by Fuel Source 2013 (EIA)[5]*

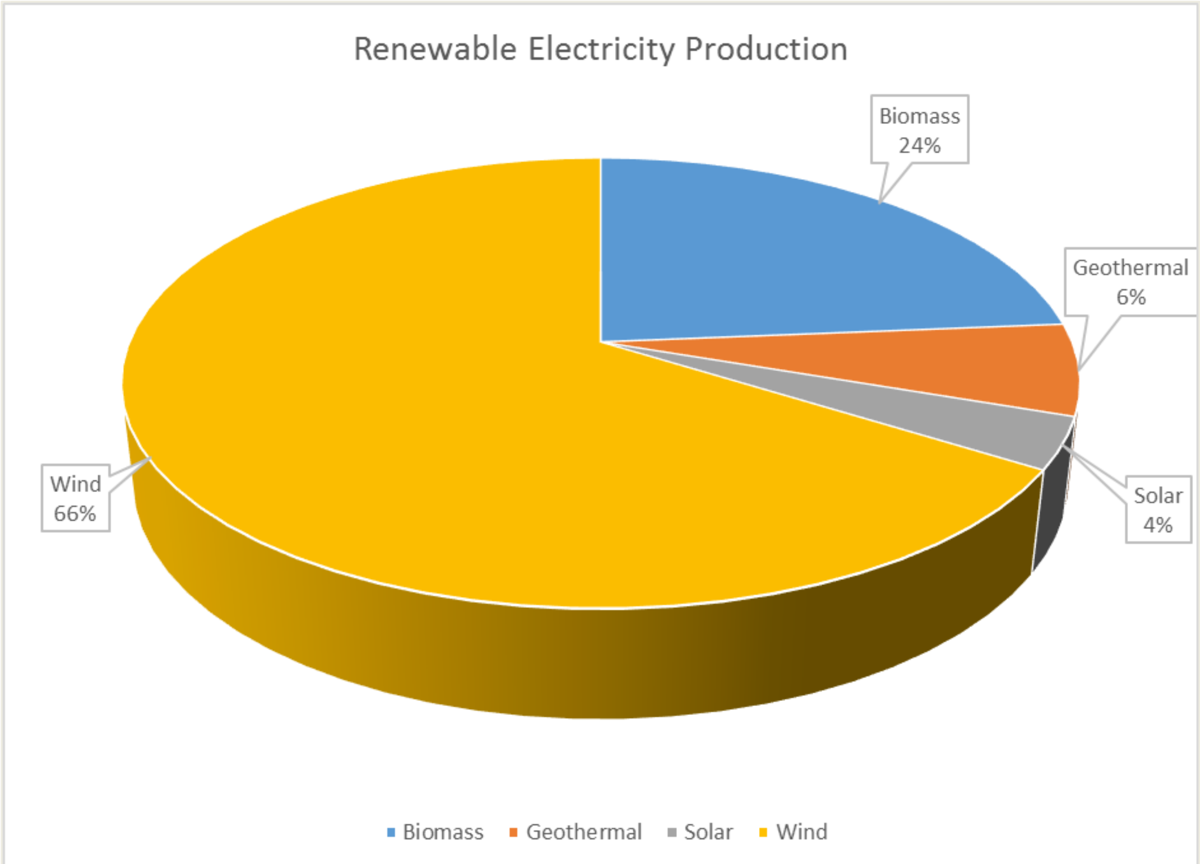


*Figure 4: Electricity Generation by Fuel Source (EIA)[5]*

Immense planning goes into the application and operation of these generation technologies. Since all of the demand at any given point must be met, the power generation capacity must exist to handle it. Different loads must be handled by different components of the generation infrastructure. It would not make sense to have gas turbines handling the base load while a nuclear plant cycles up and down to handle peak demand. This would be unnecessarily costly and could possibly damage the nuclear fuel rods. Since system loads are never known before they occur, power producers must make highly-educated guesses as to how much power to produce and when. To say the guessers, or system operators, have gotten better at guessing, or dispatching[6] would be a colossal understatement. Electric utilities have a myriad of strategies they use to predict, plan and produce, in order to meet the needs of every customer.

## 1.2 Electricity Production and Power Cycles

Electricity for consumer end-use is produced in much the same way as it was a century ago: an electric generator is turned by some mechanical means. This often reduces to a choice between two basic power cycles: The Rankine Cycle and The Brayton Cycle. The Rankine Cycle[7], Figure 5 below, involves heating water (or another working fluid) until evaporation, which increases the temperature and pressure of the working fluid. Oftentimes the fluid is superheated, which increases the fluid's ability to do useful work. This hot, high pressure vapor is expanded through a vapor expansion turbine which in turn produces the radial motion used to drive an electrical generator. The Brayton Cycle[7], conversely, involves compressing a gas, heating that gas, and expanding the hot high pressure gas through a gas expansion turbine, shown in Figure 6. In many applications, the gas is air which is heated via combustion with a fuel; the exhaust gases produced are sent through the turbine. Again, this expansion process produces the radial motion that drives an electricity generator.

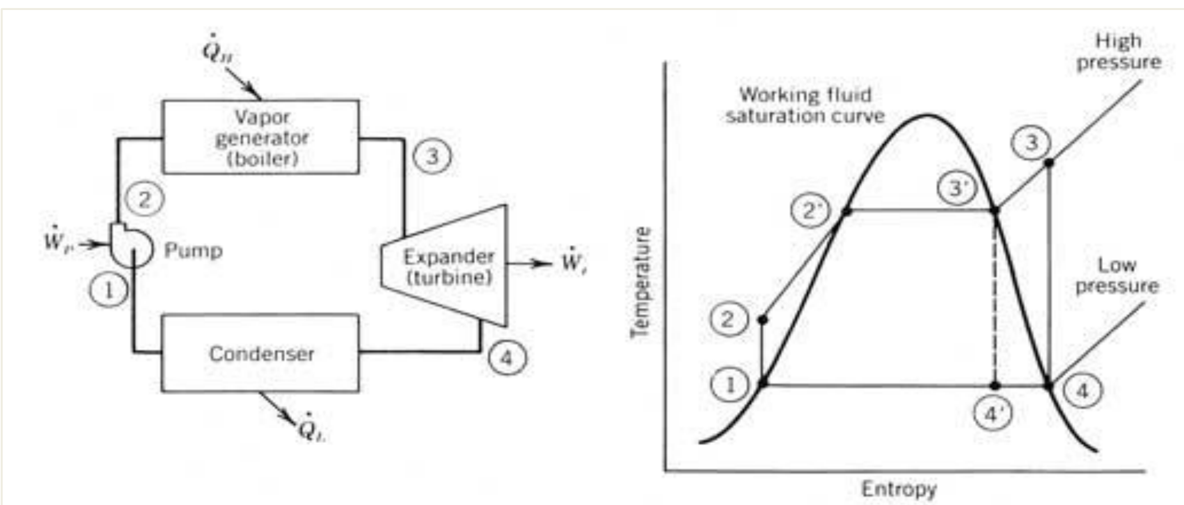


*Figure 5: Rankine (Steam) Cycle, T-s Diagram[8]*

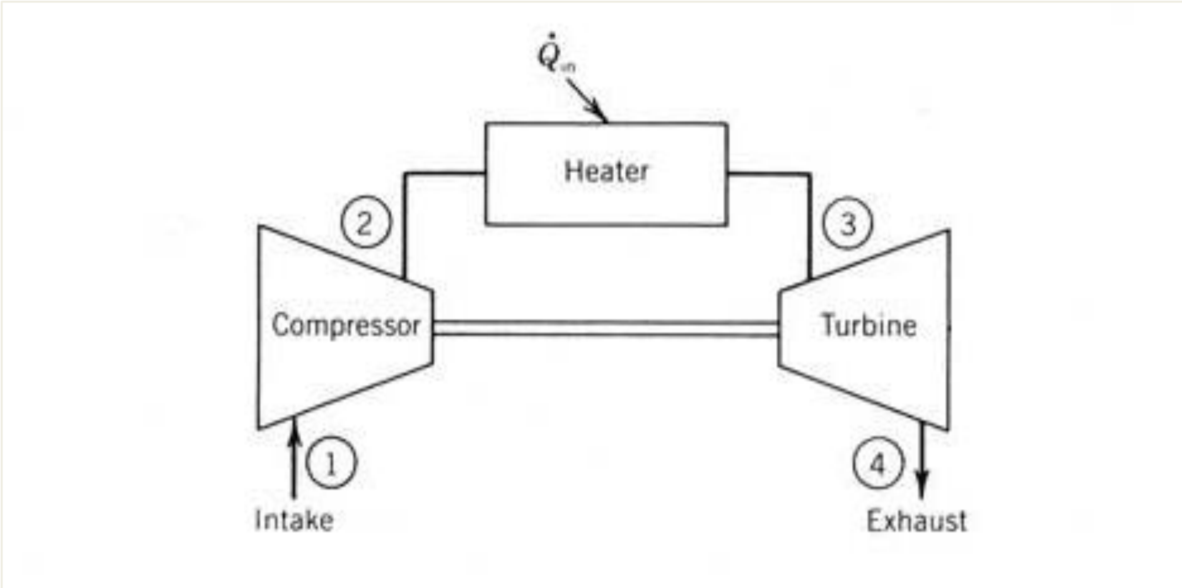


*Figure 6: Open Brayton (Gas) Cycle[8]*

The methods for heating the working fluid (vapor or compressed gas) can vary: coal, petroleum, natural gas or some other fuel can be combusted, nuclear fuel can undergo heat-releasing fission, solar energy can be concentrated to a central collector using mirrors, among others. Other methods for providing the motive force to turn generators also exist, such as water or wind turbines. Water turbines are most commonly used in conjunction with dammed streams; a potential (height difference) exists between the reservoir and the stream below, which drives water pressure and flow. Wind turbines work in much the same way, with the flow driven by atmospheric pressure gradient, the degree to which apparent in the wind's velocity. Finally, semiconductor-based solar photovoltaic and thermoelectric technologies are examples of non-mechanical energy production. Instead of turning electrical generators with some motive force, these methods use atomic excitation in a semiconductor medium to produce electric power[9].

## 1.3 Finite Resources, Renewables, and Conservation

Hydropower, wind, solar thermal and semiconductor-based energy generation are power production technologies that have no fuel costs associated with energy generation. After they are manufactured, constructed and implemented, nothing has to be combusted to make them work. They are commonly referred to as renewable energy sources. Despite the fact that the energy input for their operation is transferred or transformed, the medium of energy transmission remains (neglecting any long-term degradation effects) unchanged. Air carries kinetic energy as wind, water carries the potential energy of an elevation, and photons are pure electromagnetic radiation. Unfortunately, the wind does not always blow, the sun does not always shine and the rain does not always fall. This is another major reason to incorporate utility-scale energy storage. Inherent in many of these sources of “free-energy” are these production reliability and intermittency issues. Add to this the environmental impacts of construction and operation of these installations, water and land use concerns, and a growing pile of political and economic issues. The result is a considerable mess of problems that need solutions. This is not to say that these technologies should be disregarded; rather they should be as carefully considered as any other potential generation technology.

Non-renewable power production, conversely, is entirely dependent on finite fuel resources; they will not work unless combustion is occurring. The majority of the current electricity production infrastructure is based on using these finite resources, and for good reason: these fuels, which are extremely dense forms of energy storage, are able to drive electricity production reliably for long periods of time, independent of the time of day or the weather. The drawbacks of using these fuels, however, are formidable. Inherent in the way we use these fuels is their finite nature: once they have been burned, they cannot be used again. This incrementally depletes the overall store of available resources. In addition to this, a number of mining and extraction practices used to harvest these fuels are inherently destructive to the environment, degrading other resources such as land, water and air. These include but are not limited to heavy metal contamination, erosion, mountain-top removal, oil spills and methane emissions. Finally, many products of combustion are harmful to the health of humans and to the environment. Sulfuric acid, carbon monoxide, soot/ash, uncombusted fuel and nitrogen oxides are but a few contaminants that can pollute our environment, from which we are inseparable[10].

These benefits and liabilities of renewables and non-renewables are merely snowflakes on the tip of an iceberg of economic, scientific and political issues that are tied to energy use and production. In the end, one of the most effective means of alleviating the maladies caused by energy production is simply to use less energy; to conserve. New technologies, designs and practices, such as LED lighting, "green" building methods, and occupancy-based energy use allow us to use but a fraction of the electricity we once needed to perform the same functions. Every day, more people around the world are using electricity[11], making energy conservation through innovation crucial to our energy future. In the future, the question may not be, "how much can we produce?" but "how little can we use?"

### 1.4 System Loads and Energy Generation

Many energy production facilities (base load and intermediate load plants) are scaled to handle the largest practical amount of the system load, with smaller peak production facilities running to supply any deficit between the base/intermediate load and the system load. These peak facilities have higher associated costs than base load plants, which causes peak power to be more expensive to produce. Each facility requires land, generation equipment, and sufficient personnel for operations and maintenance; all of this only to serve a largely unpredictable customer demand, which may be for a period of only an hour[12]. The base load plants also require land, equipment, personnel and fuel, but they run much more efficiently at a constant, predictable rate. Power companies would much rather keep production costs low and supply all of the demand with the base load plants. Unfortunately, this would likely be more wasteful and expensive than supplying the demand with peak plants, as much of the power produced would go unused, as in Figure 7.

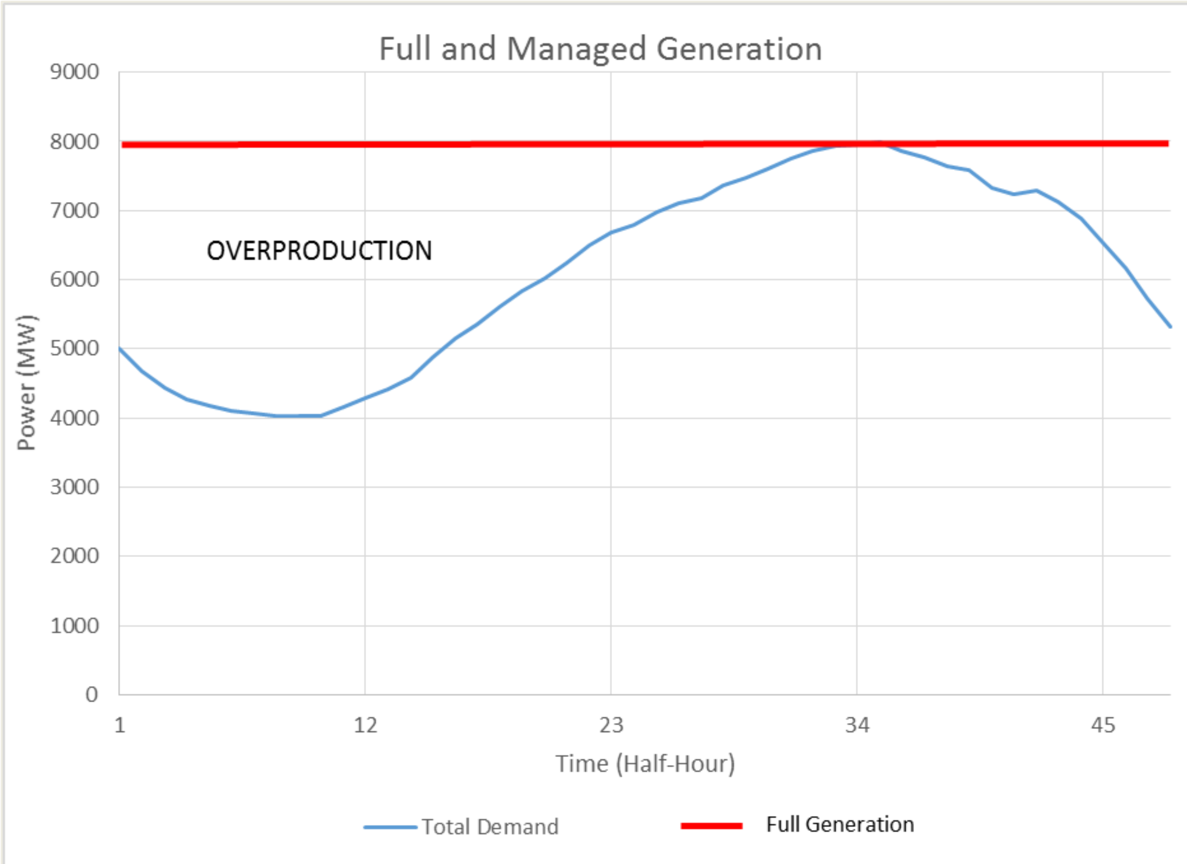


*Figure 7: Full Generation and Demand-Driven Generation [Load Data [1]]*

Generation facilities try to optimize production, but electricity demand fluctuates. Large base load facilities seldom scale back their operations significantly when the system load is lower than the base/intermediate load energy production level. It is simply less costly to run at a higher fire than to reduce or shut down production temporarily, only to have to ramp up operations again. Unfortunately, this leads to excess energy capacity or production during these "off-peak" periods. This "off-peak" energy is often sold at much lower rates than during "on-peak" periods. If this energy could be stored, it could be sold during the "on-peak" period, which could reduce reliance on the fast cycling "peaker" plants. With this, the base load production could be shifted to a more efficient point, and even more of the system load would be produced by the base/intermediate plants[13].

## 1.5 Energy Storage and Capacity

It seems that running more efficient and less expensive base load generation, coupled with energy storage for peak and shoulder ramp loads would be the answer to many of our energy conservation woes. Why do power utilities not have huge batteries next to their generation plants, storing excess energy for the next peak? In fact some do, but instead of having a mountainous array of chemical batteries for storage, some choose to incorporate other technologies that can be used at the scale of an electric utility. For example, electric motor-generators coupled to pump-turbines are used for pumped hydro storage. During off-peak generation periods, when there is excess energy production, the pump-turbines (using the motor-generator as a motor to drive the pump) pump water from a lower reservoir to an upper reservoir. Since the upper reservoir is located at some height above the lower reservoir, there exists a potential over which the energy can be stored. The excess electrical energy is converted to mechanical kinetic energy to pump the water to a higher elevation, thereby converting the kinetic energy to potential energy when the water reaches the top. The weight of the water carries this potential energy, a significant fraction of which (70%)[13] it gives back to the pump-turbine (now acting as a turbine spinning the generator) when the system is run in reverse, shown in Figure 8. Pumped hydro is conventionally only implemented where the geology is suitable for an installation. Geology in places such as Smith Mountain Lake and Leesville Lake in Virginia; Appalachian Power uses the elevation between these reservoirs to store excess off-peak energy[14]. Even if the geology is suitable, the economics of the facility construction or its operation may be too risky.

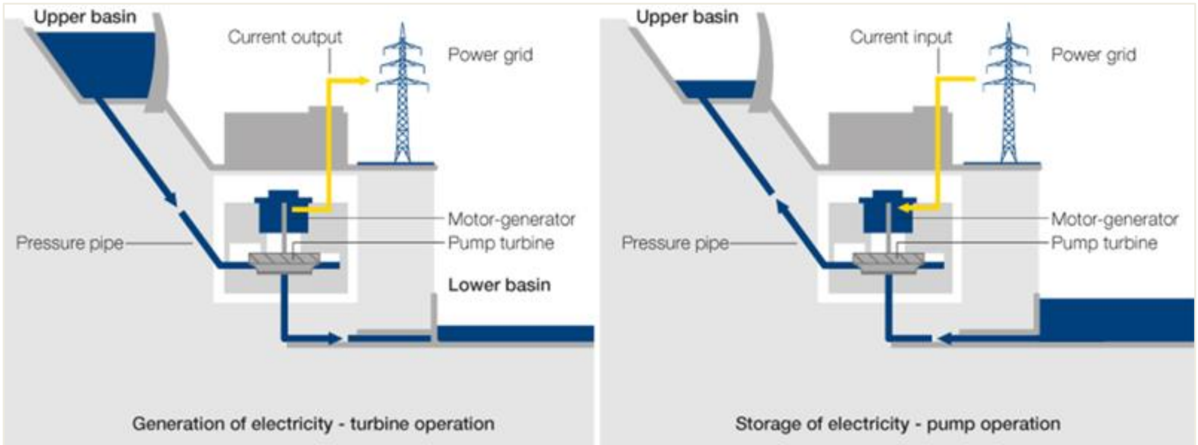


*Figure 8: Pumped Hydro Storage Cycle[15]*

Obviously and for good reason, economic feasibility dominates the decision-making process for any potential business venture. This holds just as true for the utility sector as it does for any other sector. If a technology is not energy-dense, cost-effective, and reliable with a long life span, it will not likely pass muster for an electric utility. Why would a utility even consider mechanical storage when chemical storage is readily available? They certainly have energy and power densities worth considering[2]. However, despite incredible breakthroughs in chemical battery technology over the years, the longevity of most chemical batteries is terribly lacking[16][17]. The lifecycle of chemical batteries is nowhere near those of pumped hydro, flywheel, or compressed air systems. If the life span of the installation is not more than ten years, the overnight costs will have to be quite low to justify the disposal costs for a fleet of dead batteries. Unfortunately, many energy-dense chemical batteries have a high per unit cost[18] as well as significant issues of environmental impact and energy use during manufacture[19]. Despite this, all forms of energy storage, whether they are mechanical, chemical or biological, should be researched and explored. The criteria requirements for any storage project, now and in the future, will include capacity, longevity, costs and payback.

## 1.6 The Current State of Compressed Air Storage

Compressed air has been widely used since the late 19th century, starting with mining and metal manufacturing[20]. Indeed many of the early air compressors predate the advent of electrical power, and were driven by steam engines. These systems did not necessarily incorporate receivers (vessels) to store compressed air. Despite this, storing energy with compressed air is, without doubt, not a new idea. In fact, all compressed air systems that make use of air receivers are by their nature energy storage systems. Energy is input to the mechanical system that compresses the air. This pressurized air is sent to a receiver for later use, capable of being transferred over distances via conduit. When called upon, the air is allowed to expand in a controlled manner to some lower pressure, usually through a pneumatic tool or device to perform a task.

The concept of using air to store excess energy for electricity generation is not new either. The first commercial utility-scale compressed air energy storage facility (CAES), still currently operating, was opened in Huntorf, Germany in 1978[21]. It stores off-peak energy as compressed air (up to 100 bar) in two solution-mined salt caverns with total volume of 310,000 $m^3$. It has a rated capacity of 321 MW and can operate for 2 hours. Generation is used to supply peak loads, supply a “spinning reserve” for grid system backup, and to ameliorate supply and demand fluctuations on the grid (frequency regulation). Huntorf is capable of going from an offline state to producing at full capacity without grid power (black start), in six minutes. This compressed air is used to feed the combustion and expansion processes of gas turbines.

Instead of using a large fraction (over 1/2) of the turbine shaft output power to compress the air (back work) in a traditional Brayton cycle configuration[7], the air is compressed with electrically-driven compressors. The air is compressed and instead of sending it straight into the combustion chamber and turbine, it is sent to one of the two storage caverns. It is compressed and stored at a different time than when it is needed (energy time shift). When the turbine needs to operate, the stored air is released from the receiver and into the combustion chamber. This allows the turbine to output more power to the generator, as no power needs to be devoted to compress air. This thirty-six year-old facility touts a 42% round-trip cycle efficiency for compression, storage and expansion.

The next-oldest CAES plant in operation is located in McIntosh, Alabama, and has been in service since 1991[21]. Like the Huntorf facility, the PowerSouth CAES installation stores compressed air (up to 76 bar) in a 283,000 $m^3$ solution-mined salt cavern. “Why a salt cavern?” you may ask. Water can be pumped into a salt bed formation to dissolve the salt, and the salt solution is pumped out leaving a void in the bed. Once the desired volume is dissolved away, the cavern is allowed to dry and seal itself back up, leaving an airtight, underground vessel[22]. The generation system at McIntosh also uses a gas turbine for the expansion process, but is rated for 110 MW and can run for up to 26 hours; it is used to supply peak loads and can serve as a spinning reserve. McIntosh makes use of a heat recovery system to preheat the cold compressed air with the turbine exhaust, which decreases fuel use by 25%.

The third CAES plant to come online is located in Gaines, Texas, and uses technology designed by General Compression, Inc.[21]. The 2 MW plant stores excess energy, also in a salt cavern, produced by wind turbines using proprietary “near isothermal” air compression technology. The rated run time for storage output is 250 hours. Unlike the facilities at Huntorf and McIntosh, the expansion system does not involve combustion of any fuel. The expansion process also uses proprietary “near isothermal” technology. This means that except for system losses, input energy is not lost as heat during the compression stage of the storage cycle. This is quite important, as isothermal or near isothermal compression and expansion are the “holy grail(s)” of CAES; if they can be performed quickly, reliably, and with acceptable roundtrip efficiencies, these processes have the potential to be groundbreaking. Of course, this system’s capacity is 55 times smaller than the McIntosh system. Whether or not the Gaines plant performs well, near isothermal compression and expansion on a Huntorf or McIntosh scale remains to be seen. Unfortunately, performance metrics are not available yet, even though the system has been in operation for approximately 2 years.

Other projects under construction or in the planning stages include[21]: a 200 MW, 5 hour adiabatic CAES system in Staßfurt, Germany that stores the heat removed during compression which will be added back prior to expansion; a 300 MW, 10 hour adiabatic CAES system in San Joaquin County, CA that will use an old natural gas reservoir for air storage; a 9 MW, 4 ½ hour CAES system in Queens, NY that uses steel piping as the storage receiver; and two 1 MW underwater compressed air energy storage (UCAES) facilities, one 5 m offshore from Toronto, ON in Lake Ontario, the other off the coast of Aruba. The Hydrostor demonstration facility in Canada will be able to run for 4 hours, and the facility in Aruba will run for 8 hours. Both systems will compress air with excess energy produced by wind turbines. These systems are unique in that they use inflatable receiver bladders and the hydrostatic pressure of the water column above them to store the air at a constant pressure. Finally, a 1.5 MW, 1 hour Isothermal CAES demonstration system manufactured by SustainX Inc. is in operation at the company's headquarters in Seabrook, NH. This system, according to the manufacturer, compresses air isothermally by spraying water into the compression system (piston) to absorb heat. This hot water is stored along with the air and the water is used to heat the air prior to the expansion process. This and Hydrostor's systems will not use fuel in the expansion process, using only the heat removed from compression to reheat the air.

This partial list should demonstrate that compressed air energy storage is becoming more popular rather than less. It should also be stated that one primary reason there are not more large CAES systems is the lack of geologically-suitable sites for massive underground caverns. These volumes are thought necessary for storage of air on a utility scale. This may indeed be the case, though this largely depends on a model of a centralized facility. If the storage facilities were smaller but more numerous and distributed, it may make sense to have a greater number of small scale receivers.

## 1.7 Turbomachinery and Receivers

This purpose of this section is to give the reader a basic overview of the components included in the system. The components present in the modeled system are intercooled/aftercooled compressors, adiabatic tanks, and expansion turbines with interstage heat exchangers. All generators, motors, wiring, electronics, controls, switches, piping, valves, metering equipment and structural components that will need to be considered for a full implementation were not considered here; these must be included in future, more complex modeling and simulation versions if major decisions are to be based on this model.

### 1.7.1 Compressors

Gas compressors do one thing: they bring physically push the molecules of the gas closer together, changing its volume. The same amount of mass that used to take up a larger volume can now fit into a smaller volume, increasing the mass density *($\rho$, [kg/m$^3$])*, or decreasing the specific volume *($\upsilon$, [m$^3$/kg])*, which are two sides of the same coin. Compressors do not necessarily increase the pressure of the gas, as heat can be removed which will lower the temperature. This will lower the pressure if the compressed volume remains constant. All compressors have some motive force, such as a driveshaft, and some energy input to supply the force; the work input pushes against the air that wants to expand. Compression can be accomplished in a number of ways, all of which fall under one of two compression classifications: positive-displacement compression and dynamic compression.

Positive-displacement compression, most notably performed by piston-cylinder (reciprocating) compressors and screw compressors, involves mechanically compacting the air volume into a smaller volume with a moving boundary[23]. Types are shown in Figure 9 below.

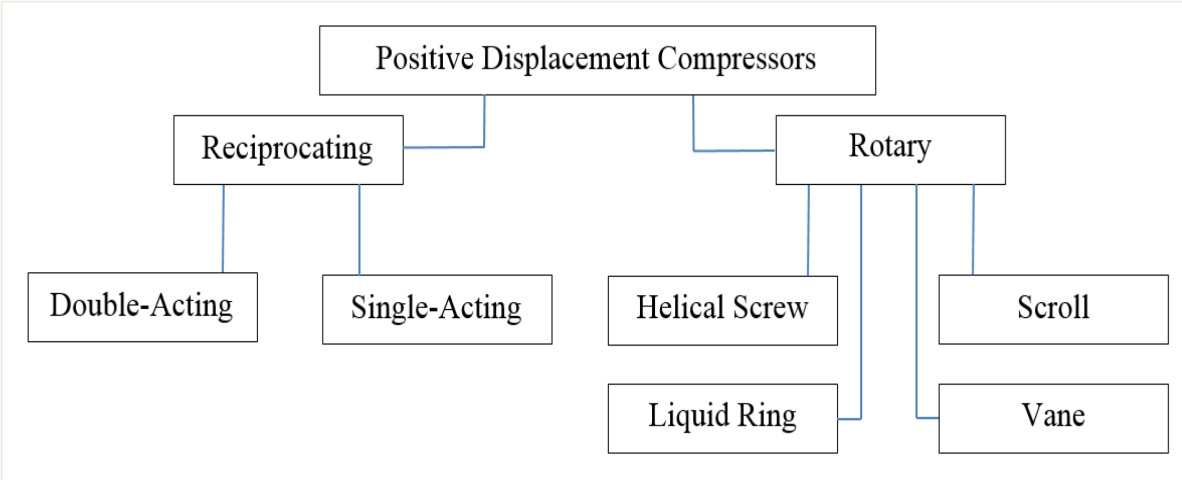


*Figure 9: Positive Displacement Compressor Types*

The piston-cylinder compressors, as shown in Figure 10, use valves to control intake and exhaust. When the design volume/pressure is reached, an exhaust valve opens to release the compressed air from the cylinder. Upon the expansion stroke, an intake valve opens to allow uncompressed air into the cylinder. This air is subsequently compressed when the intake valve closes, closing the cylinder, and the piston makes the volume smaller in the compression stroke.

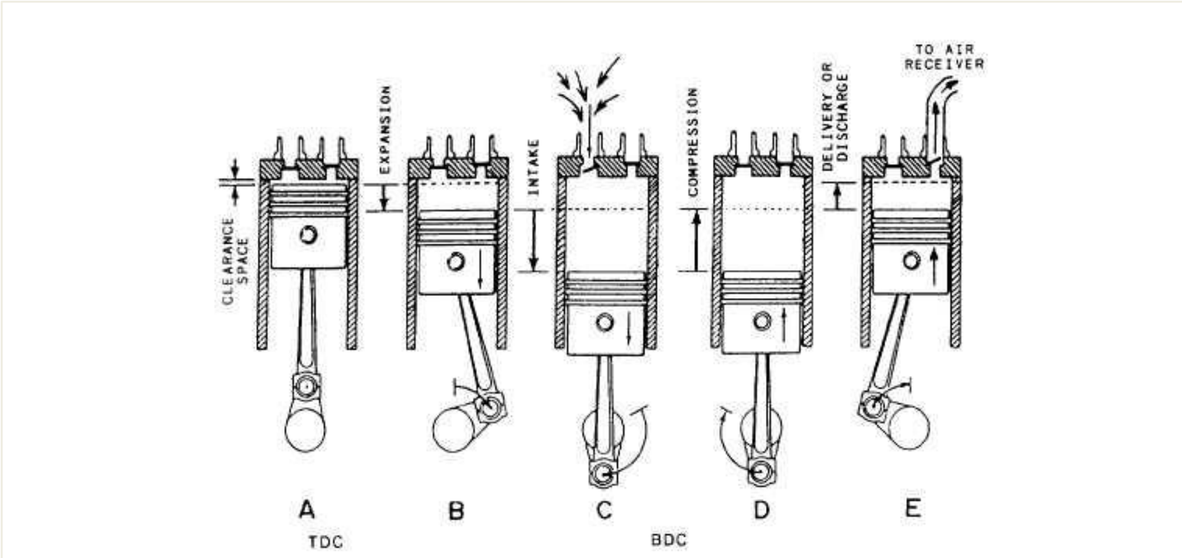


*Figure 10: Reciprocating Piston-Cylinder Compression[24]*

Screw compressors, as shown in Figure 11 are merely directional systems, with an inlet on one end and an outlet on the other end. The geometry of the screws, coupled with their counter-rotation, creates a negative pressure (suction) at the inlet and a positive pressure at the outlet.

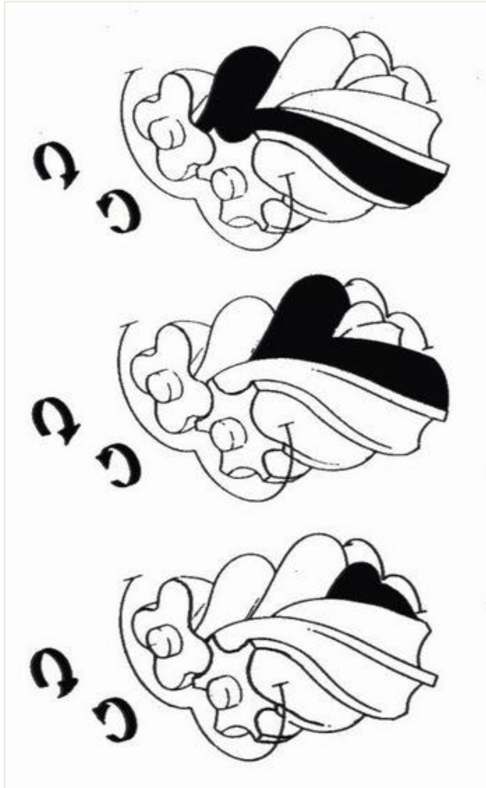

*Figure 11: Helical Screw Compression (Dark Areas Are Air)[25]*

Dynamic compression, most notably performed by axial compressors and centrifugal compressors, uses fast rotating impellers or rotors to increase the velocity of air[23]. This air is sent to either a stator (axial compressor) or a diffuser (centrifugal compressor) which partially slows the flow, converting some of the kinetic energy into potential energy by compressing the gas. In other words, the compressor "throws" the air so hard and fast at the diffuser/stator "wall" that the air molecules have no choice but to pack closer together. One major difference between axial and centrifugal compressors is the orientation of air flow. Axial compressors have air flow that is axial to the shaft; as the rotors suck air into the compressors, which is slowed by the stators and the ever reducing conical housing, it is pulled axially by the next set of rotors until it exits as compressed air at the outlet. Multiple stages or rows of rotors and stators occupy the shaft and housing, like rings of fans stacked on each other on a central shaft, as in Figure 12.

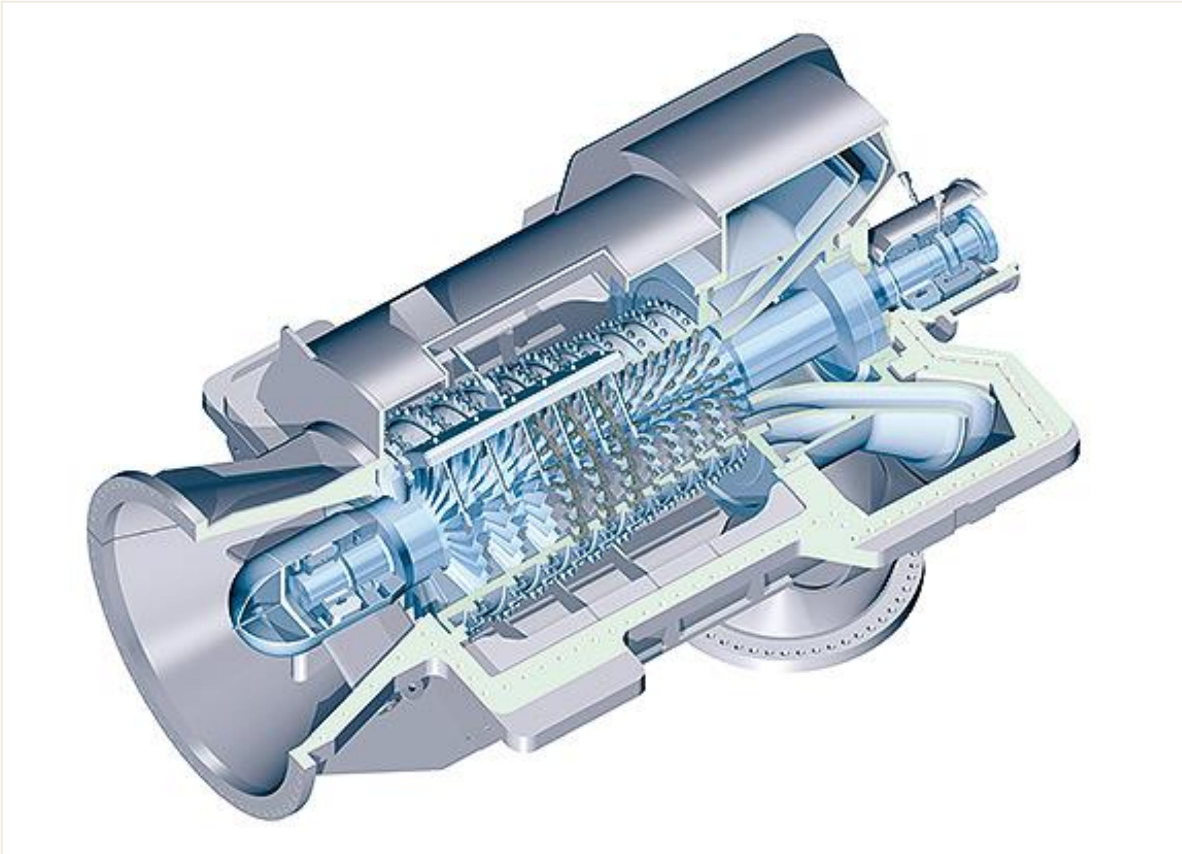

*Figure 12: Siemens STC-SX Axial Compressor[26]*

Centrifugal compressors also pull the air in axially to the inlet. However, once inside the impeller, which can be open or closed (Figure 13), the air is pushed, centrifugally, to the outside of the impeller disc. The air then travels radially towards the tip of the impeller. The tip is moving much faster than central area closer to the intake, and this is what increases the air velocity. Once the air is at the tip, it is carried then ejected into the diffuser, which slows its flow. At this point the compressed air can be routed around to the center of another impeller (next stage) to be compressed further or sent to the outlet of the compressor.

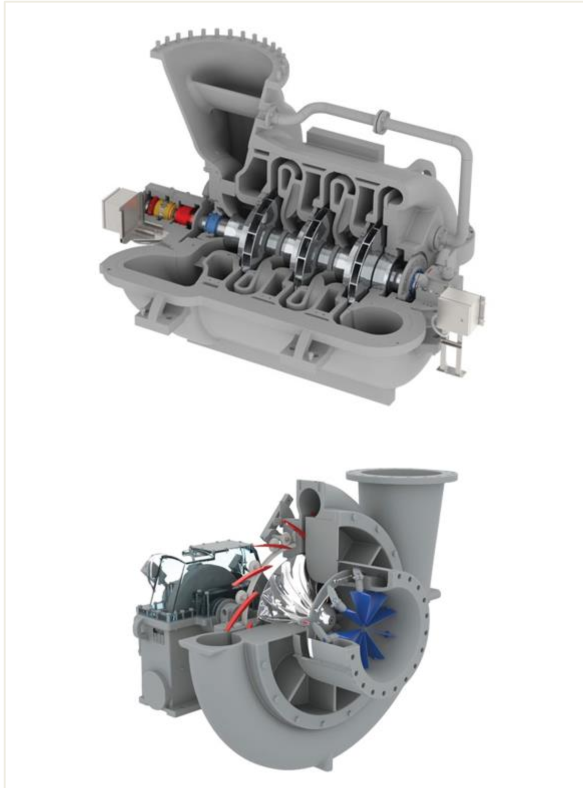

*Figure 13: GE Centrifugal Compressors (Closed and Open Impeller Designs)[27]*

A centrifugal compressor with intercooling is the type considered in the model design. There are a few reasons for this decision. First, they are very common, with a wide range of compression ratios and the ability to be included in multiple compressor/intercooler arrays; next, they are extremely robust, capable of handling high temperatures and pressures; finally, they are very efficient and can be used with variable frequency drive systems. This is not to say that another compression system will not be a better fit for any given facility. Depending on the design requirements for the system, a double-acting piston compressor or an axial compressor, among a myriad of other alternatives, may be a better choice.

### 1.7.2 Expansion Turbines

Gas turbines are dynamic compressors in reverse. Instead of using energy to make a gas smaller or to pressurize it, work is extracted from the expanding gas in a controlled process. In this setting we are only concerned with turbines used for power output, and not with thrust-producing engines used in aircraft. Different turbine technologies exist, and different operating principles exist as well. By whatever means employed, the goal is the same: to turn a shaft for useful work output. Here we focus on the two major classes: radial and axial. Examples of each are shown in Figure 14 and Figure 15. Akin to their compressor counterparts, they are dynamic machines.

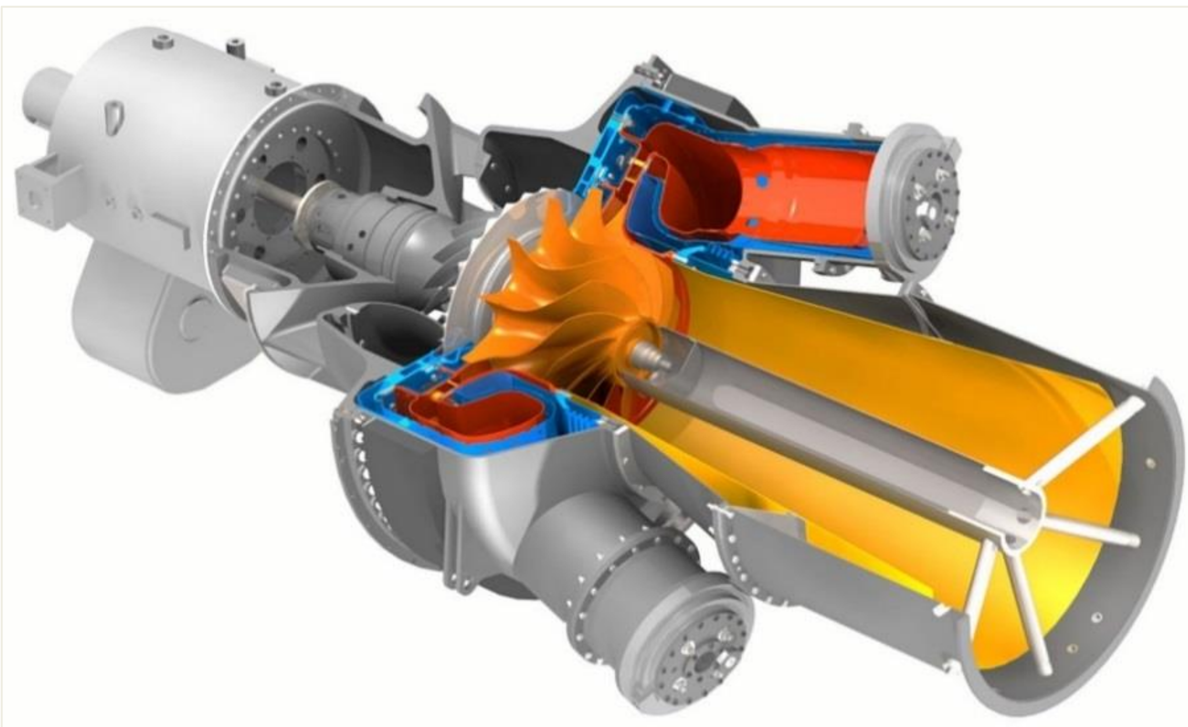

*Figure 14: OPRA 016 Radial Gas Turbine (With Centrifugal Compressor)[28]*

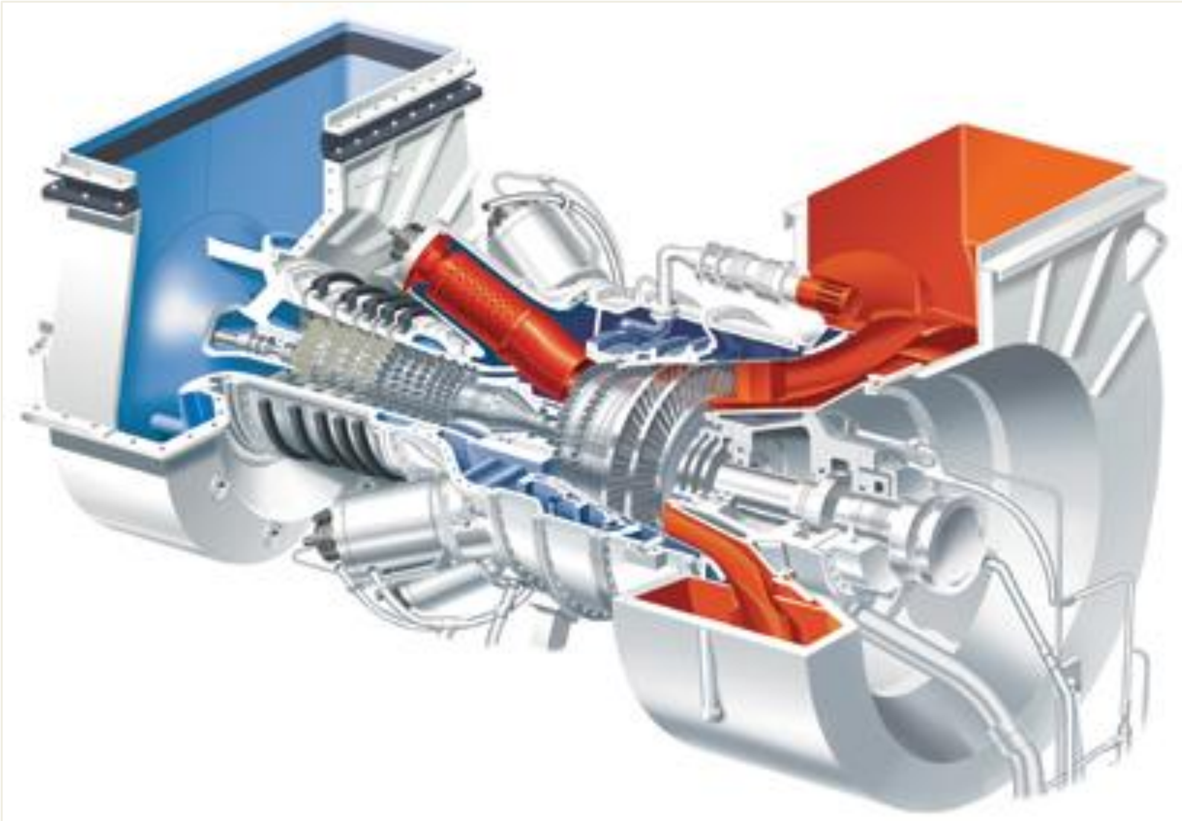

*Figure 15: Centrax CX400 Axial Gas Turbine (With Axial Compressor)[29]*

Radial turbines are much like centrifugal compressors. Compressed gas enters the turbine, changes direction, and spins the impeller as the fluid expands. Axial turbines are exactly like reversed axial compressors. The compressed fluid enters in the smaller end of the conical housing, passes stages of rotors and stators and does work moving the rotors as it expands and passes by the stators. All of this is contained by the ever increasing conical housing.

Other air expansion technologies exist, based on the positive-displacement principles discussed above, such as piston-cylinder air engines or vane expanders. Though not usually classified as turbomachines, positive-displacement expanders can perform the same function as dynamic turbomachines, but may be deficient as mass flow rates and mechanical energies get higher. As with dynamic expanders, the expanding gas in positive displacement expanders does work as it expands to a lower pressure. At least with piston-cylinder engines, however, large mass flow rates and mechanical inputs cannot be handled effectively[30]; they are inherently interrupted flow machines. At very high mass flows, axial turbines are the only machines that can transfer the thermodynamic energy of expansion into mechanical energy. For this model, centrifugal turbines with preheat and interheat exchangers are considered. The reason for this decision is much like the decision for choosing centrifugal compressors. In addition, the mass outflows for the modeled system are not high enough to consider much more expensive axial turbines. Finally, since interstage heat addition is considered, it is simpler to consider discrete single-stage turbines connected with piping and heat exchangers. This may need to be changed if the parameters change. An array of axial flow turbines with multiple stages and interconnected heat exchangers may need to be considered in the future.

### 1.7.3 Air Receivers

The crux of the storage problem is, well, storage. Energy, in order to be stored, must have a medium. In hydropower, it is water; in chemical batteries, it is the anode, cathode and electrolyte; in spinning flywheel systems, it is the flywheel itself; in compressed air systems, it is the air. All of these things take space; to have energy storage at the utility scale, a lot of space is needed. This is why it is difficult to economically justify energy storage. To give some perspective, if a 10 million gallon air tank is required to store compressed air, it will be approximately 75 feet in diameter, spanning the length of a football field. If this tank were made of steel and had a wall ½ inch thick, it would weigh over 13 tons. If the material and fabrication costs are not prohibitive, the burial costs, which would be required in order to reap the thermodynamic and safety benefits of an underground tank, may be.

The CAES facility in McIntosh, AL has an air reservoir/receiver that is 10 million cubic feet, which is 7 ½ times larger than our stadium-sized example. This is a solution-mined salt cavern, not a steel pressure vessel. No container had to be buried; water was pumped in, pumped out, then disposed of. They can "get away" with such a small volume because they compress air to

76 bar, or 1,100 psi. The higher the pressure, the more tightly packed the molecules of air become. This, by definition, increases the density of the air, which means that more air can be stored in a smaller space. If you compress to half of the pressure, and the temperature stays the same, you will need a storage vessel twice as big. Compressing further takes more power, and that is the tradeoff.

Air can be stored in rigid vessels, salt caverns, rock caverns, aquifers, pipe, and inflatable bladders. There are two main distinctions of pressure vessel types: rigid boundary and moving boundary. Rigid boundary vessels are the tanks most commonly thought of when gas storage is mentioned. They are the steel tanks or cavernous geologic voids used to store compressed gases. They have boundaries that are practically immovable by the contained gas; the volume does not change. The gas pressure generally increases as more gas is added to the volume, unless enough heat is removed to proportionally lower the temperature. As mass is removed, the pressure generally decreases, unless heat is added to proportionally increase the temperature. The maximum pressure is dictated by the strength of the construction materials, and the minimum pressure is the ambient pressure of the surroundings.

Moving boundary vessels are just the opposite. These include aquifers, flooded vaults, sealed piston-cylinders, and inflatable bladders. As more mass is added, the boundary (wall) expands to compensate and increase the volume. The pressure, given the boundary is under static load and free-moving, does not increase significantly as a result of added mass. Therefore, the pressure that is available to the system when the reservoir is full is the same pressure available when the reservoir is discharged. This is the type of receiver, bladders and flooded vaults particularly, that will be used in the Hydrostor systems described earlier. With these, the hydrostatic pressure of the water column above the receiver is the source of the static load; what keeps the pressure constant inside the reservoir is the water above and around it. The flooded vaults operate on a different principle, much like aquifers. The compressed air displaces water in a cavern. The water pressure outside of the vault or cavern (at the boundary of the air and the water) is the same as the air pressure inside, so it maintains a steady pressure inside the receiver. When air mass is removed, the water flows back into the space where the air used to be. The deeper the water, the higher the pressure.

The receiver used for modeling purposes is of a rigid boundary vessel. It has two ports: one inlet and one outlet. Compressor operations do not occur simultaneously with turbine operations (no filling and extraction at the same time). The tank is assumed to be well insulated and buried underground, with no significant heat transfer occurring during operation. This is a key assumption, which can be modeled differently in further iterations. It has not been determined what material the reservoir will be constructed from, whether steel, concrete or something else. This, and the surrounding materials and conditions will be crucial to determining what heat transfer effects, if any, may occur in a real system. This has been studied and modeled for the Huntorf CAES plant[31]. The cavern wall is shown to have heat sink-like properties: during compressor operations, the wall absorbs heat, and after a certain point during turbine operations, the air gets colder and the wall gives off heat. The caverns at Huntorf are, of course, not insulated any more than with the earth around them. Along with modeling heat transfer, future work should include the addition of a moving boundary (constant pressure) receiver.

# CHAPTER 2

## Relevant Thermodynamic Theory

Principles are the backbone of any model; if they are wrong or used incorrectly, the model will likely produce less-than-helpful information. The model may do this if everything included is correct, but with too few principles included. Despite this, assumptions and simplifications must be made. It is the job of the modeler to make the decision between the relevant and the extraneous. Variables and processes may be included in one version that are removed from the next because the difference is negligible while phenomena and behaviors of a real system may send the modeler scrambling to find suitable equations. A model may be comprehensive and complex, but it will probably not yield the same amount or level of information that a real, complete system can provide. We do the best we can then fix the parts that need improvement. The purpose of this chapter is to outline and explain the thermodynamic processes included in the system model as well as the principles behind them.

To begin, an overview of the compressed air energy storage system will be provided in order to give some perspective on how the pieces of the system interconnect, then the properties of dry air and behavior of ideal gases will be explained. Next, the principles that underlie the process of intercooled air compression[7] will be discussed, followed by the psychrometric principles[32] used for dealing with humid air and condensate. The relevant principles affecting the receiver tank[7] will be discussed, regarding the effects of mass addition and removal. Finally, the theory behind heated gas expansion through a turbine[7] will be covered.

### 2.1 System Overview

The compressed air energy storage system, shown in Figure 16 below, is intended to use excess electricity to power one or more electric motors that will run one or more intercooled centrifugal air compressors. The compressed air will be sent through an aftercooler, then into a single storage receiver via conduit piping. The receiver, which is assumed well insulated, is located underground in order to isolate the system from temperature fluctuations as well as to provide a greater margin of safety in case of wall rupture. Temperature fluctuations may be due to changing weather conditions, solar energy influx, or proximity to heat sources. Wall rupture may be caused by material fatigue, catastrophic overpressure, or damage. At periods of deficient energy production, i.e. peak loading, air will be released from the storage vessel via conduit piping into one or multiple expansion turbines. Each turbine will have a pre-inlet heat exchanger to add heat to the air prior to expansion. The source of heat will depend on availability, required magnitude and application. These sources can potentially include fuel combustion, stored heat rejected from the compression process, or waste heat from another process (industrial, electricity generation). The turbines will turn electricity generators, which will contribute to supplying electrical system loads.

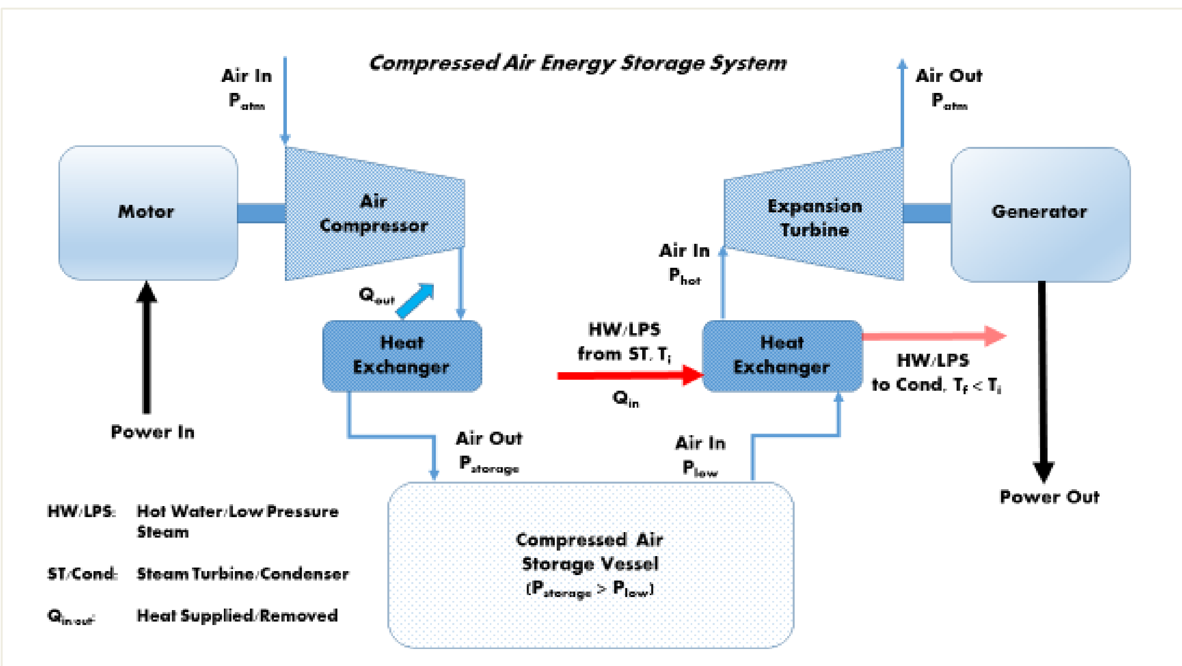


*Figure 16: Basic Compressed Air Energy Storage System Components and Layout*

## 2.2 The Properties of Dry Air and Ideal Gas Behavior

Atmospheric air is a mixture of gases, including but not limited to nitrogen, oxygen, argon, carbon dioxide, water vapor, and trace amounts of other compounds and elements. Generally for calculation purposes, a molar mass of 28.965 *kg/kmol* is used for dry air. This is a crucial point, as the contribution from water is not included in this molar mass; moist air will be covered in the psychrometrics section. The mass is calculated proportionally, with the molar masses of each element/molecule. Diatomic nitrogen ($N_2$) has an atomic weight of 28.013 *kg/kmol* and a proportion of 78.1 %, diatomic oxygen ($O_2$) is 31.998 *kg/kmol* and 20.95 %, monatomic argon (Ar) is 39.948 *kg/kmol* and 0.93 %, and carbon dioxide ($CO_2$) is 44.010 *kg/kmol* and 0.03 % (all proportions are approximate and neglect other trace elements)[33]. Therefore:

$$\left[(0.7809\times 28.013)_{N_2}+(0.2095\times 31.998)_{O_2}+(0.0093\times 39.948)_{Ar}+(0.0003\times 44.010)_{CO_2}\right]\frac{kg}{kmol}$$
$$=28.965\frac{kg}{kmol}$$

As a mixture of these gases, primarily nitrogen and oxygen, dry air behaves as an ideal gas at *high* temperatures. High temperatures are those that are higher than the critical temperature of a substance, above which the substance will always be in gaseous form regardless of its pressure. Relative to the critical temperature of air (132.5 *K*, -221°*F*), all temperatures in the scope of the system will be high.

Ideal gas behavior means that regardless of the gas pressure, the individual molecules do not interact with each other aside from elastic collisions. The behavior of the gas can be explained with the ideal gas equation of state:

$$PV = mR_{gas}T \tag{1}$$

$P$ is the absolute pressure [*kPa*] of the substance; $V$ is the total volume [$m^3$] of the substance that the mass occupies, given its temperature and pressure; $m$ is the total amount [*kg*] of substance mass; $R_{gas}$ is the gas constant [$kJ/kg\text{-}K$] for the substance; and $T$ is the absolute temperature [$K$] of the substance.

The absolute pressure (force per area) includes the ambient or atmospheric pressure. If temperature is kept constant, at absolute pressures lower than the ambient pressure we have a partial vacuum, where fewer molecules (mass) than “normal” are present to interact or collide in a given space. At zero absolute pressure, either a perfect vacuum must exist, or all particle motion must stop. A perfect vacuum is a state necessarily devoid of matter; no particle interactions occur because there are not any particles! If matter does exist, but is motionless, there will be no temperature. In both theoretical cases, there will be no pressure, as there will be no collisions to impart the force over area.

The absolute temperature of a substance is a measure of the average kinetic energy of its molecular motion; how fast, overall, the molecules are moving. The volume of a substance is the amount of space its mass occupies, whether or not the substance is physically contained. This is not to say that a parcel of gas, with no container to define it, is not free to move about. This is merely to say that for any given combination of pressure and temperature (above the critical temperature), a certain amount of mass will occupy a certain volume (whatever shape that volume may take). The mass of a substance is the amount of its constituent matter that is present. The gas constant is a coefficient necessary to maintain the proportionality of the equation. $R_{gas}$ is defined as:

$$R_{gas} = \frac{R_u}{M_{gas}} \tag{2}$$

$R_u$ is the universal gas constant, which is the same for all substances, equal to 8.31447 $kJ/kmol\text{-}K$. $M_{gas}$ is the molar mass of the substance, as discussed above for air.

Next, a very important property is the specific heat of air. Generally, specific heat is the amount of energy required to increase the temperature of a certain amount of mass *(kg)* of a substance one degree, or $kJ/kg\text{-}K$. There are two types of specific heat: constant volume specific heat *($c_v$)*, and constant pressure specific heat *($c_p$)*. Constant volume specific heat requires that the substance volume remain constant while the process occurs. Not surprisingly, constant pressure specific heat requires the substance pressure to remain constant. The difference between these values is always the gas constant $R_{gas}$ of an ideal gas:

$$c_v = c_p - R_{gas} \tag{3}$$

The ratio of specific heats *(k, [unitless])* is a convention that becomes useful in calculations used later, and is expressed as:

$$k = \frac{c_p}{c_v} \tag{4}$$

Finally, the specific internal energy *(u, [kJ/kg])* and specific enthalpy *(h, [kJ/kg])* will be explained. Both properties describe magnitudes of energy (*kJ*) per unit of mass (*kg*) for a substance, and both properties are strictly temperature-dependent for an ideal gas. The specific internal energy of a gas includes all of the thermal and chemical energies present in one *kg* of the substance. Specific enthalpy includes everything that *u* is composed of, but adds the product of the substance's absolute pressure and specific volume *(v, [$m^3/kg$])*, the reciprocal of density:

$$h = u + Pv \tag{5}$$

The *Pv* term [$kPa(m^3/kg) = kN/m^2(m^3/kg) = kN\text{-}m/kg = kJ/kg$] can be interpreted as the energy required to "make room" for the substance in its surroundings, to displace its environment. Since the working fluid in the system is air, an ideal gas, dividing the mass from equation (1) gives:

$$Pv = R_{gas}T$$

The right side is a constant. Substituting the above into equation (5) yields:

$$h = u + R_{gas}T$$

Since the specific internal energy and specific enthalpy are strictly temperature-dependent for ideal gases, changes in each with respect to temperature can be related to the temperature-dependent specific heats:

$$\frac{du}{dT} = c_v\left(T\right) \tag{6}$$

$$\frac{dh}{dT} = c_p\left(T\right) \tag{7}$$

Integrating both:

$$\int_{u_2}^{u_1} du = \int_{T_1}^{T_2} c_v\left(T\right) dT$$

$$\int_{h_2}^{h_1} dh = \int_{T_1}^{T_2} c_p\left(T\right) dT$$

If changes in temperature are relatively small, changes in the specific internal energy or specific enthalpy can be expressed *approximately* and with minimal error using average values of specific heat:

$$\Delta u = \frac{c_v\left(T_2\right) + c_v\left(T_1\right)}{2}\left(T_2 - T_1\right) \tag{8}$$

$$\Delta h = \frac{c_p\left(T_2\right) + c_p\left(T_1\right)}{2}\left(T_2 - T_1\right) \tag{9}$$

In this way, the functional relationship between the two specific heat endpoints does not need to be known. The integral is approximated by the midpoint rectangle approximation method. The closer the endpoints, the lower the error and vice versa. Experimentally-obtained tabular data can also be used to determine these property changes, which is the method used in the model.

## 2.3 Intercooled Air Compression

### 2.3.1 Compression Methods

Air compression can be modeled in one of three ways: isothermally, isentropically, or polytropically, shown in Figure 17. The first thing that must be understood about any gas compression is that it creates heat. The main difference among these processes is how the heat is dealt with.

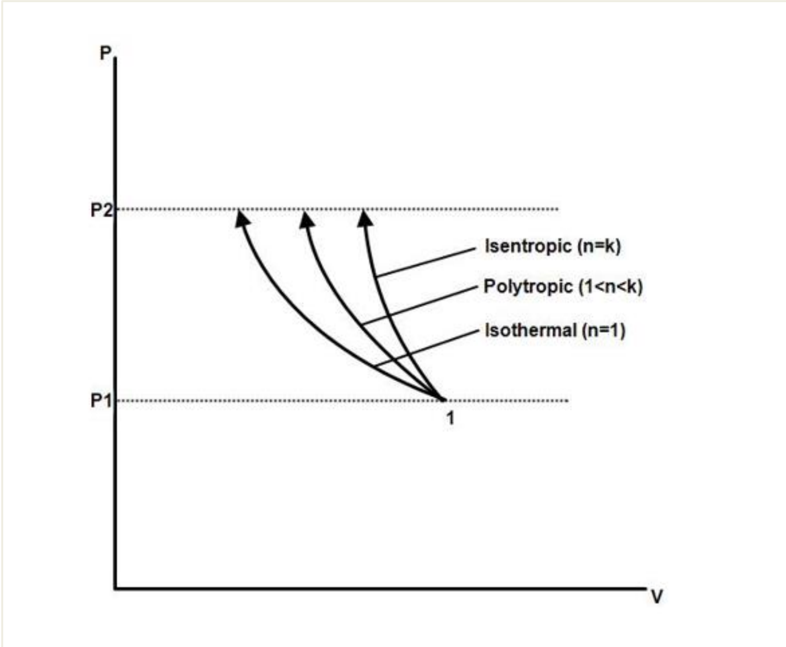


*Figure 17: P-v Diagram of Isothermal, Polytropic and Isentropic Compression[7]*

The specific form of the ideal gas equation of state:

$$Pv = R_{gas}T$$

the isentropic relations for ideal gases and constant/average specific heats:

$$\frac{T_2}{T_1} = \left(\frac{P_2}{P_1}\right)^{\frac{k-1}{k}} = \left(\frac{v_1}{v_2}\right)^{k-1} \tag{10}$$

and the expression for reversible work in a steady flow (open) system:

$$w_{rev} = \int_{P_1}^{P_2} vdP = \Delta h \tag{11}$$

must be considered for the three processes.

During isothermal compression, air is compressed with the instantaneous removal of heat; the gas stays at a constant temperature during the entire process. This method of gas compression requires the least amount of energy input. For isothermal processes, since the temperature does not change, the specific enthalpies for the initial and final states cannot be used to calculate the work required. We must use:

$$v = \frac{R_{gas}T}{P}$$

Therefore:

$$w_{rev,iso} = \int_{P_1}^{P_2} R_{gas}T \, \frac{dP}{P}$$

The expression for specific isothermal compression work *($w_{comp,iso}$, [kJ/kg])*, from one pressure to another, is:

$$w_{comp,iso} = R_{gas}T \ln \frac{P_2}{P_1} \tag{12}$$

Isentropic compression is exactly the opposite of isothermal compression with regards to heat. There is no heat removed in the process. Any and all heat generated by the compression process is kept within the air, requiring the most energy input (of the three methods) for the same pressure ratio. More energy is required to compress hotter air. There is certainly a change in temperature, the change in enthalpy can be used to find the required compression work. The expression for isentropic compression is: $w_{comp,ise} = \Delta h = \dfrac{kR_{gas}\left(T_2 - T_1\right)}{k-1}$ (13)

Since $k = c_p / c_v$ and $k - 1 = R_{gas} / c_v$ , equation (13) is really is equivalent to equation (9), for the change in enthalpy using an average constant pressure specific heat $c_p$. Equation (13) may seem like a convolution of an already simple expression, but there are two good reasons for doing this. First, specific heat ratio values, since they are ratios, change less over temperature ranges than do the specific heat values. Second, if the final temperature is unknown, equation

(13) can be put into terms of the initial temperature, the pressure ratio, and the specific heat ratio:

$$w_{comp,ise} = \frac{kR_{gas}T_1}{k-1}\left[\left(\frac{P_2}{P_1}\right)^{\frac{k-1}{k}} - 1\right] \tag{14}$$

by substituting:

$$T_2 = T_1\left(\frac{P_2}{P_1}\right)^{\frac{k-1}{k}}$$

from equation (10). Equation (14) may require iterations to determine the suitable average value for $k$, as it cannot be calculated without previously knowing the outlet temperature.

Polytropic compression has a similar expression for work:

$$w_{comp,poly} = \frac{nR_{gas}\left(T_2 - T_1\right)}{n-1} = \frac{nR_{gas}T_1}{n-1}\left[\left(\frac{P_2}{P_1}\right)^{\frac{n-1}{n}} - 1\right] \tag{15}$$

where $n$ is the ratio of specific heats, falling between the values of 1 (indicating isothermal compression) and $k$ (indicating isentropic compression). Polytropic compression has some but not all heat removed *during* the compression process. It requires less work than isentropic compression but more work than isothermal compression.

### 2.3.2 Staged Intercooling

Obviously higher temperature air takes more work to compress than colder air. We can see this by inspection of the expressions for compressor work, simply by increasing the inlet temperature. Why would anyone choose the process that requires more work? Why aren't all compressors isothermal compressors if the goal is to increase the pressure most efficiently? The easy answer is that isothermal compression or anything close is very difficult to actually perform, particularly on a large scale. Compression cooling will result in polytropic compression, at best. However, with staged air compression and cooling between stages, isentropic compression work can be reduced significantly.

Staged compression merely breaks the compression process into smaller steps, as shown in Figure 18 below.

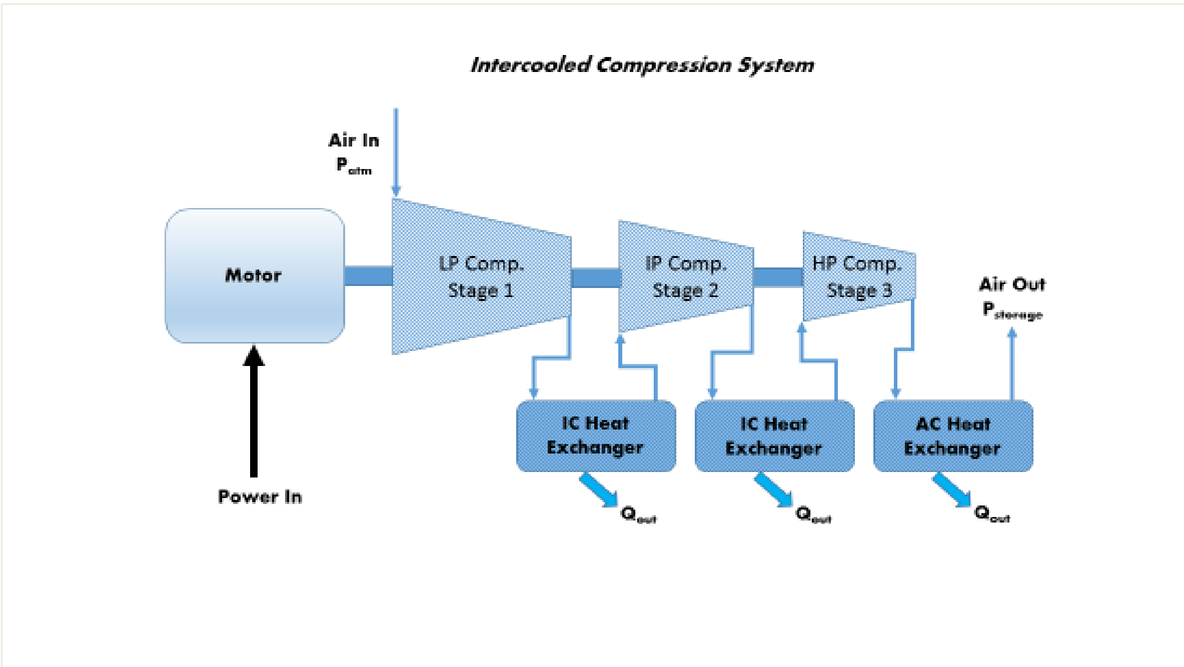


*Figure 18: Staged Intercooled Compression System with Aftercooling*

Instead of compressing air from 14.7 *psi* to 214.7 *psi* in one step (a pressure ratio of 14.6:1), which may require a very large and sturdy compressor, the air may be compressed in multiple stages to split up the work. The first stage may compress the air from 14.7 psi to 58.2 psi, then a second stage may further compress the air at 58.2 psi to 214.7 psi. Each stage has a compression ratio of 3.82:1 instead of 14.6:1. It will take more work to compress the air in the second stage, because the air is hotter, but the machinery does not have to accomplish it all in one step. Equal stage ratios can be found by first calculating the overall compression ratio:

$$r_c = \frac{P_{final}}{P_{initial}} \tag{16}$$

The stage ratio $r_p$ can be calculated with $r_c$ and the number of stages $s_c$:

$$r_p = r_c^{\frac{1}{s_c}} \tag{17}$$

This way, each compression stage will accomplish the same pressure rise as all the others. The work for the process can be calculated with the sum:

$$w_{comp,ise} = \sum_{i=1}^{N} \left( \frac{k_i R_{gas} T_i}{k_i - 1} \left[ \left( \frac{P_{i+1}}{P_i} \right)^{\frac{k_i - 1}{k_i}} - 1 \right] \right)_i \tag{18}$$

$N$ is the total number of stages, $T_i$ is the stage inlet temperature and $P_i$ is the stage inlet pressure.

The beauty of staged compression comes when the air between stages is cooled to the initial inlet temperature. This way, if each stage has the same pressure ratio and inlet conditions, each stage will require the *exactly the same* amount of work to compress the air (Figure 19).

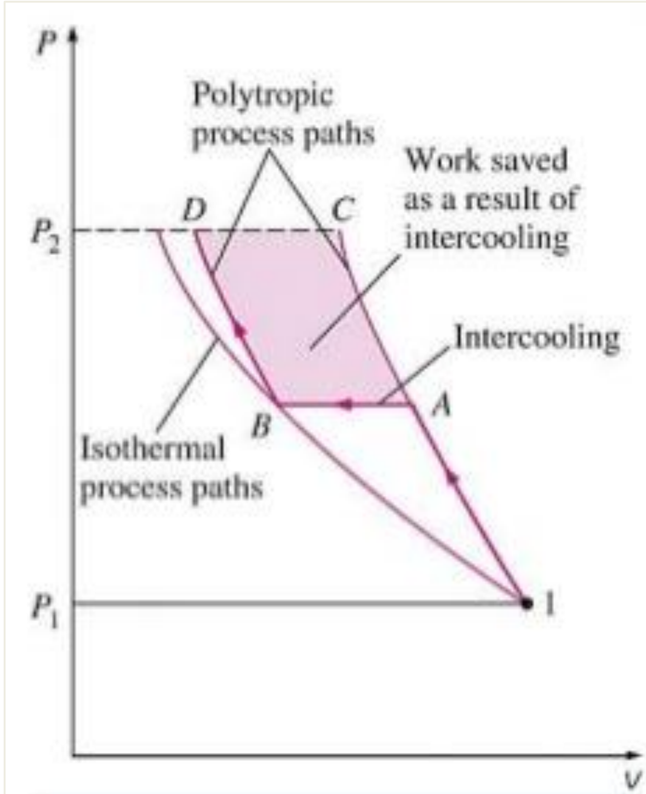


*Figure 19: Staged Compression with Intercooling (Polytropic Processes Shown)[7]*

Equation (18) becomes:

$$w_{comp,ise} = N\left(\frac{kR_{gas}T_1}{k-1}\left[r_p^{\frac{k-1}{k}} - 1\right]\right) \tag{19}$$

As the number of compression stages goes to infinity, the lower limit of compression work goes to the single stage isothermal compression work. The question now becomes, why not make compressors with hundreds of intercooled stages? This would require a very complex and expensive machine; the contribution of each additional stage beyond three or four is negligible (at least for centrifugal compressors) compared to the added machinery and maintenance costs. The method of compression used in the model is staged isentropic compression with intercooling and aftercooling. Aftercooling is the cooling process that occurs after the final stage of compression.

### 2.3.3 Isentropic Compressor Efficiency

All of the expressions for compressor work to this point have been for reversible compressor work, which assumes perfect machinery without losses. This is impossible, and so is true isentropic compression. So it seems it is all for naught, and we should quit now. Never! We must, however, introduce the concept of isentropic compressor efficiency *($\eta_c$, [unitless]).* This can be expressed in a number of ways, using enthalpies or temperatures, but the basic representation is:

$$\eta_c = \frac{\text{Isentropic Compressor Work}}{\text{Actual Compressor Work}} = \frac{w_{comp,ise}}{w_{comp,act}} \tag{20}$$

$\eta_c$ can take any value between 0 and 1, and takes account of all the irreversibilities and entropy generation that occur in real processes due to friction, heat transfer, etc. The actual temperature and specific enthalpy of the air exiting each compressor stage will be higher than it would be if the process were reversible. The actual process will require more work to perform the same task than the ideal process.

## 2.4 Psychrometrics

Some water vapor, the amount of which literally depends on the weather, exists in atmospheric air. This is the air that is compressed, and humid air has different properties than dry air. To start, dry air can be compressed further than water vapor. Once water vapor condenses to liquid, it cannot be compressed further. A related issue is the critical temperature of water. It is 647 *K* (705°*F*) which is not entirely out of the possible range of temperatures for vapor in the system. This means that water could be present in the system at some point as a liquid, or even worse, as a solid. Just as the compression process heats up air, humid or not, the expansion process will tend to cool the air. If this cooling brings the temperature of the expanding gas down to the water vapor's dew point or freezing point, liquid water or ice could condense or solidify inside the turbine. This is likely to be quite damaging to the machine, as water is incompressible. To avoid this, the water that can condense should be removed from the compressed air prior to expansion.

For this model, compression is modeled with dry air, as it is not expected that the relatively small amount of water in the air will cause significant increases (order of magnitude) in energy input. If this point needs to be revised, new code may be written to compensate for the compression of moist air. As the water vapor will contribute only approximately 1% to the total air mass, its effects are expected to be negligible. The model does estimate the amount of water removed, as well as the minimum temperature the compressed air can reach before condensation occurs. This minimum temperature is most relevant to the expansion process.

Hotter air can hold more water vapor than colder air, and the dew point temperature is the temperature at which the air is saturated with water vapor. This is the point of 100 % relative humidity *(RH)*. Relative humidity is defined as:

$$RH = \phi = \frac{m_v}{m_g} = \frac{P_v}{P_g} \tag{21}$$

$m_v$ is the amount of water vapor mass in the air, and $m_g$ is the amount of water vapor mass in the air at saturated conditions. $P_v$ is the partial pressure of the water vapor in the air, and $P_g$ is the partial pressure of water vapor in saturated air. If the air gets any colder at 100 % *RH*, water will begin to condense until the air has reached a saturated equilibrium once more. If the air gets hotter, it will cause the air to become less saturated, able to hold more water vapor. The relative humidity is not the same as the humidity ratio *($W_s$)*, which is defined as:

$$W_s = \frac{m_v}{m_a} = 0.62198\frac{P_v}{P_a} \tag{22}$$

$m_v$ is the water vapor mass in the air, $m_a$ is the mass of dry air, $P_v$ is the partial pressure of the water vapor in the air, and $P_a$ is the partial pressure of the dry air. $P_a$ can also be expressed as

$$P_a = P - P_v$$

$P$ is the total air pressure. The humidity ratio is the ratio of water to dry air, not a percentage of saturation like relative humidity.

When air is compressed, everything in it gets hotter. Potentially, if the air is very humid, temperatures of outlet air will be lower because the specific heat of water vapor is higher than that of dry air. This means that more energy has to be consumed per *kg* of water in order to raise its temperature one degree *(C, K)* than for air. The pressure also increases, which means that each of the partial pressures increases as well. If the total pressure increases two-fold, so do the partial pressures for air and water vapor. The key is when this compressed air is cooled back to the inlet temperature. If this air temperature goes below its dew point temperature, some of the water vapor will condense. The air will be fully saturated unless it is heated again. If the air is further cooled at this pressure, more condensation will occur. The humidity ratio of the inlet air $(W_{s,in})$ is found from the dew point temperature, which can be determined from the relative humidity and the dry-bulb temperature $(T_{db}, [K])$ of the air. The dry bulb temperature is the temperature most commonly used; it is the air temperature. If the dry-bulb and dew point temperatures coincide, the air is saturated, 100 % RH.

The vapor pressure $(P_{v,cooled})$ and total pressure $(P_{outlet})$ of the cooled, compressed air is used to find its humidity ratio $(W_{s,cooled})$ with equation (22):

$$W_{s,cooled} = 0.62198 \times \frac{P_{v,cooled}}{P_{outlet} - P_{v,cooled}}$$

This ratio can be used to find both the amount of condensate removed $(W_{s,rem})$ as well as the minimum temperature $(T_{min})$ the air can be cooled to, at atmospheric pressure, before condensation occurs. The water removed is:

$$W_{s,rem} = W_{s,in} - W_{s,cooled} \tag{23}$$

If the *compressed* air is allowed to cool further, below the intercooler/aftercooler temperature, more water will condense out. The minimum temperature is valid only at atmospheric pressure; if the temperature drops below any minimum temperature for its corresponding pressure, condensation will occur. The only way to ensure this does not occur is to aftercool the compressed air more or to use a desiccant system to further remove water from the system. Dehumidification beyond intercooling and aftercooling to the inlet temperature was not included in this model.

### 2.5 Mass Storage

The filling and emptying of a receiver vessel can be modeled as a uniform flow process. This assumes that the fluid flows at the inlet and outlet are uniform, and that the fluid properties do not change with time or position over the inlet or outlet cross-sections. The effects of water vapor in the air are assumed negligible, and only dry air properties are considered. This is important, as even though water condensate has been removed from the compressed air through intercooling and aftercooling, the air entering the receiver is saturated in its compressed state.

Mass is added and removed depending on the operation of the system. The internal energy and temperature of the tank are changed based on the addition of mass at the inlet stream temperature or subtraction of mass at the receiver temperature. Both the inlet and outlet have steady flow; the inlet stream properties do not change, and is assumed to have constant temperature, pressure and mass flow rate. The outlet stream temperature depends on the temperature of the receiver, while the design pressure and mass flow rate for the inlet of the expansion/turbine system remain constant.

Essentially, this receiver can be modeled using the First Law of Thermodynamics for an open system, shown below in Figure 20. To start, the mass $(m_{CV})$ of the receiver (control volume) at any time is:

$$m_{CV} = m_{CV,initial} + m_{inlet} - m_{outlet} \tag{24}$$

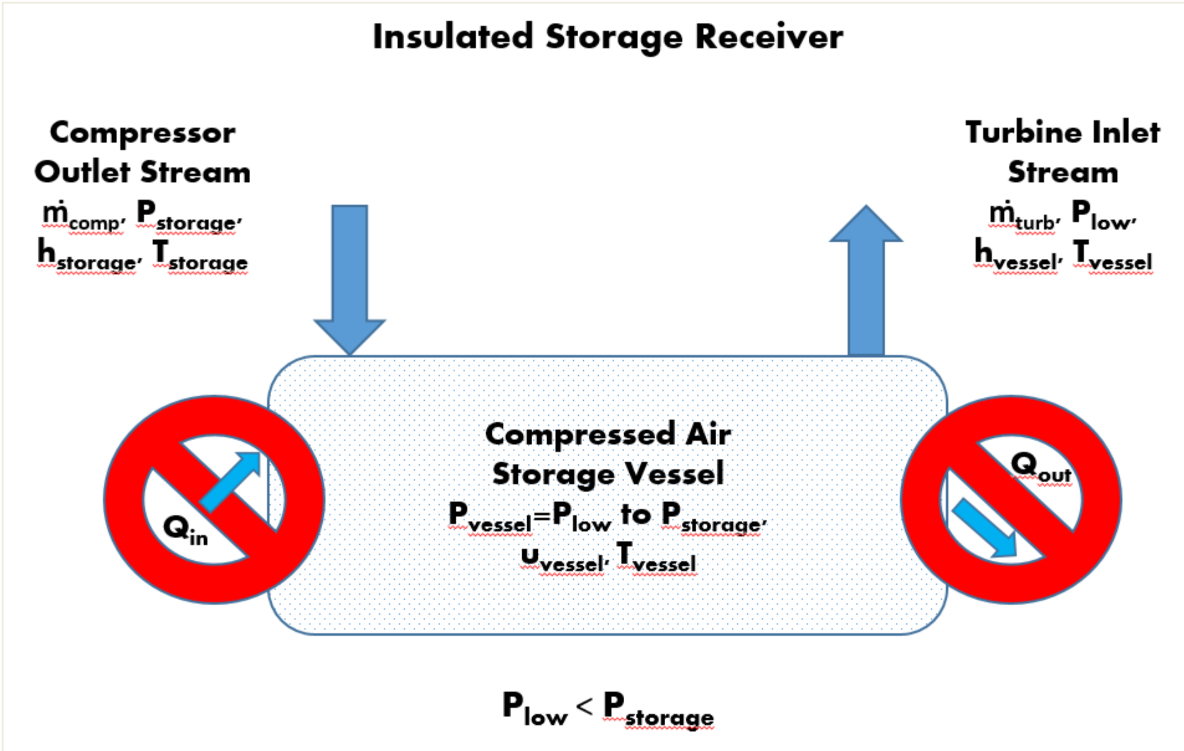


*Figure 20: Insulated Storage Receiver Diagram*

The receiver has an initial mass *($m_{CV,initial}$)* before any mass addition or subtraction begins. The mass flow rates of the inlet stream and outlet stream, over the period (time step) of "filling" or "emptying," respectively, produce the inlet or outlet mass terms *($m_{inlet}$, $m_{outlet}$)*. Using the First Law, we can say that the change in total internal energy for the control system is equal to the net heat *($Q_{net}$)* minus the net work *($W_{net}$)* plus any flow energy from inlet streams minus any flow energy from outlet streams:

$$U_{CV2} - U_{CV1} = Q_{net} - W_{net} + \left[ \sum_{i=1}^{N} m_i \left( h + ke + pe \right)_i \right]_{inlets} - \left[ \sum_{j=1}^{M} m_j \left( h + ke + pe \right)_j \right]_{outlets} \tag{25}$$

In this situation, it is assumed the receiver is well insulated from its surroundings and no heat transfer occurs through the tank wall. There is no boundary work occurring with this rigid tank, and no mechanisms exist to supply work to or extract work from the control volume. The receiver is not moving with respect to its surroundings, and kinetic and potential energies can be neglected. There is one inlet port and one outlet port on the receiver, and the energy added to or removed from the system is due to the enthalpies of the inflowing or outflowing mass, respectively. Therefore, equation (25) condenses to:

$$U_{CV2} - U_{CV1} = (mh)_{inlet} - (mh)_{outlet} \tag{26}$$

and becomes:

$$(mu)_{CV2} = (mu)_{CV1} + (mh)_{inlet} - (mh)_{outlet} \tag{27}$$

Since both the specific internal energy and specific enthalpies of the inlet and outlet streams are temperature-dependent, they can be interpolated from tabular values. The transient receiver and exit stream temperatures can be found as long as the mass flow rates and initial control mass are known. The transient receiver pressures can be calculated with the ideal gas equation of state, as long as the receiver temperature and mass are known.

## 2.6 Heated Air Expansion

Well folks, this is where the rubber meets the road; when we see if we can get any energy back out of our cooled compressed air. Heated isentropic expansion is exactly like cooled isentropic air compression in reverse. When compressed air is allowed to expand through an expander or turbine, the expansion does work on the impeller or rotor, forcing it to turn a shaft. The volume of the air increases, the pressure decreases, and the temperature decreases, subject to the ideal gas law. If the compressed air enters the turbine at the ambient temperature and allowed to expand to the atmospheric pressure, the uncompressed air at the outlet will be much colder than the surroundings. The air temperature could be lower than the pressure dew point temperature at any point in the process, which means that condensation may occur in the turbine.

Governed by the isentropic relations, the work out of a turbine can be expressed as the change in enthalpy between the inlet and the outlet:

$$w_{turb,out} = h_{inlet} - h_{outlet} \tag{28}$$

Unfortunately, unless the heat of compression has been stored and is able to be added back to the air before it enters the turbine, the work extracted from the process will be significantly less than the work expended to compress it. Waste heat from an industrial or generation process may be used, but it must be at a high enough temperature and in sufficient quantity. This is one of the reasons why the Brayton Cycle, in practice, combusts fuel with the compressed air to create hot exhaust gases to send through the gas turbine. The temperature drives the enthalpy, and the greater the temperature gradient over the expansion process, the more work that will be available.

Similar to compressors, turbines can be split into stages with intermediate heat exchangers, as shown in Figure 21.

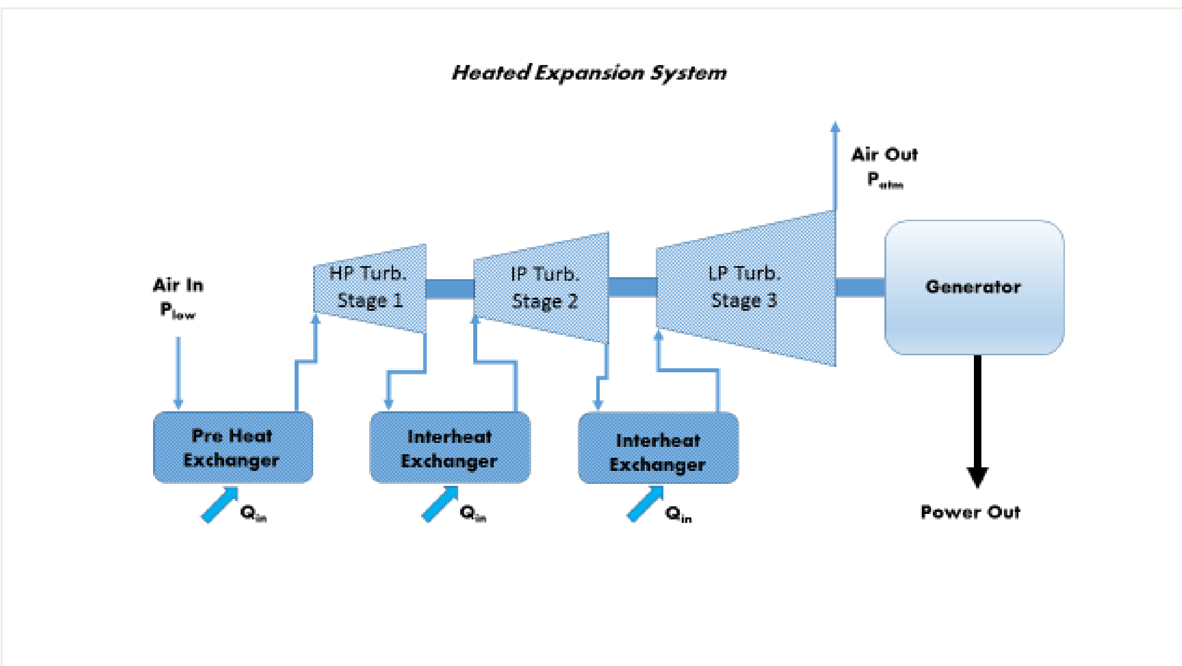


*Figure 21: Staged Heated Expansion System (Turbines & Heat Exchangers)*

The entire pressure ratio of the process is:

$$r_t = \frac{P_{final}}{P_{initial}} \tag{29}$$

which is a fraction $\left(0 \le r_t \le 1\right)$, can be split into equivalent stage ratios *($r_s$)*:

$$r_s = r_t^{\frac{1}{s_t}} \tag{30}$$

Each turbine stage is responsible for a discrete pressure gradient drop. If the air is heated to the inlet temperature of the first stage, each turbine will output an equal amount of work, just like the reverse of intercooling staged compression. Therefore, not as much heat will need to be added to the air at once in order for the most work to be extracted from the expansion process. The inlet temperatures will be lower, but the outlet temperatures will be higher. This is due to the stage ratio term *($r_s$)* in the total sum of stage work expression for a turbine:

$$w_{turb,ise} = N\left(\frac{kR_{gas}T_1}{k-1}\left[1 - r_s^{\frac{k-1}{k}}\right]\right) \tag{31}$$

As more stages are added, the larger $r_s$ becomes, closer to the *upward limit* of one; this is the point of isothermal expansion. This is the reverse of what happens with each added compression stage: the stage ratio $r_p$ gets closer to the *lower limit* of one, the point of isothermal compression. Adding compression stages causes the compressor outlet temperature to get lower.

As with compressors, turbines cannot contain the expansion process isentropically or reversibly. There must be some entropy generation, irreversibilities or losses. Just like the isentropic compressor efficiency (only reversed), the isentropic turbine efficiency is a ratio of work:

$$\eta_T = \frac{\text{Actual Turbine Work}}{\text{Isentropic Turbine Work}} = \frac{w_{turb,act}}{w_{turb,ise}} \tag{32}$$

The outlet temperature will be hotter for an actual process than it will for isentropic process for the same inlet temperature.

$$\eta_T = \frac{T_1 - T_{2,act}}{T_1 - T_{2,ise}}$$

Since $T_1 - T_{2,ise} \geq T_1 - T_{2,act}$ is necessarily true for $\eta_T \leq 1$, $T_{2,act} \geq T_{2,ise}$.

The power of the reheat source *($kW_{REHEAT}$)* must be equal to the amount of specific enthalpy/heat *($\Delta h_{REHEAT}$)* needed per stage at the mass flow rate $(\dot{m})$:

$$kW_{REHEAT} = \dot{m} \times \Delta h_{REHEAT} = \dot{m} c_p \left(T_1 - T_2\right) \tag{33}$$

All that needs to be done is to reheat the inlet temperature sufficiently so that the outlet temperature will be the ambient temperature, right? We can do this, but the work output may not be very high. This is because the process is isentropic, and we must incorporate temperature-dependent relative pressures:

$$\frac{P_2}{P_1} = \frac{P_{r2}}{P_{r1}} \tag{34}$$

The inlet or outlet temperature must be known in order to find its corresponding relative pressure, and the stage ratio must be known. These can be used to find the unknown relative pressure and its corresponding temperature. Since the enthalpy is temperature-dependent as well, the change in enthalpy can be calculated with the endpoint temperatures. Despite this, the range over which the temperatures correspond to the pressure gradient may not yield the range of enthalpies we desire. The higher the inlet temperature, the greater the enthalpy change, but the higher the outlet temperature (wasted energy).

For example, if we desire an outlet temperature of 300 *K*, and have a turbine stage ratio ($r_s = P_2 / P_1$) of 1/3, the $P_{r2}$ value will be:

$$P_{r1} = \frac{P_{r2}}{r_s} = \frac{1.3860}{1/3} = 4.158$$

This corresponds to an inlet temperature of approximately 410 *K*, giving a temperature gradient of 110 *K*. The change in specific enthalpy will be approximately 111 *kJ/kg*. Conversely, if an inlet temperature of 800K were chosen, the $P_{r2}$ value will be:

$$P_{r2} = r_s \times P_{r1} = \frac{1}{3} \times 47.75 = 15.92$$

This corresponds to an outlet temperature of approximately 595 *K*, giving a temperature gradient of 205 *K*. The change in specific enthalpy is approximately 220 *kJ/kg*. Both processes are over the same pressure gradient. We cannot simply tailor the change in enthalpy required to the specific outlet or inlet temperature we need. It would be wonderful to take a requirement of 500 kJ/kg per stage and a desired outlet temperature of 300 *K*, and end up with an inlet temperature of 790 *K*. This will work if you have a stage ratio of approximately 1/33, but not something lower. This is one of the reasons that optimization for stage number and reheat is not as straightforward as it seems, depending on your initial conditions and constraints.

# CHAPTER 3

## Computer Modeling

Computer modeling of physical systems is often the most economically feasible way of predicting behaviors and outcomes for multiple system configurations without having to invest the resources (most commonly time and money) to study the real physical system. Unfortunately, no model can fully define every characteristic and behavior of a given system; the late British statistician George Box once wrote, "all models are wrong, but some are useful[34]." My advisor, Dr. Stephen Terry, stated this repeatedly during my research and programming process. Models can become extremely complex and complicated quickly, out of an effort to be complete and to take every factor into account. Despite our ability to think and reason, we humans are still beholden to our physical limitations, perceptions, bias and overwhelming ability to make mistakes; to err is human. We can make models as simple or as convoluted as we desire, but the goal of saving time and money could be defeated by endless additions towards a model's completeness. What we need is something that satisfies, to an acceptable extent, the goals we would fulfill by observing the actual system. The model should output reasonably accurate answers that can be analyzed and used in a decision-making process. It was the primary intent of this endeavor to provide a model that is useful and hopefully that goal has been accomplished. The following sections will explain the model structure as well as each part in detail.

### 3.1 Fortran and Excel

The language of choice for this compressed air storage model is Fortran, specifically the Fortran 95 revision, with Microsoft Excel used as a check for calculations. It should be noted that having a duplicate system in another language or system is NOT a method of validating the model; it is a method of reflection. If the model is to be validated, it must be validated against real data from a physical system to quantify the accuracy of the model results. Fortran was chosen for its ability to interconnect with an existing model of a nuclear reactor-based electricity generation system, also written in Fortran. One of the constraints for the modeling process was that it model a storage system that was directly connected to the generation plant. Excel was chosen as the check program for its ubiquity, ease of use, and ability to chart, table and plot data.

## 3.2 Fortran Model Structure

The general structure of the model, written in Fortran, is based on subroutines. Initially the design included one main program that ran all of the branch subroutines, but this was changed to streamline data input. In the original arrangement, variable values such as temperature and pressure inlet conditions, were requested and the user would input them as the program progressed. The format of model was changed to set the main program as a data input page, the old main program as a main subroutine to the data input page, and all of the other subroutines as branches off of this main subroutine. The format can be changed back to the original configuration for general use, but for repeated use the old format is too cumbersome.
The program proceeds, in a general sense, as follows: the main data input program feeds the predetermined user input data to the main subroutine, which it calls. This main subroutine then reads in and stores air property data from a data file. The main subroutine calls a branch subroutine to calculate the specific heats of air at the specified compressor inlet conditions. The main subroutine then calls the compressor work subroutine, which calculates and returns the energy per kilogram required to compress air at the given conditions. The main subroutine calculates the specific enthalpies for the compressor inlet and outlet streams before checking the existence of the specified excess power data file that will be used for sizing the system. If this data file does not exist, a data entry subroutine is called to allow the user to input excess power data to a new file. With this file open, another subroutine is called by the main subroutine to determine a minimum receiver size for the system, using the excess power data provided. The user is given the opportunity to enter another value for the receiver volume to be used in the rest of the program. Next a humidity calculator subroutine is called to determine

the humidity ratio of, pressure dew point temperature for, and moisture content removed from the compressed air. The next subroutine to be called calculates the change in receiver temperature as the receiver is being charged. Finally, at least for the sizing model, the main subroutine calls an expander subroutine which calculates the amount of air reheat needed, inlet and outlet temperatures, as well as the power and energy output for one through six turbine/heat exchanger couples. For the running version of the model, which takes the sized system and simulates a running system for one year, two more subroutines are called. The first is the subroutine that handles the runtime calculations, and the other is based on the expander subroutine. This second subroutine is called inside the runtime subroutine, so it is a subroutine inside a subroutine inside yet another subroutine. This nested subroutine calculates the power and energy generated by the expander array. All calculated data is output to data files for further analysis. Each subpart, shown in Figure 22 below, will be explained in greater detail.

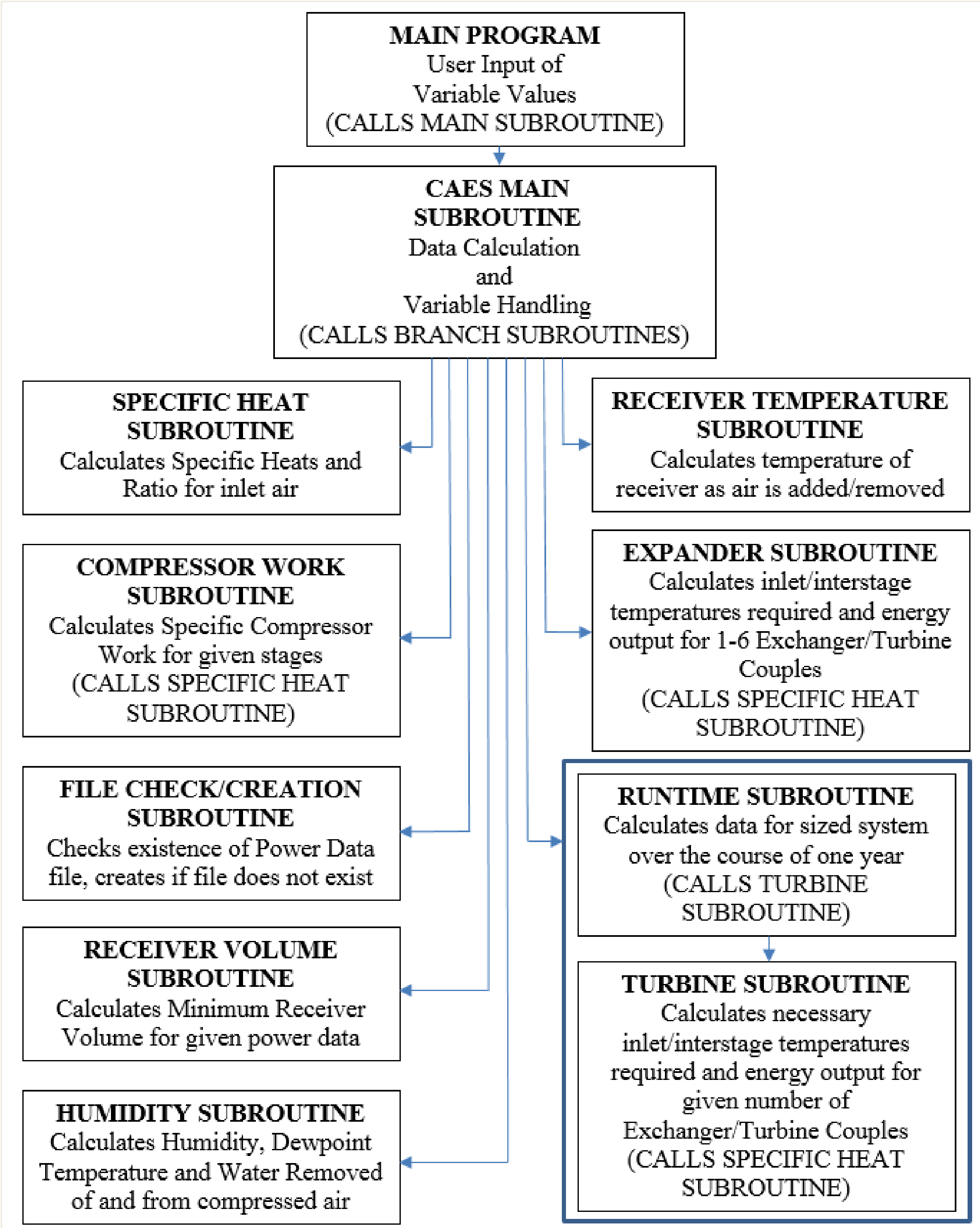


*Figure 22: Compressed Air Energy Storage Sizing and Runtime Program*

### 3.2.1 User Input (Main Program)

The user input page requires the user to assign values to the following variables:

- Number of Compression Stages *(stages, [unitless])*
- Ambient Air Pressure at the compressor inlet *($P_{in}$, [kPa])*
- Final Air Pressure at the compressor outlet *($P_{out}$, [kPa])*
- Working Pressure at the inlet of the expansion system *($P_{low}$, [kPa])*
- Compressor Inlet Air Temperature for all stages *($T_{in}$, [K])*
- Expected Steady State Temperature of the discharged receiver *($T_{tank}$, [K])*
- Isentropic Compressor Efficiency *($eta_{percent}$, [unitless, percentage])*
- Excess Data File Name *(fname, [character])*
- Number of Data Points (hours) in file *(n, [hours])*
- Isentropic Turbine Efficiency *($eta\_{tpercent}$, [unitless, percentage])*
- Wet Bulb Air Temperature at the compressor inlet *($T_{wb}$, [K])*

These variables are assigned values before the program is compiled. The original configuration of the Compressed Air Energy Storage Model did not incorporate this data input module. The user input and variable value assignment was handled entirely by the CAES MAIN program, which is now serving as the main subroutine. The drawback of the current configuration is that the user must compile each time a control variable is changed. The benefit is that this configuration is faster, practically, for multiple runs over multiple control variable changes, such as Working Pressure. Another benefit of this configuration is that iteration of control variables may written into the code of this main input program to quickly test multiple configurations. These variables, once assigned, are fed into the CAES MAIN subroutine when called.

### 3.2.2 CAES Main (Main Subroutine)

The main subroutine handles all of the other branch subroutines, except for the TURBINE SYSTEM subroutine, and performs certain calculations and output operations. Not all output to screen or to file is handled by this module; some occurs within the individual branch subroutines.

First, the ideal gas properties of dry air are read in to variable arrays from a data file (WARK DATA). These temperature-dependent properties, for various values of air temperature *($T_{tab}$, [K])* are:

- Specific Enthalpy *($h_{tab}$, [kJ/kg])*
- Relative Pressure *($P_{rtab}$, [unitless])*
- Specific Internal Energy *($u_{tab}$, [kJ/kg])*
- Relative Specific Volume *($v_{rtab}$, [unitless])*
- Reference Entropy *($s^o_{tab}$, [kJ/kg-K])*

These properties are read in for a non-continuous *($T_{tab}$)* temperature range of 200-2250 K. The universal gas constant *($R_u$, [kJ/kmol-K])* is assigned the value 8.3144627 kJ/kmol-K and the gas constant for air is calculated to be 0.287002 kJ/kg-K using a molar mass for dry air of 28.970 kg/kmol. The subroutine then writes to screen:
"This program will calculate the isentropic compression work required for staged air compression, the minimum storage receiver volume required given a data set of excess power values, and the reheat necessary to increase the enthalpy of the air in order to have ambient turbine outlet conditions. The energy output with the given reheat is calculated to determine the roundtrip cycle efficiency." (Wording depends on the turbine outlet conditions)

Next the subroutine writes to the screen all of the variable values assigned in the USER INPUT module, calculates and writes to screen the overall compression ratio *($r_c$, [unitless]),* then calls the SPECIFIC HEAT subroutine. This module calculates and exports the constant volume specific heat for air *($c_v$, [kJ/kg-K])*, the constant pressure specific heat for air *($c_p$, [kJ/kg-K])*, and their ratio *($k$, [unitless])* all at the inlet temperature of the compressor. The CAES MAIN writes these three variable and their values to the screen.

The next module to be called is the COMPRESSOR WORK subroutine, which calculates the ideal specific compressor work for staged and intercooled isentropic air compression. The variables that are exported back to the CAES MAIN module are the compression ratio per stage *($r_p$, [unitless])*, the compressor outlet temperature for each compression stage *($T_{out}$, [K])* and the specific compressor work *($w_{in}$, [kJ/kg])* required to compress the air. The outlet temperature and specific compressor work are calculated with the isentropic compressor efficiency.

The CAES MAIN module then calculates the specific enthalpies at the compressor inlet and outlet temperatures *($h_{comp\_hot}$, [kJ/kg]),( $h_{comp\_cold}$, [kJ/kg])*. The process for calculating these values is iterative and interpolative using the ideal gas properties of dry air table data. First, the table temperatures bounding the inlet or outlet temperature are located in the array and an interpolation quotient is generated for each. The specific enthalpy values corresponding boundary temperatures are used in conjunction with these interpolation quotients to calculate the specific enthalpy values. The values output from the COMPRESSOR WORK module are written to the screen as are the amount of specific enthalpy removed as heat in each compressor stage and the total specific enthalpy removed as heat during the entire process.

The existence of an excess power data file, which provides hourly excess power/energy data for a specific PEAK EXCESS time period, is checked. If the file, which was specified under the character variable *fname*, does exist, the program progresses to the next subroutine; if it does not exist, the EXCESS ENERGY FILE CHECK subroutine is called. This subroutine allows the user to input *n* integer number of hours and a corresponding hourly excess power level.

With the excess power data, CAES MAIN calls the RECEIVER VOLUME subroutine. This subroutine calculates the minimum allowable tank volume *($V_m$, [$m^3$])* given a constant tank temperature throughout the tank filling process. This volume is only a base estimate, and must be reconsidered given a greater amount of data. The subroutine also calculates a mass differential *( $\Delta m$ , [kg])* based on the "discharged" and "full" states as well as the excess energy available for compression, for which it calculates a sum *($W_{dot\_sum}$, [kWh])*. Finally, the maximum compressor (array) power required *(Comp_powmax, [kW])* as well as the maximum mass flow rate *($m_{dotmax}$, [kg/s])* for the period are calculated and exported to the CAES MAIN module.

The suggested receiver volume calculated in the RECEIVER VOLUME subroutine is written to the screen, and the user is asked to input the desired volume if different. This volume is written to the screen, and the HUMIDITY PROPERTIES subroutine is called. This module calculates and exports the minimum temperature *($T_{min}$, [K])* the compressed air can be cooled to before condensation occurs, as well as the amount of water *($W_{rem}$, [unitless])* per kg of air the compression process removes given the inlet, interstage, and outlet conditions of the compressor.

The next module to be called is the RECEIVER TEMPERATURE subroutine, which calculates and exports the base mass amount *($m_{discharged}$, [kg])* for the receiver at discharged and steady conditions as well as the receiver temperature *($T_{fill}$, [K])* after a period of filling. This calculation assumes no mass exits the receiver during the fill period.

The final module that is called for the sizing section of the program is the EXPANDER/HEAT EXCHANGER subroutine. This subroutine calculates but does not export to CAES MAIN the inlet *($T_{hxpd}$, [K])* and outlet *($T_{hout}$, [K])* temperatures based on the number of expansion stage / reheat exchanger couples, as well as the amount of energy *($W_{hot}$, [kWh])* that will be produced if the expansion of the stored mass occurs uninterrupted. These calculations include the isentropic turbine efficiency, and the calculations are iterated over six expansion stages (couples). If the program is used merely for sizing, it ends here; if it is being used for the run model as well, the RUNTIME subroutine is called.

The RUNTIME subroutine also does not export any calculated variables to CAES MAIN, but reads in per minute excess power data from file and runs the sized system through fill and empty cycles. It uses a variation of the EXPANDER/HEAT EXCHANGER subroutine, the TURBINE SYSTEM subroutine, to calculate the power output *($kW_{out}$, [kW])* of the turbine powertrain, the yearly energy produced by the array *(kJSUM, [kJ])*, the turbine inlet temperature *($T_{hxpd}$, [K])*, and the minimum amount of reheat *(kJREHEAT, [kJ])* needed to heat the air up to the specified inlet temperature. RUNTIME also calculates the temperature *($T_{rec}$, [K])*, pressure *($P_{rec}$, [kPa])*, mass in *($kgs_{in}$, [kg])*, mass out *($kgs_{out}$, [kg])*, total mass *($m_{tank}$, [kg])* and specific internal energy of the receiver system *($u_{tank}$, [kJ/kg])*. These variables are output to screen and data files. The remaining sections explore the details of each of the branch subroutines called by the CAES MAIN subroutine.

### 3.2.3 Specific Heat (Branch Subroutine)

The SPECIFIC HEAT subroutine uses the method of specific heat determination for air outlined in the NIST Thermodynamic Properties of Air[33]. The method uses eleven numerical coefficients, three derived coefficients, and five polynomial equations to calculate the constant pressure specific heat *($c_p$)*. These coefficients and equations are listed in Appendix A, and the resulting equation is:

$$c_p = R_{gas} \times \sum_{i=1}^{5} Eq_i \qquad (35)$$

This method is solely dependent on the air temperature, and is only valid for dry air. As discussed in the previous chapter, however, the effects of air are expected to be negligible. In order to calculate a mass-based specific heat, instead of a molar one, the gas constant for air

$(R_{gas})$ is used in place of the universal gas constant $(R_u)$. In order to calculate the constant volume specific heat $(c_v)$, the gas constant for air is subtracted:

$$c_v = c_p - R_{gas} \tag{36}$$

To calculate the ratio of specific heats $(k)$, the constant pressure specific heat is divided by the constant volume specific heat:

$$k = \frac{c_p}{c_v} \tag{37}$$

All three of these values are exported back to the CAES MAIN subroutine were they are written to the screen.

### 3.2.4 Compressor Work (Branch Subroutine)

The COMPRESSOR WORK subroutine begins by calculating the stage ratio $(r_p)$ for the compression process:

$$r_p = r_c^{1/s_c} \tag{38}$$

Where $r_c$ is the overall compression ratio and $s_c$ is the number of compression stages. Then the amount of pressure rise per stage $(P_{int}, [kPa])$ is calculated:

$$P_{\text{int}} = \left(P_{in}^{\,s_c - 1} \times P_{out}\right)^{1/s_c} \tag{39}$$

Where $P_{in}$ is the initial air pressure at the inlet and $P_{out}$ is the final air pressure at the outlet. The outlet temperature $(T_{out}, [K])$, the specific heat ratio for the outlet temperature $(k_{out})$ and an average specific heat ratio $(k_{avg})$ are all initialized prior to iterating the specific compressor work $(w_i, [kJ/kg])$ per stage and the outlet temperature values. These calculations take place until the difference between average specific heat ratio values between iterations is less than or equal to 0.0000001. $w_i$ is calculated as:

$$w_i = \frac{\frac{k_{avg} \times R_{gas} \times T_{in}}{k_{avg} - 1} \times \left( r_p^{\,k_{avg}-1 / k_{avg}} - 1 \right)}{\eta_c} \tag{40}$$

Where $\eta_c$ is the isentropic compressor efficiency. The average specific heat ratio is calculated as:

$$k_{avg} = \frac{k_{in} + k_{out}}{2} \tag{41}$$

The outlet temperature, for each stage given the interstage temperature is intercooled to the inlet temperature is:

$$T_{out} = T_{in} + \left( w_i \times \frac{k_{avg} - 1}{k_{avg} \times R_{gas}} \right) \tag{42}$$

After these two calculations are made, the SPECIFIC HEAT subroutine is called for the iterated outlet temperature, the new average specific heat ratio is calculated, and the process iterates with the specific heat difference check. Each iteration is output to the screen, and when the process is complete, the stage specific compressor work is multiplied by the number of stages to calculate the total compressor work *($w_{in}$, [kJ/kg]):*

$$w_{in} = w_i \times s_c \tag{43}$$

These values are then output to the screen.

### 3.2.5 Excess Energy File Check (Branch Subroutine)

If called, this subroutine is tasked with creating a data file that will serve as the source of information for the peak period of excess power production. The subroutine asks the user to input the number of hourly data points to be input, and allocates the array *($W_{dotxs}$, [kW])* depending on this number. The user is then asked to input the data hour by hour as the values are echoed back to the screen.

### 3.2.6 Receiver Volume (Branch Subroutine)

The RECEIVER VOLUME subroutine begins by initializing the receiver volume, $V_{m,}$ to 1 m$^3$. It calculates the air densities of the high pressure air *(* $\rho_{high}$ *, kg/m$^3$)* and the low pressure air *(* $\rho_{low}$ *, kg/m$^3$)* at the steady state temperature $T_{tank}$:

$$\rho_{high/low} = \frac{P_{high/low}}{R_{gas} \times T_{\mathrm{tank}}} \tag{44}$$

Then the subroutine allocates arrays for the excess power data, $W_{dotxs}$, the compressor mass flow rate *($m_{dot}$, [kg/s])*, the hourly compressed mass *($m_{comp}$, [kg])*, the hourly compressor power *($W_{dot}$, [kW])* and the hourly fraction of excess power *(foe, [unitless])*. Then the excess power data are read into $W_{dotxs}$ from the excess power data file. All of the excess power values are summed *($W_{dot\_sum}$, [kWh])*, and the hourly fraction of excess values are calculated by dividing the hourly $W_{dotxs}$ by the $W_{dot\_sum}$ value. All of the hourly compressor power values are initialized to zero and each value is iteratively calculated until the FIRST hourly compressor value is greater than the FIRST excess power value. The iterations stop and the previous iteration results *(i-1)* are used as the final values. The iteration process begins with finding the masses at the low and high pressures respectively:

$$m_{low} = V_m \times \rho_{low} \tag{45}$$

$$m_{high} = V_m \times \rho_{high} \tag{46}$$

The difference between the two values gives the mass differential *(* $\Delta m$ *, [kg])*. The hourly compressed mass is the fraction of excess multiplied by the mass difference:

$$m_{comp} = foe \times \Delta m \tag{47}$$

and the mass flow rate is calculated as the hourly compressed mass divided by 3600 seconds/hour:

$$m_{dot} = \frac{m_{comp}}{3600\, s/h} \tag{48}$$

The power required to compress the mass is the specific compressor work, $w_{in}$ [kJ/kg] multiplied by the mass flow rate $m_{dot}$ [kg/s]:

$$W_{dot} = m_{dot} \times w_{in} \tag{49}$$

After one iteration for all of the $W_{dot}$ values, the receiver volume is increased by 1 $m^3$ and the iteration is restarted. When done, the calculated values are output to screen and to a file named *resultsXXXX.dat,* where *"XXXX"* is *"300K," "MIN,"* or *"MAX"* depending on the outlet conditions specified for the turbine array. The hourly mass flow rate as well as the mass compressed per hour are written to a file named *massflow.dat* to be used in the RECEIVER TEMPERATURE subroutine.

Finally, it should be stated that constant power values are assumed for each hour. This is a rather bold assumption, but for sizing purposes it is functional. It also makes calculation of hourly energy simple, as a constant power [kW] for an hour is the same magnitude in units of energy [kWh].

### 3.2.7 Humidity Properties (Branch Subroutine)

The HUMIDITY PROPERTIES subroutine begins by allocating data arrays for humidity ratio ($W_{rat}$, [$kg_w/kg_{da}$]) data and their corresponding temperatures ($T_{rat}$, [K]). It then reads in and stores a partial set (253-313 K) of these data from a data file. This file is derived from the table of Thermodynamic Properties of Moist Air at Standard Atmospheric Pressure[32]. The temperature-dependent humidity ratio corresponds to the compressor inlet air wet bulb temperature ($T_{wb}$) assigned earlier, and characterizes the amount of water vapor mass that exists per dry air mass in saturated air. The inlet air in the analyses performed is assumed to be at a higher dry bulb temperature ($T_{in}$), so the inlet air is not fully saturated. The model can, however, calculate the humidity of the compressed air and water removed given different variable values.

Next, a partial set (273-393 K) of saturated water absolute (vapor) pressure *($P_w$, [kPa])* data and corresponding temperatures *($T_{vp}$, [K])* read in from a data file and stored. This file is derived from the table of Thermodynamic Properties of Water[32]. The absolute pressure of saturated water is temperature dependent and indicates the partial pressure of the water in the air at a given temperature.

The humidity ratio *($W_{s\_in}$, [$kg_w/kg_{da}$])* of the inlet air is assigned based on the dew point temperature of the inlet air, as is the inlet vapor pressure *($P_{w\_in}$, [kPa])*. The vapor pressure *($P_{w\_cool}$, [kPa])* of the intercooled and compressed air is assigned based on the inlet/interstage/aftercooled temperature *($T_{in}$)*. With these and the overall compression ratio, the vapor pressure *($P_{w\_comp}$, [kPa])* of the final aftercooled air is calculated:

$$P_{w\_comp} = r_c \times P_{w\_cool} \tag{50}$$

With $P_{w\_cool}$, the humidity ratio *($W_{s\_comp}$, [$kg_w/k_{da}$])* of the compressed and aftercooled air can be calculated:

$$W_{s\_comp} = 0.62198 \times \frac{P_{w\_cool}}{P_{out} - P_{w\_cool}} \tag{51}$$

The minimum temperature *($T_{min}$, [K])* to which the air can be cooled before condensation occurs is interpolated iteratively over the humidity ratio data table values. The amount of water removed per kg of air is calculated:

$$W_{rem} = W_{s\_in} - W_{s\_comp} \tag{52}$$

All relevant variable values are written to the screen.

### 3.2.8 Receiver Temperature (Branch Subroutine)

The RECEIVER TEMPERATURE subroutine first reads in and records the ideal gas properties of air into allocated arrays, as described in the CAES MAIN subroutine description. This is performed due to the data corruption issues encountered when directly passing arrays between subroutines. The base mass in the receiver *($m_{discharged}$, [kg])* at the steady state temperature *($T_{tank}$)* and turbine inlet (working) pressure *($P_{low}$)* is calculated with the mass based ideal gas equation:

$$m_{\text{discharged}} = \frac{V_{\text{tank}} \times P_{low}}{T_{\text{tank}} \times R_{gas}} \tag{53}$$

This mass can fluctuate during normal operation depending on the momentary temperature and pressure conditions of the air in the receiver. These properties are NOT independent, and the mass entering and exiting the receiver are assumed to be limited with pressure controls *($P_{low} \leq P_{rec} \leq P_{out}$)*. If the total mass in the receiver falls below the $m_{discharged}$ amount but the temperature of the receiver is high enough to keep the receiver pressure above the turbine inlet pressure, mass will be allowed to exit the receiver. The calculated $m_{discharged}$ amount is only used as a starting point to provide a basis for further calculations, particularly the transient receiver temperature.

Arrays for the hour, hourly mass flow rate *($m_{dot}$)* and hourly compressed mass *($m_{comp\_h}$)* are allocated and assigned with values read in from the file *massflow.dat* created in the RECEIVER VOLUME subroutine. Each of the hourly compressed mass values are divided into 3600 per second compressed mass values, and assigned to the allocated array *($m_{comp\_s}$)*. These per second values, as well as their corresponding second counts, are written to a data file named *massflow_sec.dat.*

Arrays for the control volume specific internal energy *($u_{cv}$, [kJ/kg])*, control volume mass *($m_{cv}$, [kg])*, and control volume temperature *($T_{cv}$, [K])* are allocated and initialized. These values will be used for a steady-state steady flow first-law analysis of a tank-filling[7]. The equation, which is iterated over the entire time range, is:

$$u_{cv2} = \frac{\left(m_{cv1} \times u_{cv1}\right) + \left(m_{comp\_s} \times h_{fill}\right)}{m_{cv2}} \tag{54}$$

The control volume mass $(m_{cv2})$ and internal energy $(u_{cv2})$ of each iteration become the old values $(m_{cv1}, u_{cv1})$ for each subsequent calculation, while the inlet stream enthalpy $(h_{fill}, [kJ/kg])$ is constant and the mass added to the system $(m_{comp\_s})$ is variable. The initial control volume mass is assumed to be $m_{discharged}$, the initial temperature of the receiver is assumed to be the steady state temperature $T_{tank}$ and the initial specific internal energy is assumed to be the corresponding ideal-gas dry air value for the steady state temperature, which is interpolated from the table values. The inlet stream enthalpy is interpolated from table values using the aftercooled compressor outlet air temperature $(T_{in})$. Each per second temperature is interpolated from table values using the per second specific internal energy values. The control volume mass, temperature, and specific internal energy values for each second are written to the screen, as are the final receiver temperature, total internal energy, and final mass in the receiver.

This subroutine assumes two things which may cause the temperature values to be artificially high. First, that the pressure increase over the process is exactly linear, from the discharged "base" pressure $(P_{low})$ to the cutoff "fill" pressure $(P_{out})$, which equals the compressor outlet stream pressure. This is because the mass differential is based on the second assumption, that the discharged receiver and the charged receiver have a steady temperature $(T_{tank})$. This assumption was made in the RECEIVER VOLUME subroutine, and was made as a conservative measure to minimize the receiver volume. Dense cooler air will take up less volume at the same pressure and mass than hotter air. These values for receiver mass, temperature, and pressure could be further iterated among the subroutines until more refined values were established. This was not done here, and the receiver temperature results should be considered an estimate. These values simply show a trend of temperature increase as a result of mass addition.

### 3.2.9 Expander/Heat Exchanger (Branch Subroutine)

Two major versions of the EXPANDER/HEAT EXCHANGER subroutine are used. Both calculate the pressure drop per turbine stage, the inlet and outlet temperatures, the inlet and outlet enthalpies, and the energy produced for the process. The main difference between the two versions is that the first version specifies and holds the turbine outlet temperature, and the second version sets and hold the required output energy per turbine stage. The two versions are described below.

#### 3.2.9a Expander/Heat Exchanger [$T_{h\_out}$ Defined] (Branch Subroutine)

This version of the subroutine is used when the user wants to specify an outlet temperature *($T_{h\_out}$)* for the turbine gases. This outlet temperature is held for all stages equally, while the inlet temperature *($T_{hxpd}$)*, the work energy output *($W_{hot}$)* and the inlet and outlet enthalpies are all calculated.

First, the results file is reopened to be appended, then the ideal gas properties of air are read into allocated arrays, as described in the CAES MAIN subroutine description. All of the operations from this point are repeated iteratively over six stages of turbine/heat exchanger couples, and begins by calculating the turbine stage expansion ratio *($t_{rp}$, [unitless])*:

$$tr_p = \left(\frac{P_{in}}{P_{low}}\right)^{\frac{1}{s_T}} \tag{55}$$

Where $P_{in}$ is the same pressure as at the inlet of the compressor (the ambient pressure) and $s_T$ is the number of turbine stages. An array for intermediate pressures *($P_{int}$, [kPa])*, between turbine stages, is allocated based on the number of expander couples and these pressures are calculated iteratively:

$$P_{\text{int}} = P_{\text{max}} \times tr_p \tag{56}$$

Where $P_{max}$ is the inlet pressure of the turbine stage. The time for expansion *($t_{xpd}$, [s])*, which is based on the mass differential ( $\Delta m$ ) and the mass flow rate *($m_{dot\_xpd}$, [kg/s])* through the expansion system is calculated:

$$t_{xpd} = \frac{\Delta m}{m_{dot\_xpd}} \tag{57}$$

It must be noted that the mass flow rate through the turbine array is automatically taken as the maximum mass flow rate through the compression system. This is an arbitrary decision and can be changed in the code, possibly to a user input parameter. This expansion period is converted to hours *(TIME, [h])* for later calculations:

$$TIME = \frac{t_{xpd}}{3600\, s/h} \tag{58}$$

The turbine outlet temperature *($T_{h\_out}$)*, which is assumed for all turbine stages, is assigned, and the corresponding relative pressure value *($P_{r\_out}$)* is interpolated from table data. Since the stage ratio is known, the relative pressure for the inlet (given a perfectly isentropic process) is calculated:

$$P_{r\_in} = \frac{P_{r\_out}}{tr_p} \tag{59}$$

With this, the isentropic inlet temperature *($T_{sin}$, [K])* is interpolated from the air table data, and the SPECIFIC HEAT subroutine is called to find the specific heat and ratio values for the isentropic turbine inlet and outlet temperatures. An average value is determined, with equation (41), and this value is used to find the isentropic specific stage work out (here, $kJ_{psreq}$, *[kJ/kg]* is used for the stage work out variable, which is NOT the way it is used in the second version).

$$kJ_{psreq} = \frac{k_{avg} \times R_{gas}}{k_{avg} - 1} \times \left( tr_p^{\frac{k_{avg}-1}{k_{avg}}} - 1 \right) \times T_{\sin} \tag{60}$$

The turbine inlet temperature including the isentropic turbine efficiency $(\eta_T)$ is then calculated:

$$T_{hxpd} = T_{h\_out} + \frac{T_{\sin} - T_{h\_out}}{\eta_T} \tag{61}$$

The specific enthalpies for the inlet *($h_{turb\_in}$, [kJ/kg])* and the outlet *($h_{turb\_out}$, [kJ/kg])* are interpolated using air table data and their corresponding temperatures, and the difference between the two, along with the mass flow rate, the expansion time in hours, and the stage number, is used to find the work energy output *($W_{hot}$, [kWh]):*

$$W_{hot} = m_{dot\_xpd} \times \left(h_{turb\_in} - h_{turb\_out}\right) \times s_T \times TIME \tag{62}$$

All of the relevant variable values are written to the results file as well as to the screen, and the process is iterated for the next number of stages. If the sizing model is being run, this is the final branch subroutine to be called.

### 3.2.9b Expander/Heat Exchanger [$kJ_{psreq}$ Defined] (Branch Subroutine)

This version of the subroutine is used when the user wants to specify that all of the input energy *($W_{dot\_sum}$)* is recovered, calculating the necessary turbine inlet and outlet temperatures. The subroutine begins and runs exactly as the other version (with the same results and stage iterations) until the module calculates the variable $kJ_{psreq}$. In the alternate version, this variable is used as the specific isentropic (maximum) turbine stage work [kJ/kg]. Here it is used to calculate the work energy per stage required to recuperate the energy input for compression. The calculation is:

$$kJ_{psreq} = \frac{W_{dot\_sum}}{s_T \times TIME \times m_{dot\_xpd}} \tag{63}$$

The subroutine initializes an average value for the ratio of specific heats to perform iterative calculations to find the isentropic turbine inlet temperature and the turbine outlet temperature. These calculations are iterated until the difference between the $k_{avg}$ values of the current iteration and the previous iteration is less than or equal to 0.000001, and begin with:

$$T_{\sin} = \frac{-kJ_{psreq}}{\left(\frac{k_{avg} \times R_{gas}}{k_{avg} - 1}\right) \times \left(tr_p^{\frac{k_{avg}-1}{k_{avg}}} - 1\right)} \tag{64}$$

The SPECIFIC HEAT subroutine is called to find the $k_{in}$ value corresponding to $T_{sin}$, then the corresponding relative pressure *($P_{r\_in}$)* term is interpolated from the air table data. With this, and equation (59), the relative pressure term for the outlet is calculated which is used to interpolate the turbine outlet temperature *($T_{h\_out.}$)*. The SPECIFIC HEAT subroutine is again called for the outlet temperature to calculate the corresponding $k_{out}$ value. Using equation (41) to find the next $k_{avg}$ value, the subroutine checks the difference between the old and new values. Once the exit condition is satisfied, and acceptable values for $T_{sin}$ and $T_{h\_out}$ are calculated, the turbine inlet temperature that incorporates the isentropic turbine efficiency can be calculated with equation (61).

Next, the specific enthalpy values that correspond to $T_{hxpd}$ and $T_{h\_out}$ are interpolated, just as they are in the alternate version, with air table data. The total energy output *($W_{hot}$)* is calculated with equation (62), and should be within a few percentage points of the input energy. All of the relevant variable values are written to the results file as well as to the screen, and the process is iterated for the next number of stages. If the sizing model is being run, this is the final branch subroutine to be called.

### 3.2.10 Runtime (Branch Subroutine)

This subroutine takes the sizing information and simulates a running system over a period of time (one year). The module begins by reading the ideal gas properties of air into allocated arrays, as described in the CAES MAIN subroutine description. The user is asked to input the desired number of turbine/heat exchanger stages, and this value is read in.

A data file (currently named *dayta_year.dat*) that contains hourly averaged power data is read into an allocated array *(g)* of size 8760. Each hourly data point is then divided into per-minute averaged power data and allocated into an array *($kJ_{in}$, [kJ/min])*. Arrays for per-minute properties are also allocated for mass into the system *($kgs_{in}$, [kg])*, mass out of the system *($kgs_{out}$, [kg])*, receiver temperature *($T_{rec}$, [K])*, receiver pressure *($P_{rec}$, [kPa])*, total receiver mass *($m_{tank}$, [kg])*, the specific internal energy of the air in the receiver *($u_{tank}$, [kJ/kg])*, the specific enthalpy of the unheated heat exchanger inlet stream (receiver outlet) *($h_{empty}$, [kJ/kg])*, the total per-minute energy required for reheating the turbine inlet air *($kJ_{REHEAT}$, [kJ])*, the per-minute average power required to supply this reheat *($kW_{REHEAT}$, [kW])* and the per-minute energy produced by the expansion system *($kJ_{SUM}$, [kJ])*.

At this point, the TURBINE SYSTEM subroutine is called. This subroutine is EXACTLY LIKE the corresponding EXPANDER/HEAT EXCHANGER subroutine, except it does not iterate over six stages and it outputs a turbine system power rating *($kW_{out}$, [kW])*. The subroutine also return parameters vital for simulating the running system, such as the turbine inlet temperature *($T_{hxpd}$)*, the total change in specific enthalpy across the array *($\Delta h_{hot}$)*, and the specific enthalpy at the turbine inlet *($h_{turb\_in}$)*.

If the value of the per-minute input power is positive, indicating excess energy, the POTENTIAL per-minute input mass is calculated as:

$$kgs_{in} = \frac{kJ_{in}}{w_{in}} \tag{65}$$

Where $w_{in}$ is the specific compressor input work necessary for compression. The output mass for this minute is automatically assumed to be zero, as the system is not expected to store and produce energy simultaneously.

If the value of the per-minute input power is not positive, indicating no excess energy, the POTENTIAL per-minute output mass is calculated as:

$$kgs_{out} = m_{dot\_xpd} \times 60s \tag{66}$$

This is merely the mass flow rate over one minute's time. The input mass is assumed to be zero, as no excess power is available air compression. The potential yearly energy that can be stored *($kJ_{inPOT}$)* is calculated as a sum of all the per-minute excess energy values:

$$kJ_{inSUM} = \sum_{i=1}^{525600} kJ_{in,i} \tag{67}$$

This can also be accomplished by summing the original 8760 data values from the data file. The initial receiver temperature and pressure are initialized to $T_{tank}$ and $P_{low}$, respectively. The initial receiver specific internal energy corresponding to $T_{tank}$ and the specific enthalpy of the aftercooled compressor outlet stream *($h_{fill}$, [kJ/kg])* corresponding to the temperature *($T_{in}$)* are interpolated with air table data. The remaining per-minute receiver properties are then iteratively calculated over the entire range of minute values and assigned to their respective arrays.

If the receiver pressure drops to a point less than or equal to 1.001 times the turbine inlet pressure, mass outflow is stopped ($kgs_{out}$ = 0 kg) and the expansion process is halted. The pressure at the turbine inlet is not permitted to go below the design pressure. Otherwise, if the receiver pressure rises to or above the compressor outlet pressure *($P_{out}$)*, mass inflow is stopped ($kgs_{in}$ = 0 kg) and storage is halted. If there is excess energy, it will be wasted, as the compressor will not be able to physically push any more mass into the receiver if the receiver pressure is higher than the compressor outlet stream pressure. With these constraints, the per-minute masses in or out are summed to the previous minute's system mass:

$$m_{\text{tank,i}} = m_{\text{tank,i-1}} + kgs_{in,i} + kgs_{out,i} \tag{68}$$

With the calculated mass, the specific internal energy of the receiver is calculated:

$$u_{\text{tank,i}} = \frac{\left(m_{\text{tank,i-1}} \times u_{\text{tank,i-1}}\right) + \left(kgs_{in,i} \times h_{fill}\right) - \left(kgs_{out,i} \times h_{empty,i-1}\right)}{m_{\text{tank,i}}} \tag{69}$$

Using this calculated specific internal energy, the receiver temperature and exit stream enthalpy are interpolated using air table data. The ideal gas equation is used to find the receiver pressure:

$$P_{rec,i} = \frac{m_{\text{tank,i}} \times R_{gas} \times T_{rec,i}}{V_{\text{tank}}} \tag{70}$$

If the per–minute mass out is positive, the per-minute energy out is also positive. It is calculated as:

$$kJ_{SUM,i} = kW_{out} \times 60s \tag{71}$$

If the receiver temperature is greater than or equal to the calculated turbine inlet temperature, the energy needed for reheat is assumed to be zero. Otherwise, it is calculated as:

$$kJ_{REHEAT,i} = \left[\left(s_T - 1\right) \times \Delta h_{hot}\right] + \left[\left(h_{turb\_in} - h_{empty,i-1}\right) \times kgs_{out,i}\right] \tag{72}$$

This accounts for the remaining stages of the turbine array (first grouping in equation) separately from the first stage (second grouping), as they may require different amounts of reheat. The first stage requires that reheat is applied to a variable receiver outlet temperature, while the subsequent stages will all require the same amount of reheat, as the turbine should have the same outlet temperature regardless of stage.

Finally, the per-minute average power required for reheat is calculated:

$$kW_{REHEAT,i} = \frac{kJ_{REHEAT,i}}{60s} \tag{73}$$

The per-minute mass, temperature, specific internal energy and exit stream enthalpy values are written to an output file named *WHOLE_YEAR_XXXX.DAT* where the suffix *'XXXX'* is *'300K', 'MIN',* or *'MAX'* depending on the turbine outlet temperature. The per-minute power and energy values are written to an output file named *POWER_OUT_XXXX.DAT* where the suffix *'XXXX'* is *'300K', 'MIN',* or *'MAX'* depending on the turbine outlet temperature.

### 3.2.11 Turbine System (Branch Subroutine)

The TURBINE SYSTEM subroutine also has two versions, almost exactly like their counterpart EXPANDER/HEAT EXCHANGER subroutines. The two main differences between the subroutines is the TURBINE SYSTEM subroutine only considers one user-specified stage number, for which it calculates and exports the turbine inlet temperature ($T_{hxpd}$), the total change in specific enthalpy across the array ($\Delta h_{hot}$), and the specific enthalpy at the turbine inlet ($h_{turb\_in}$). This subroutine also calculates the power output of the turbine array at the specified mass flow rate ($kW_{out}$, *[kW]*):

$$kW_{out} = \frac{W_{hot}}{TIME} \tag{74}$$

Where TIME is the number of hours the rated array can run for, given the sizing input, and $W_{hot}$ is the total energy out [kWh].

# CHAPTER 4

## System Analysis and Results

The completed model framework can now be used to size and simulate a compressed air energy storage system. The purpose of this chapter is to demonstrate the different ways in which the model was used, and the observed system changes. The relationships determined from this study will be used towards the ultimate goal: optimization of the expansion system. The system was simulated under six general configurations, for which certain variables were changed and other variables kept the same:

1) Turbine/Heat Exchanger Couple Number (Stages) and Turbine Outlet Temperature
2) Turbine Inlet Pressure
3) Final Compressor (Storage) Pressure
4) Turbine Inlet and Storage Pressure (10 bar Gradient)
5) Turbine Mass Flow Rate
6) Compressor Inlet Dew Point Temperature

All system configurations have six variable values in common: number of compression stages *($Stages_{comp}$)*, compressor inlet temperature *($T_{comp,in}$ , [K])*, compressor intercooling/aftercooling temperature *($T_{intercool}$, [K])*, compressor inlet pressure *($P_{comp,in}$, [kPa])*, number of expansion stages for system sizing *($Stages_{turb,size}$)* and expansion system outlet pressure *($P_{turb,outlet}$ [kPa])*. These values are listed in Table 1; the compressor system variables that did not have common values throughout were the final compressor pressure (storage pressure) and the inlet dew point temperature. These values were changed in configurations 3, 4 and 6.

*Table 1: Air Compression Variable Values Used for All Simulations*

| $Stages_{comp}$ | $T_{comp,in}$ (K) | $T_{intercool}$ (K) | $P_{comp,in}$ (kPa) | $Stages_{turb,size}$ | $P_{turb,outlet}$ (kPa) |
|---|---|---|---|---|---|
| 3 | 300 | 300 | 101.325 | 1-6 | 101.325 |

These six variables were kept the same in order to allow the various configurations to have a common baseline or control. The compression-specific variables were assumed to only affect compression directly, and the expansion stage numbers are iterated over six stages in the sizing model only. The number of expansion stages for the simulation model is chosen as a separate variable. Both the isentropic compressor efficiency as well as the isentropic turbine efficiency were held at 80% for all sizing and simulation runs.

Three points that must be clarified are the differences between the fraction of input energy, the energy input return and the overall system efficiency. The fraction of input energy only measures the output energy as a fraction of the total energy input. This ratio does not consider any additional reheat energy, nor does it consider the heat removed from the compressed air. It is simply expressed as:

$$\text{Fraction of Input Energy} = \frac{\text{Energy Output from Expansion}}{\text{Energy Input for Compression}} \times 100\% \tag{75}$$

This is only a measure of the how the changes in the system variables change the energy output; this is certainly not a measure of system performance.

The input return, on the other hand, is the amount of energy output generated by the expansion system, with the reheat energy subtracted, compared to the amount of energy input to the compression system. It can be expressed as:

$$\text{Input Return} = \frac{\left(\text{Energy Output From Expansion - Reheat Energy}\right)}{\text{Energy Input for Compression}} \times 100\% \tag{76}$$

This expression does not treat the reheat energy as "just another energy input." It treats it as a fraction of the energy output that must be used to actually generate the energy. For example, if the system were to generate electricity, but instead of putting all of the electricity to the grid, it would have to send some back to the reheat system in order to reheat the air at each turbine inlet. The system is, in this case, chasing its tail: it is generating electricity to supply reheat in order to generate electricity. This sort of ratio can result in negative numbers. A system with a 100% return would be a completely reversible system without losses or secondary inputs. A system with a 0% return is a system that requires an equal amount of reheat for the energy it generates. A negative % return is a system that requires more energy in reheat than the energy it generates.

Further, this expression does not consider the rejected energy of compression; it is treated as a heat loss but it is still included as input energy. This may seem misleading, particularly if the rejected heat is stored and used to supply some of the reheat; then the accounting changes. This method of performance evaluation, however, is the most conservative for an energy storage system. It assumes maximum input and minimum return, and assumes no conservation measures are in place. The reason for this is that such measures are not included in the model yet.

The overall system efficiency is what most people are used to dealing with, regarding performance. Regardless of the way the energy is handled, the efficiency boils down to a simple ratio:

$$\text{Overall System Efficiency} = \frac{\text{Useful Energy Output}}{\text{Total Energy Input}} \times 100\% \tag{77}$$

The useful energy output is the energy generated by the expansion system. The rejected heat of compression is not treated as useful output here because it is not being used. It could be, however, if it were being used for some purpose such as reheat or preheat for the primary generator. The reheat energy, though a secondary input here, is treated as an input nonetheless. This way, a more accurate view of the system as a generation entity can be evaluated. It may be the case, as stated previously, that a facility will use some sort of fuel for electricity production. Such a system may not pass solely as an energy storage system. Its implementation may, however, increase the efficiency, productivity, or profitability of a traditional generation facility. Since this analysis is focused on straightaway energy storage and return, the input return method is used as the primary measure of a system’s predicted performance.

## 4.1 Excess Energy

The models outlined and explained in the previous chapter can only be run given the requisite excess energy input data. As shown in Figure 23, excess power is generated and used to run the air compression system. If the minimum power threshold is not met by the excess energy, the compressor system will not operate, and the energy will not be stored. How this sub-threshold energy is managed was not considered for this model; it will have to be considered, however, for a real system. The data in Figure 23 serve as the data used in the sizing model and represent the peak day of excess power generation. Each data point represents, for the purposes of this model, a steady power value for the corresponding hour. In this way, power data (*kW* for the hour) and energy (*kWh*) are interchangeable. In practice, however, it is unlikely that excess power will be generated at such a consistent rate.

The excess energy is generated primarily in the evenings through the morning, when energy usage is typically low. When the system in not producing excess energy, it is assumed that the generation system needs to produce more energy to supplement system loads. Therefore, in the simulation model, whenever the CAES system is not storing energy, it is either generating energy or it is off (discharged). The expansion/generation system operates at a constant rate. Whether or not a true system would operate in this manner (immediately and steadily) is completely up to the system designer or manager. It is possible that a variable, load-following expansion/generation system, which would more closely mirror the compression system, would better suit an actual system.

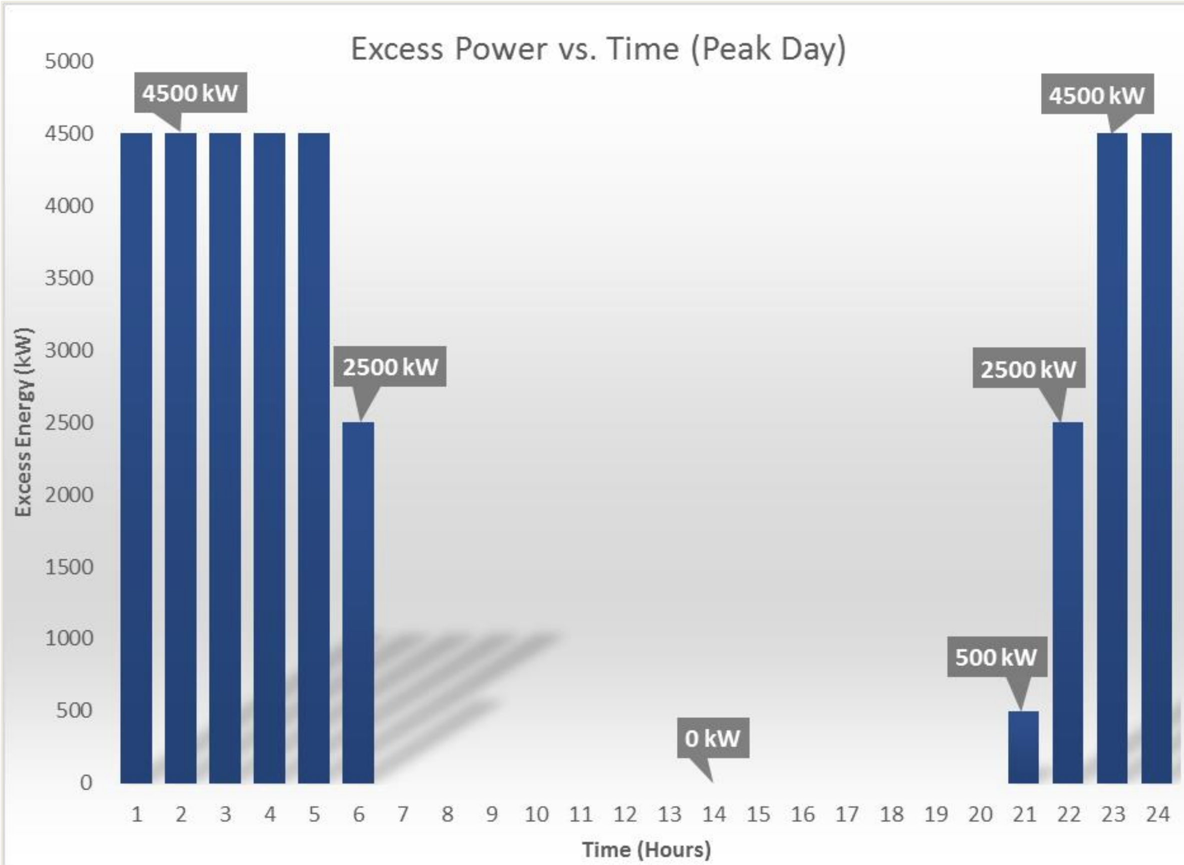


*Figure 23: Excess Power vs. Time (Peak Day)*

The next set of data, used with the simulation model, is the yearly excess power data. This data, shown in Figure 24, is also in discrete per-hour terms. The power levels are functionalized to change daily over the course of the year (8670 hours). For the first half of the year (day 0-182.5), the values steadily decrease in a linear fashion. In the latter half of the year (day 182.5-365), the values steadily increase linearly. Therefore, the excess power data has a very shallow "V" shape when viewed over the course of a year, and the data are exactly symmetric. The first and last days have the same data, the second and second-to-last days have the same data, and so forth.

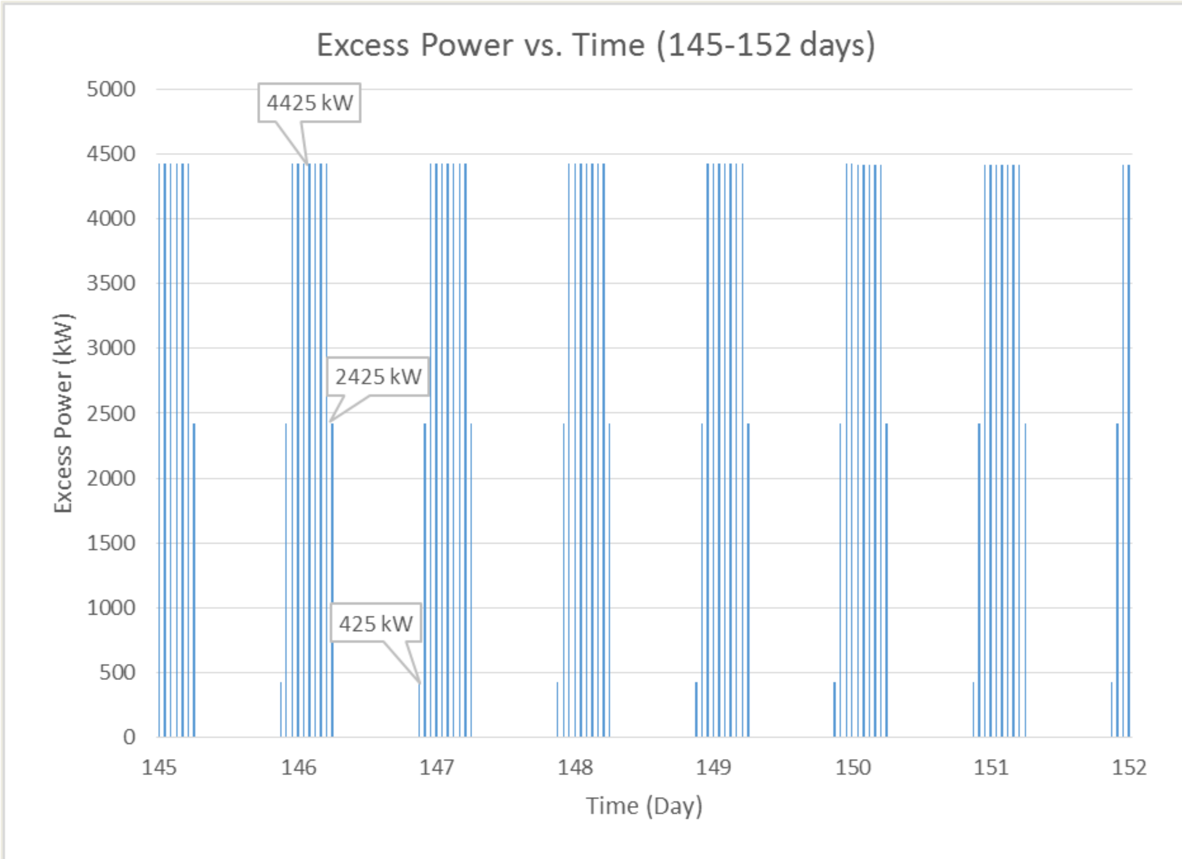


*Figure 24: Yearly Excess Power vs. Time (Day 145-152)*

It is unlikely that a true system or even real data will behave in this highly predictable manner. This model must be run with real system data before any validation from real world systems should take place. This is to safeguard the decision making process from possible errors or incorrect results caused by programming bugs or data-induced behavioral trends.

### 4.2 Receiver Conditions

Inherent in the operation of the system are charge-discharge cycles. Mass is added to the receiver at a certain mass flow rate and pressure, then removed at another mass flow rate and pressure. These occur in separate, discrete charge and discharge processes. At any point, these processes will affect the temperature and pressure of the air in the receiver. This can be seen in Figure 25, mass rates into the compressor are variable, dependent on power input while the mass rate out is static. The data pertaining to the receiver conditions in this section were collected from the simulation of a system with a 300 K turbine outlet temperature over three expansion stages.

For the first week of operation (day 0-7), this difference is not easily seen. This is largely a result of the method by which the excess power data were fabricated, and of the way the expansion mass flow rate is sized for the simulation model. The data for the peak day are exactly the same data as the first and last days in the yearly schedule of excess power. The data are at a minimum on the middle day of the year and at maxima on the first and last days. Also, the mass flow rate for the expansion system is based on the maximum mass flow rate of the compression system on the peak sizing day. Therefore, the extremely slight negative slope of the mass flow rate into the compressor, for days 1-7, makes differences between the flow rates in and out almost impossible to see. Also of note, the overall total mass has a slightly increasing trend despite charge-discharge cycles, signifying that the system has not yet reached steady state.

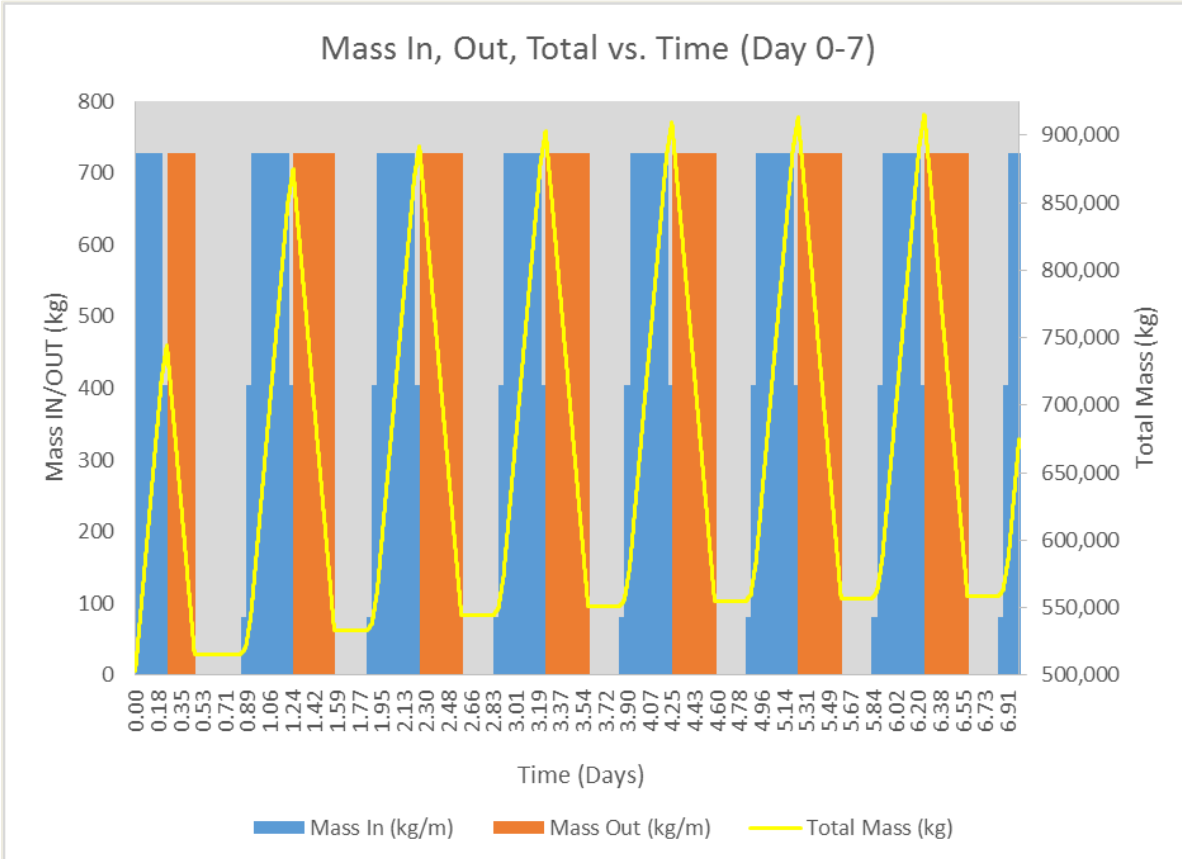


*Figure 25: Receiver Mass In/Out & Total vs. Time (Day 0-7)*

Data in Figure 26 below are taken from the middle week of the year, from day 179-186. The discrepancy between the maximum compressor mass flow rate and the expansion system mass flow rate is much clearer, and this is the point at which they are most different. The total receiver mass appears steady and centers at approximately 725,000 *kg*, which is not significantly different from the 7$^{th}$ day in Figure 25.

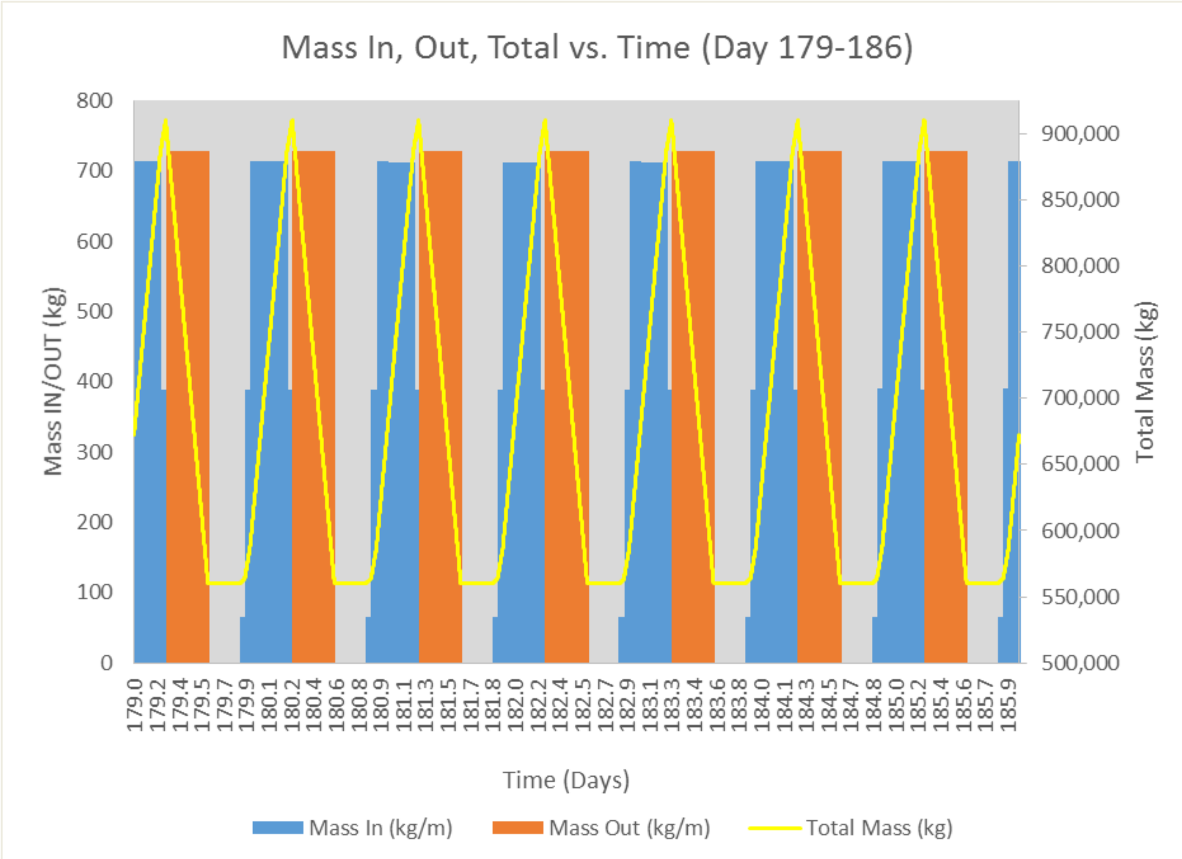


*Figure 26: Receiver Mass In/Out & Total vs. Time (Day 179-186)*

The temperature and pressure correspond proportionally to mass changes. Both increase when mass increases, and both decrease when mass decreases. In Figure 27, the data from the first week shows that the pressure reaches steady state relatively quickly. This is expected because the pressure limits are controlled: mass addition stops when the receiver pressure equals the compressor outlet pressure, and mass reduction stops when the receiver pressure falls below the expansion system's inlet pressure. Both boundaries are logical: the upper pressure limit will not be exceeded as the pressure needed to push more mass into the receiver is not available. The lower pressure limit should not be exceeded since significant pressure will not be available to operate the first turbine stage.

The temperature also seems to reach steady state quickly, centering at around 300 *K*. As expected, the temperature increases during charge processes and decreases during discharge processes.

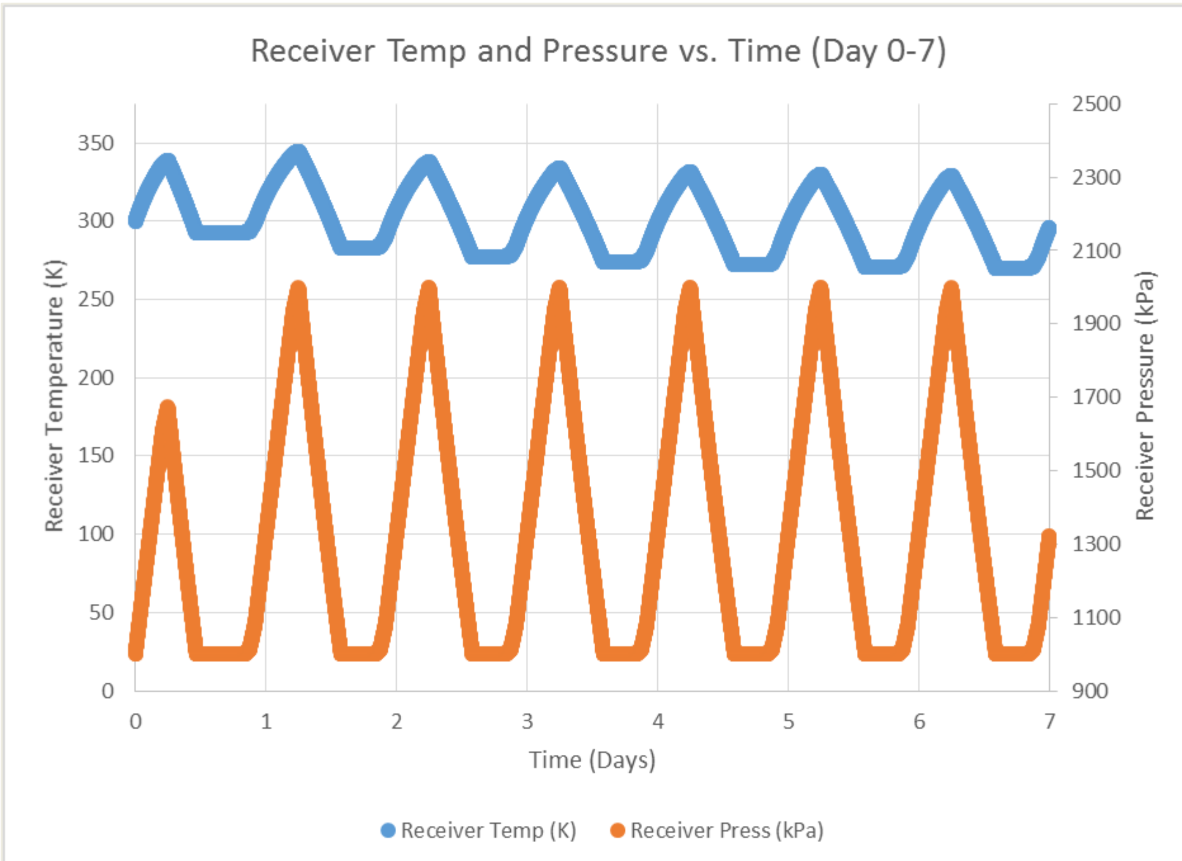


*Figure 27: Receiver Temperature & Pressure vs. Time (Day 0-7)*

The temperature and pressure data from the middle of the year, shown in Figure 28, are not significantly different from the data in the latter part of the first week.

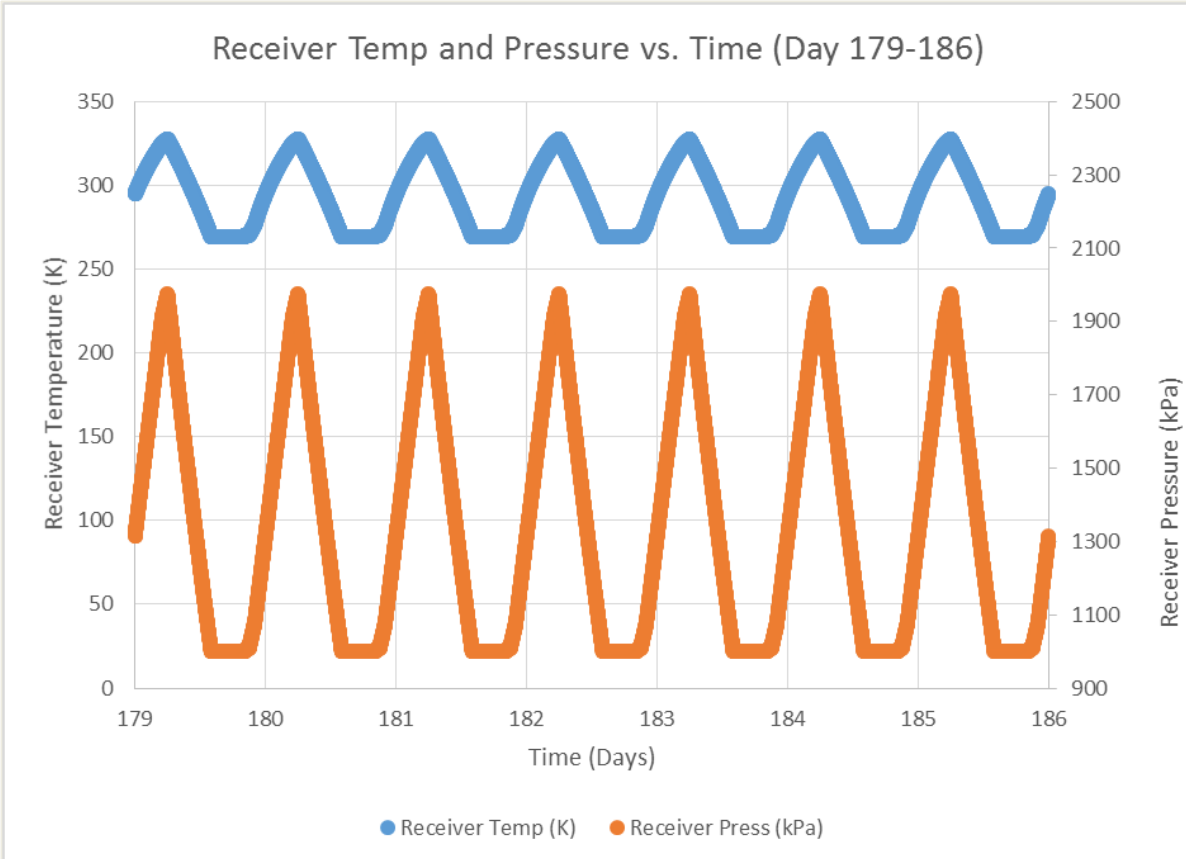


*Figure 28: Receiver Temperature & Pressure vs. Time (Day 179-186)*

A detail view of the middle (minimum) day of the year is shown in Figure 29, which shows the temperature and pressure process slopes more clearly. The charge process (increasing slopes) are not constant for either the temperature or the pressure, which is largely dependent on the variable mass flow rate of the compressor system. The discharge process (decreasing slope) is approximately constant due to the constant mass flow rate of the expansion system. The zero-slope region indicates a fully discharged receiver with no mass influx. All of the behaviors regarding temperature and pressure fluctuations, as well as mass flow and total mass appear reasonable.

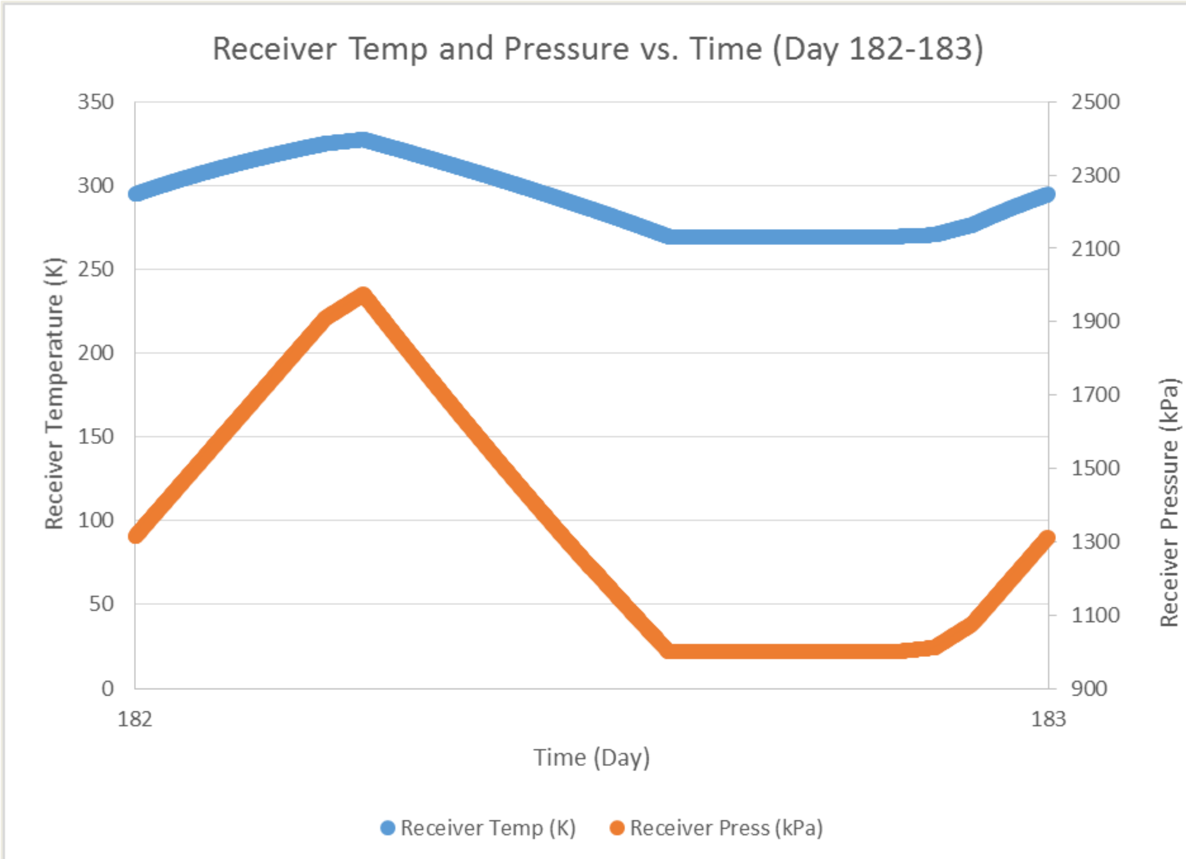


*Figure 29: Receiver Temperature & Pressure vs. Time (Day 182-183)*

### 4.3 Turbine Stage / Heat Exchanger Number and Reheat

The first system configuration set to run was that in which three turbine outlet temperature conditions were specified, and these three systems were run with 1-6 turbine/heat exchanger stages. The variables and their values are shown in Table 2. The turbine outlet conditions were specified explicitly for two of the systems, and the third was calculated based on an energy output constraint. This constraint was that the energy output of the expansion system would equal the energy input to the compression system, regardless of losses and energy required for reheat. The expansion system was sized to return the input energy of compression on the peak day. Therefore, it is quite likely that the expansion system is oversized, for whatever benefits or liabilities that it may carry.

*Table 2: Turbine Stage Number and Temperature Gradient Variations*

| Run | $Stages_{turb,run}$ | $P_{storage}$ (kPa) | $P_{turb,in}$ (kPa) | $T_{dp,in}$ (K) | $T_{turb,out}$ (K) | $\dot{m}_{turb}$(kg/s) | V ($m^3$) |
|---|---|---|---|---|---|---|---|
| 1 | 1-6 | 2000 | 1000 | 295 | 300 | 12.129 | 43300 |
| 2 | 1-6 | 2000 | 1000 | 295 | $T_{dp,comp}$ | 12.129 | 43300 |
| 3 | 1-6 | 2000 | 1000 | 295 | ? | 12.129 | 43300 |

The turbine outlet temperatures that were explicitly specified were chosen to represent both the ambient compressor inlet temperature and the minimum (dew point) temperature. The dew point temperature is significant because it is the temperature to which the stored air can fall to (at atmospheric pressure) before condensation and ice occur. All three outlet conditions assume a final outlet pressure of atmospheric pressure.

The receiver is sized at 43,300 $m^3$ for all three systems, as the differences in expansion temperatures and stage numbers does not affect this. Also, the receiver volume is greater than the sizing model specifications, on the order of 1.4 times. This will be discussed later, but using the calculated volume did not yield a system that was able to store all of the energy for the year. The receiver would reach the maximum pressure too quickly during charge cycles when the volume was below the threshold size. The receiver volume was iterated over 100 $m^3$ increments until a suitable volume was reached, which was used for simulation.

First, the amount of reheat required for each of the three systems is different, which is expected. In Figure 30, the total reheat energy (for all stages) and receiver temperature are shown as functions of time for the system with a 300 K turbine outlet temperature. This system figure is shown for a three stage expansion system. The reheat is at a minimum when the temperature is highest, and at a maximum when the temperature is lowest. There is no reheat when the expansion system is off or charging. This input energy is required to raise the air temperature from that of the receiver to the required turbine inlet temperature, which in this case is 391 *K*

.

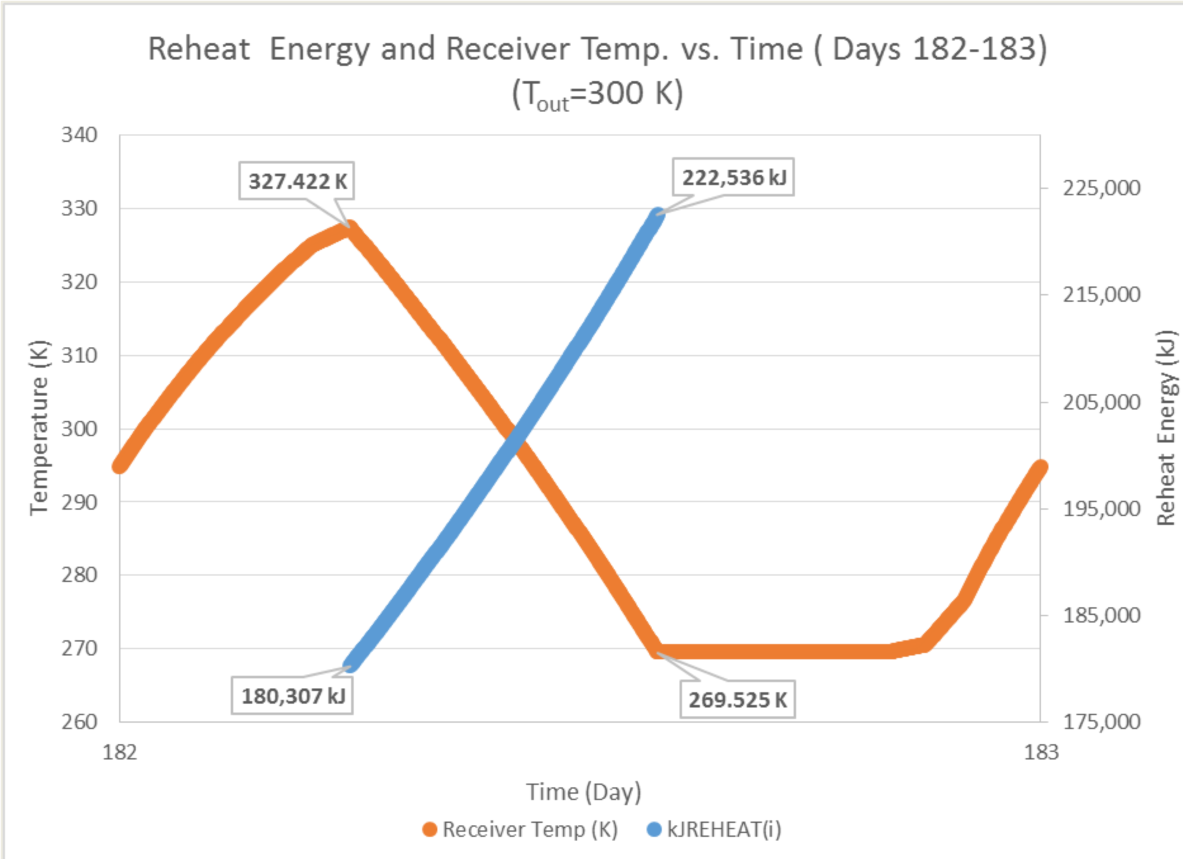


*Figure 30: Reheat Energy and Receiver Temperature vs. Time (Day 182-183) ($T_{out}$=300 K)*

For the three stage system that has outlet air at the dew point temperature, represented by Figure 31, the reheat energy is noticeably less than that of the first system. This makes sense, as the turbine inlet temperature (338 *K*) will not need to be as high to keep the outlet temperature at the dew point temperature (259 *K*) as it would to maintain the higher 300 K temperature. At its lowest receiver temperature, the air would have to be raised 68 K once for the turbine inlet and 79 K twice between stages.

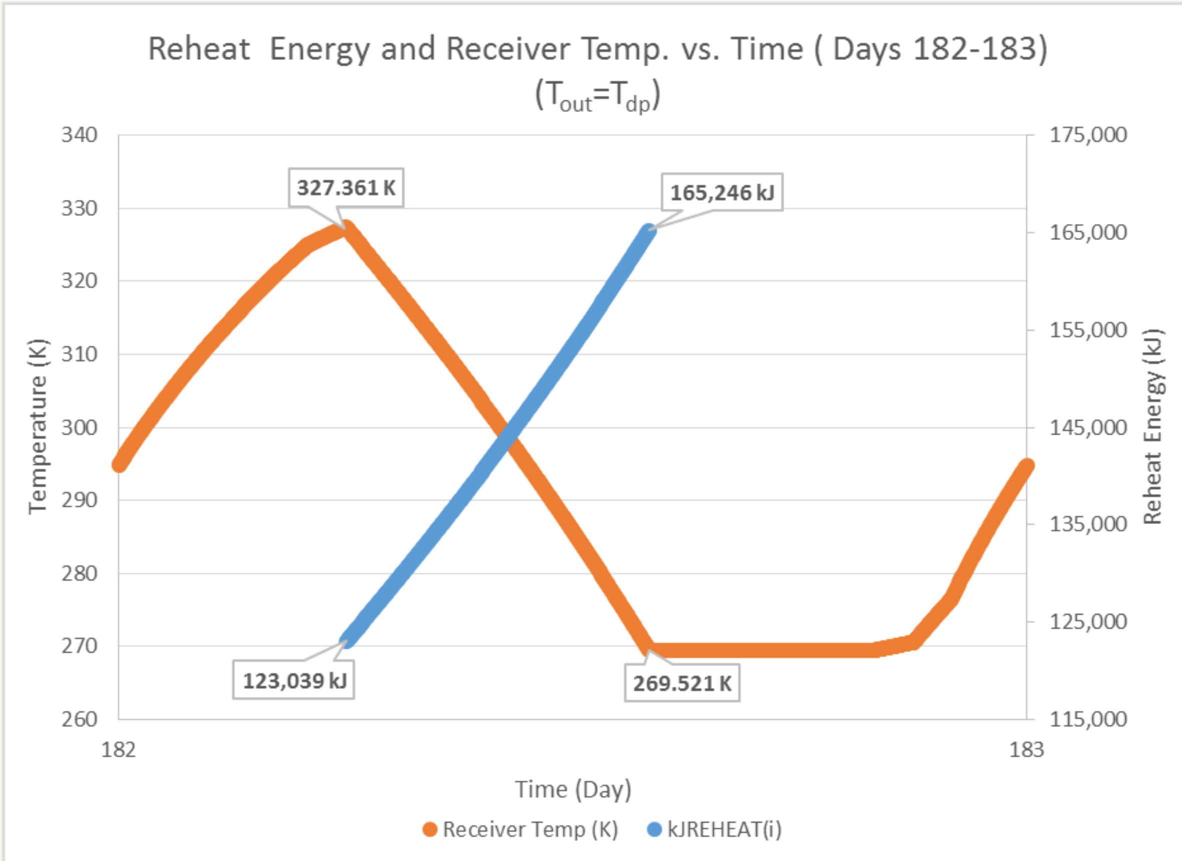


*Figure 31: Reheat Energy and Receiver Temperature vs. Time (Day 179-186) ($T_{out}$=$T_{dp}$)*

The third three stage system, represented by Figure 32, has reheat values of over 2 times greater than the first (300 *K*) system. This is needed to maintain an inlet temperature of 655 *K*. This high temperature is required to generate the amount of energy input to the system. Immediately, it becomes clear that a significant amount of energy will have to be expended on reheat to raise the temperature over a gradient of, at times, 385 *K*. This results in an outlet temperature of 509 *K*, so the interstage reheat will only require increasing the temperature 146 *K*; this is still significant. The other thing to note is that air at 509 *K* and atmospheric pressure is exiting the last turbine, which carries heat that is assumed to be unused.

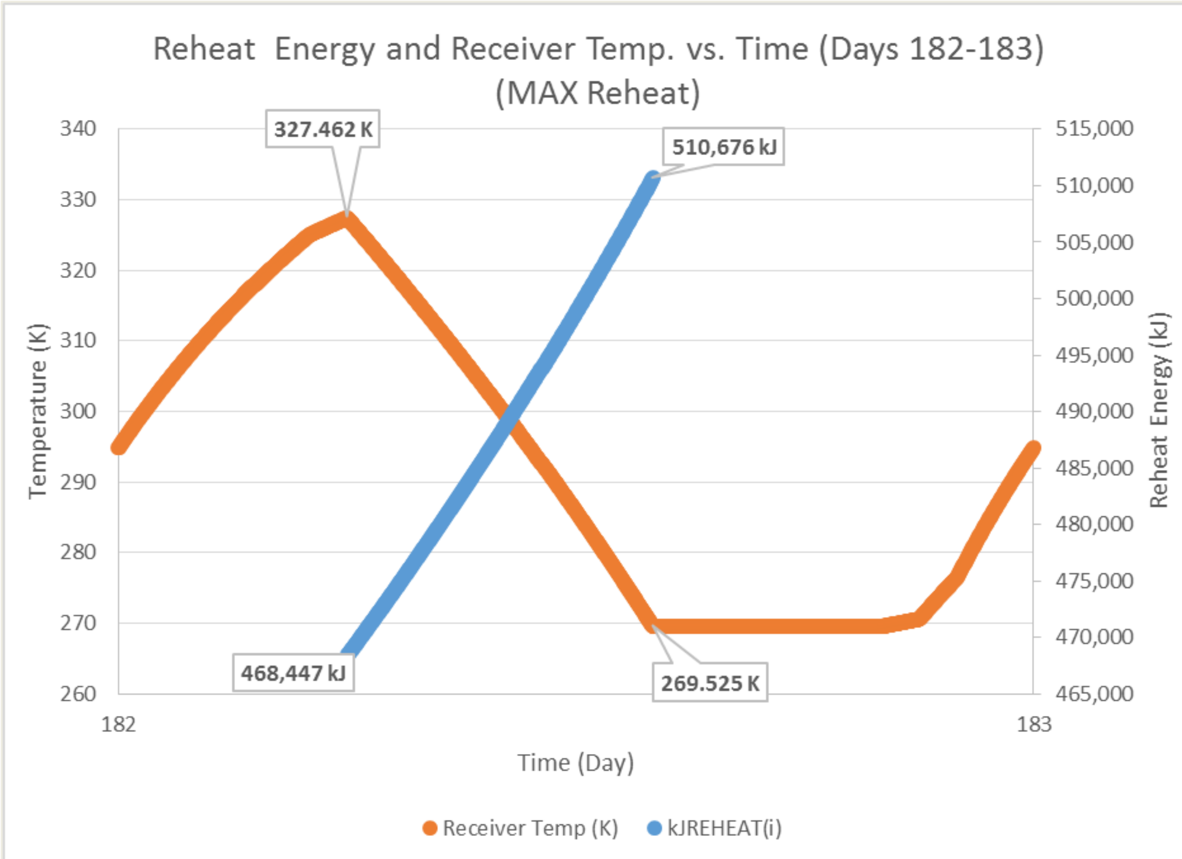


*Figure 32: Reheat Energy and Receiver Temperature vs. Time (Day 179-186) (MAX Reheat*

Figure 33 shows a lot of information for the system with full reheat. It shows the turbine inlet and outlet temperatures, the change in specific enthalpy across each turbine, and the amount of energy output from the expansion system as a total and a percentage of energy input, all as functions of the turbine/heat exchanger stage number. These figures are all based on the sizing model, and particular attention should be paid to the amount of input energy "returned." This percentage does not account for any input or output energy between compression and expansion, such as reheat. It is merely the "fraction of input energy" generated by the expansion system. It is also only valid for the peak day, as it is calculated by the sizing model. As stages are added to this system, the necessary reheat and the power out per stage reduce. The outlet temperature increases as well, which is the primary reason for the lost work output per stage. The inlet and outlet temperatures are converging to approximately 570 *K*, leaving a smaller and smaller temperature gradient with each additional expansion stage. With an infinite number of interheated stages, the system would reach isothermal expansion at the convergence temperature.

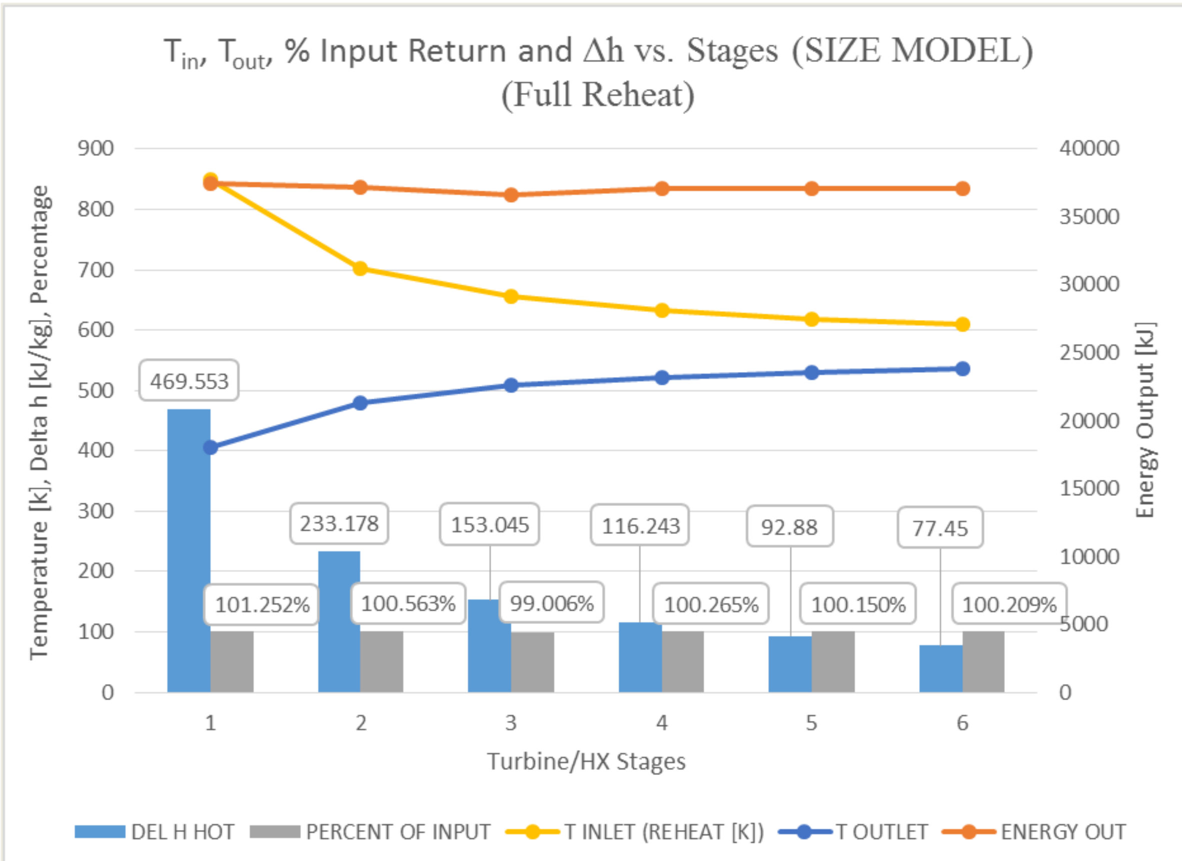


*Figure 33: $T_{in}$, $T_{out}$, % Input Return and Δh vs. Stages (SIZE MODEL) (Full Reheat)*

For the system with the 300 $K$ outlet temperature, shown in Figure 34, the inlet and outlet temperatures are converging to 300 $K$. Both the energy output and the percentage of input reduce with stage number. The “fraction of input energy” generated seems to be converging to approximately 55%. The change in specific enthalpy across the turbine also decreases with each additional turbine stage.

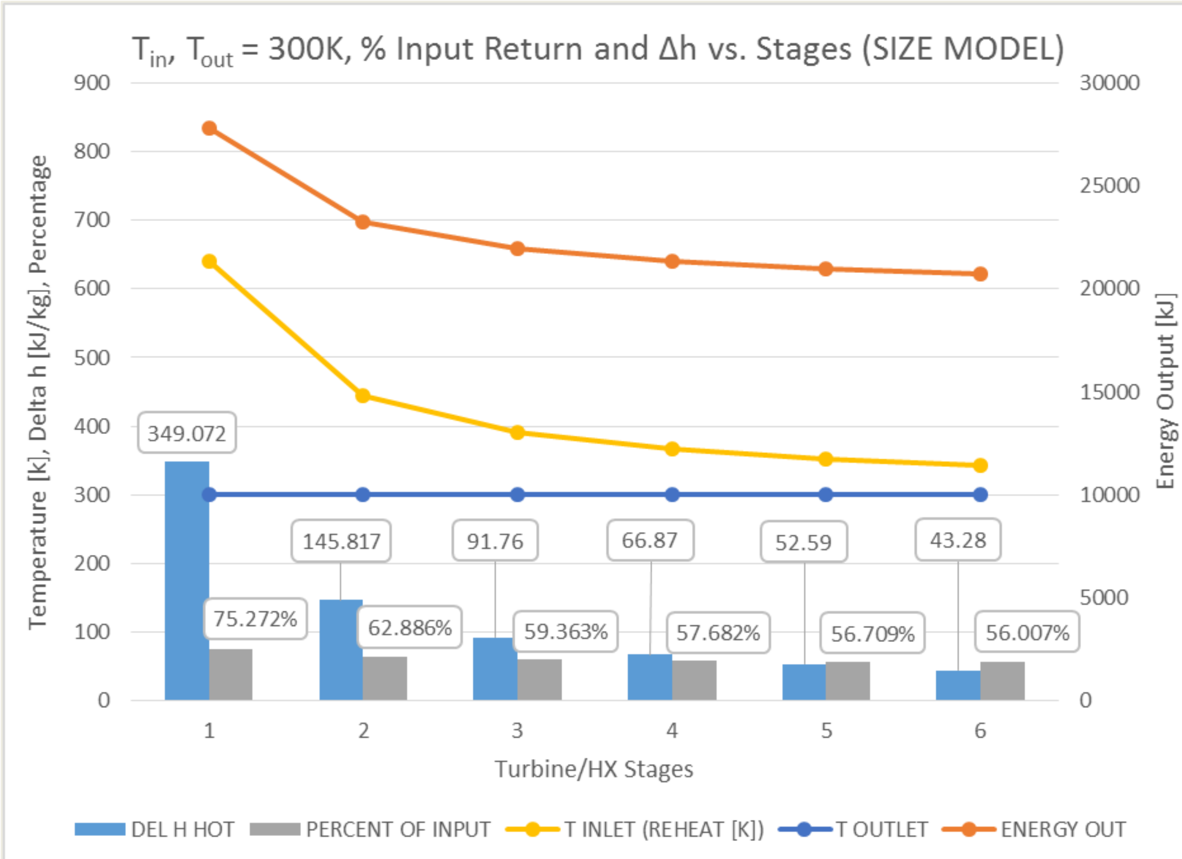


*Figure 34: $T_{in}$, $T_{out}$ = 300 K, % Input Return and Δh vs. Stages (SIZE MODEL)*

With the "dew point" system, shown in Figure 35, the input returns are even lower. This is expected because the energy inputs between compression and expansion are minimal. The "fraction of input energy" generated seems to converge at approximately 48%, while the inlet and outlet temperatures are converging at the dew point temperature (259 K). The reduction in the temperature gradient drives the reduction in the specific enthalpy gradient, and since neither reduction is strictly linear, the single stage turbine system will return more energy than will a multiple number of stages. The tradeoff is, of course, input energy for reheat. More heat must be supplied initially to run one big turbine than to run multiple smaller ones.

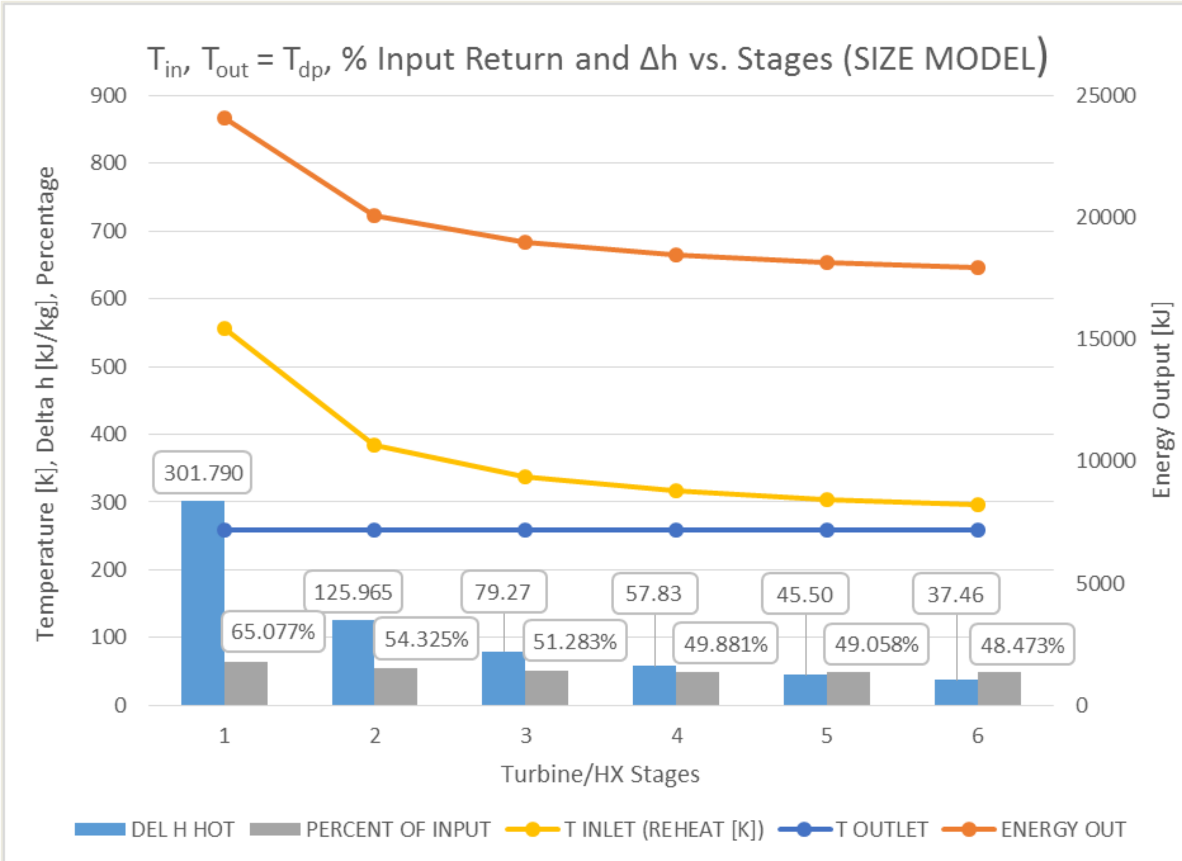


*Figure 35: $T_{in}$, $T_{out}$ = $T_{dp}$, % Return and Δh vs. Stages (SIZE MODEL)*

The effects of the reheat energy become clear when the results of the system simulations are displayed. The system with the maximum reheat and maximum energy output is represented in Figure 36. This shows total output energy from the expansion system as percentage of input energy (compression and reheat). This is the "input return." It also shows the power of the expansion system and the size of the reheat mechanism necessary to supply the required energy to the air, prior to expansion. These are all functions of the turbine stage number. Perhaps not surprisingly, in order to output from the expansion system the same amount of energy that was put into the compression system, between 53.7 % and 89.5 % ADDITIONAL energy has to be put into the system in the form of reheat. Matters only get worse with successive stages, and this is likely caused by the decrease in power output with each successive stage and the unused heat exiting the final turbine stage. The temperature of this outlet air only increases as stages are added.

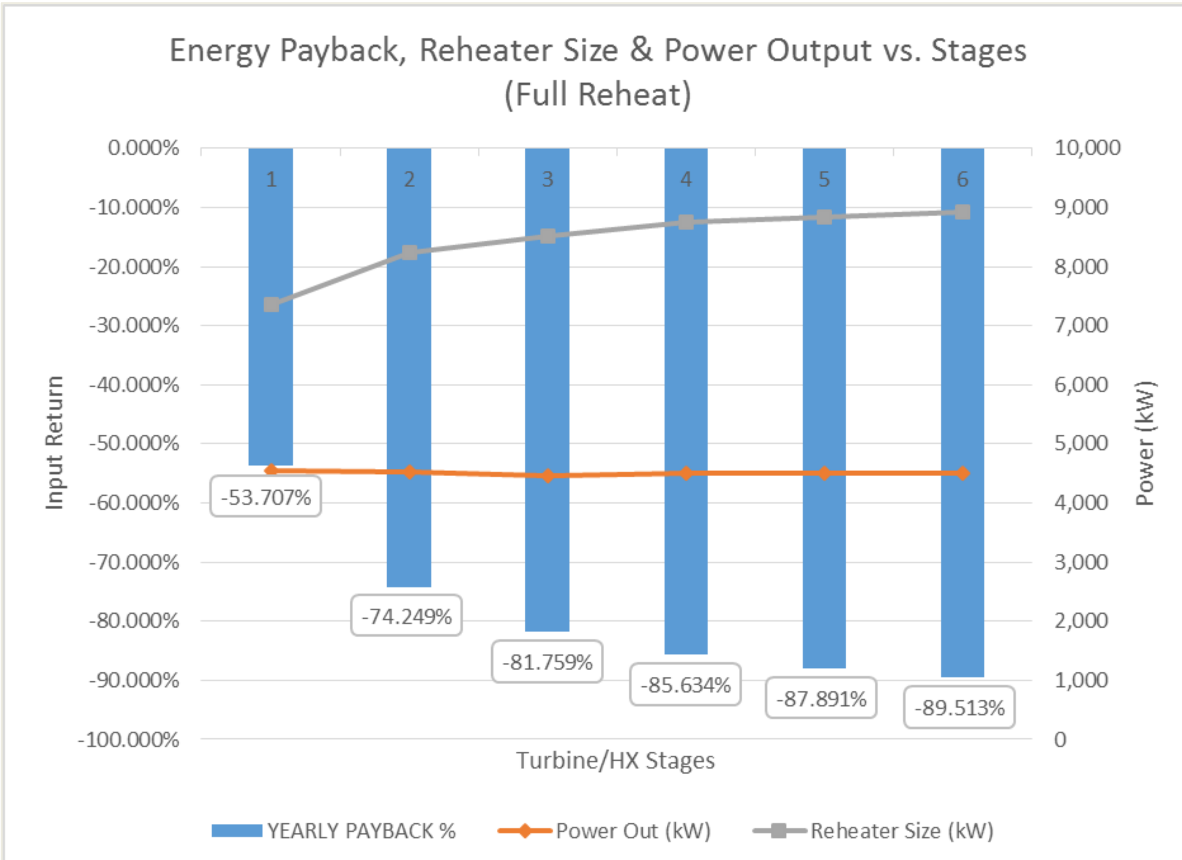


*Figure 36: Energy Payback, Reheater Size & Power Output vs. Stages (Full Reheat*

Figure 37 shows the input returns, system power out and system reheater size for the system with a 300 *K* turbine outlet temperature. The trends with a fixed output temperature (300 K) tell a different story that those of the maximum reheat system. With successive stage numbers, the returns increase while power out and reheater size decrease. This is most likely due to a smaller required reheat, and though the returns are still negative, it show a more promising trend than the previous case. It is not clear that the returns converge to a limit, but it is unlikely that more than six stages will be considered for marginal gains.

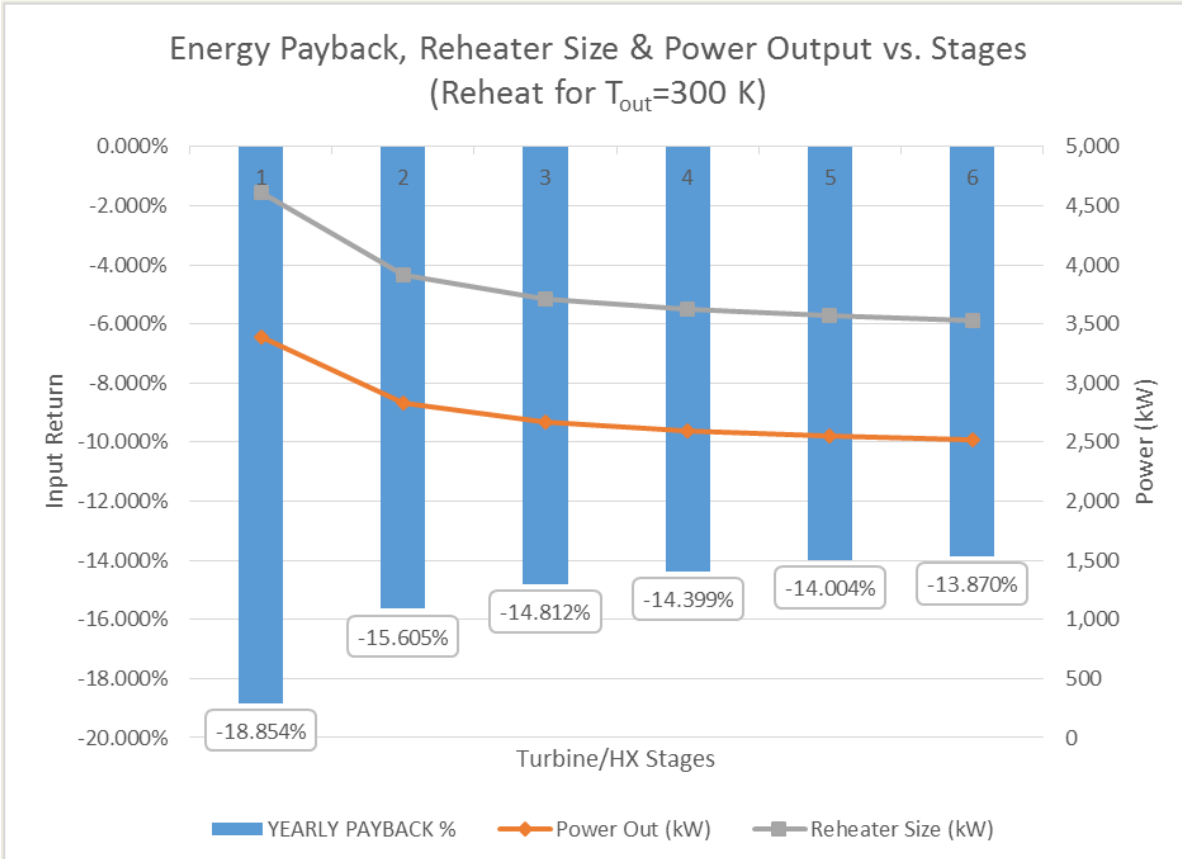


*Figure 37: Energy Payback, Reheater Size & Power Output vs. Stages (Reheat for $T_{out}$=300 K)*

Finally, Figure 38 shows the trends for the system with the minimum possible reheat. This system, though close, does not make the jump from red to black. It appears that the limit of input return is zero *for this case*. This makes sense, because after losses from inefficiencies and heat rejection, the energy left in the air to be expanded is relatively low. It is the heat added to the compressed air that increases its enthalpy and its ability to do work. The process of reheating the air in order to keep its temperature above the dew point apparently requires slightly more energy in heat than it produces in work.

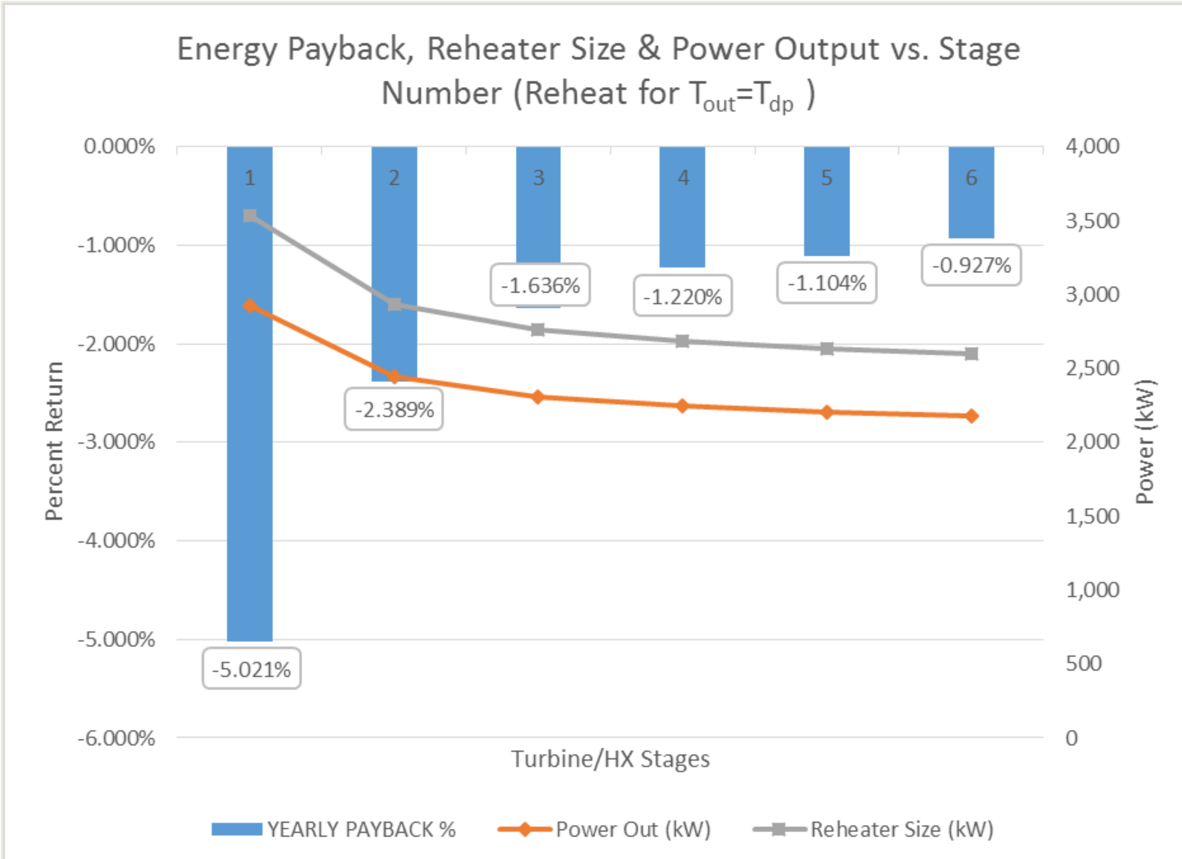


*Figure 38: Energy Payback, Reheater Size & Power Output vs. Stages (Reheat for $T_{out}$=$T_{dp}$ )*

### 4.4 Turbine Inlet Pressure

The second system configuration set changed the expansion system inlet pressure from 1000 *kPa* to 1900 *kPa* in increments of 100 *kPa* (1 bar), while keeping the storage pressure at 2000 *kPa*. The input variables and their values are listed in Table 3.

*Table 3: Turbine Inlet Pressure Variations*

| Run | $Stages_{turb,run}$ | $P_{storage}$ (kPa) | $P_{turb,in}$ (kPa) | $T_{dp,in}$ (K) | $T_{turb,out}$ (K) | $\dot{m}_{turb}$(kg/s) | V ($m^3$) |
|---|---|---|---|---|---|---|---|
| 4 | 3 | 2000 | 1000 | 295 | 300 | 12.129 | 43300 |
| 5 | 3 | 2000 | 1100 | 295 | 300 | 12.129 | 48100 |
| 6 | 3 | 2000 | 1200 | 295 | 300 | 12.129 | 54200 |
| 7 | 3 | 2000 | 1300 | 295 | 300 | 12.129 | 61900 |
| 8 | 3 | 2000 | 1400 | 295 | 300 | 12.129 | 72300 |
| 9 | 3 | 2000 | 1500 | 295 | 300 | 12.129 | 86700 |
| 10 | 3 | 2000 | 1600 | 295 | 300 | 12.129 | 108500 |
| 11 | 3 | 2000 | 1700 | 295 | 300 | 12.129 | 144900 |
| 12 | 3 | 2000 | 1800 | 295 | 300 | 12.129 | 218000 |
| 13 | 3 | 2000 | 1900 | 295 | 300 | 12.129 | 440600 |

As a result of these changes, the receiver volume must be increased in order to store all of the yearly excess energy. This requirement is due to two things: the larger amount of mass required to maintain a higher "discharged" receiver pressure (given the same conditions); and the smaller pressure gradient between the storage pressure and the turbine inlet pressure. The mass required to maintain the higher base pressure will vary according to the volume of the receiver and its temperature; a tiny volume may fulfill the same pressure requirements as a large tank, but it is unlikely to fulfill the mass flow or time requirements. The smaller tank will inevitably "fill up" sooner than the larger tank. For example, if a receiver, sized correctly for a lower turbine inlet pressure, was used for a system with a higher inlet pressure requirement, it would reach its new "discharged" state sooner than it would under the previous regime. The charge and discharge processes would be abbreviated, proportional to the smaller pressure gradient, and less mass would be able to be stored during each charge process. This leaves less mass available for expansion, and depending on the excess energy available, potentially less energy stored. As the gradient between the storage pressure and turbine inlet pressure decreases, the receiver volume requirement increases.

These changes are shown in Figure 39. The minimum receiver volume is calculated by the sizing model using the ideal gas equation. This calculation assumes a steady state discharged receiver temperature of 300 *K*. This size is also based on one charge process over the course of the peak day, and does not consider any discharge processes. This calculation, since it is performed in the sizing model, does not consider the yearly excess energy data; it does not consider any schedule other than the peak day. Therefore, at least for these yearly excess energy data input, the receiver volumes are too small. For all of the turbine inlet pressures, the volume required to store all of the excess energy over the course of the year is approximately 1.4 times larger than the calculated minimum volume. These ratios are also shown in Figure 39 below.

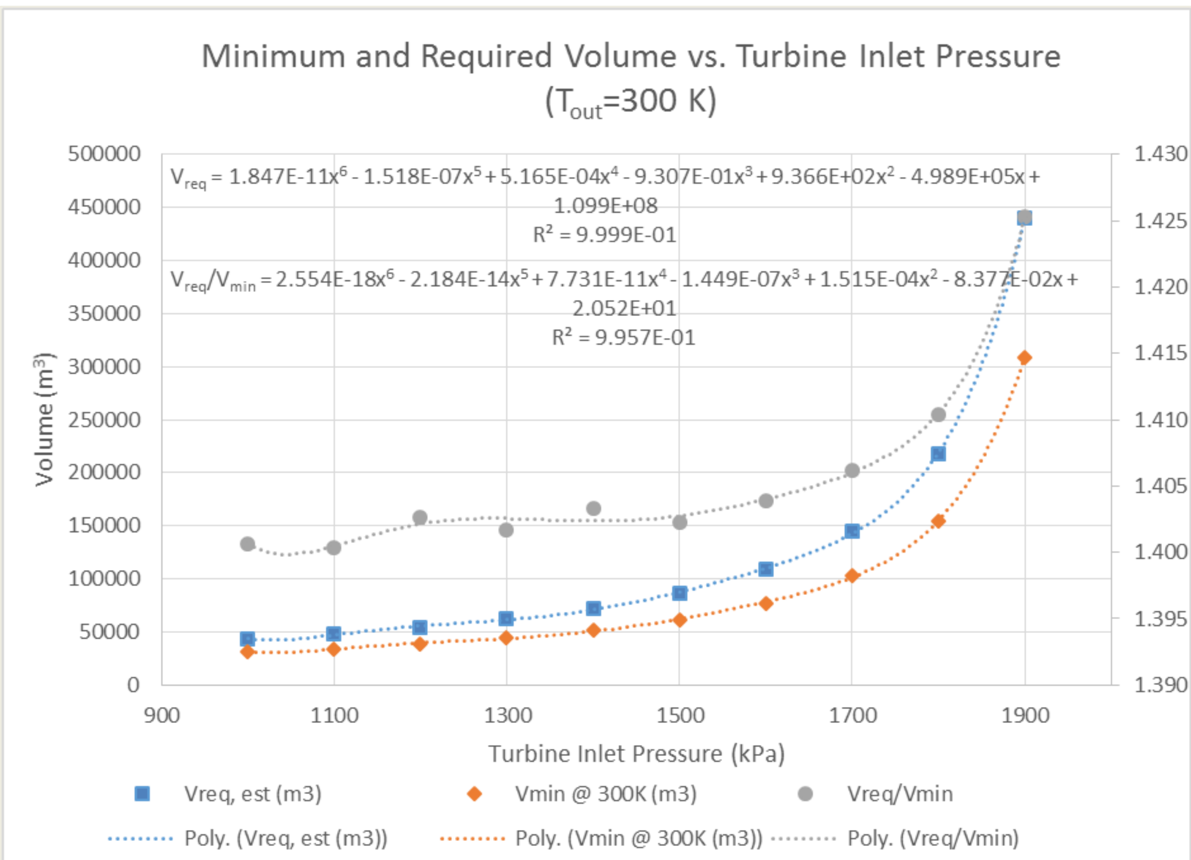


*Figure 39: Minimum and Required Volume vs. Turbine Inlet Pressure ($T_{out}$=300 K)*

As the turbine inlet pressure increases, the minimum and maximum receiver temperatures converge on the outlet temperature of the compressor. This is due to the larger receiver volume and the proportionally smaller effect that the compressed mass added to or removed from the receiver has on temperature and pressure. The turbine outlet temperature coincides with, for this simulation, the compressor outlet temperature. This is merely a coincidence; given different constraints for the turbine outlet temperature, the receiver temperatures will still function independently of the expansion array.

The factor that does affect the expansion array is the receiver temperature. Depending on the flow-induced temperature fluctuations in the receiver, the first reheat stage may have to put more or less energy into the air to raise its temperature. Since the temperature fluctuations will be less severe given a larger volume, the reheat system will probably not have to cycle up and down to compensate. The unexpected trend observed in the expansion system was the increased turbine inlet temperature. It corresponds to a consistent 122 $K$ increase in temperature over the minimum receiver temperature. This behavior is unexpected because it would seem that more power could be derived from a larger pressure gradient through the expansion system than a smaller one. This does, however, translate to a higher power output, as shown in Figure 41, and the increased inlet temperature is likely to maintain the constant turbine outlet temperature of 300 K.

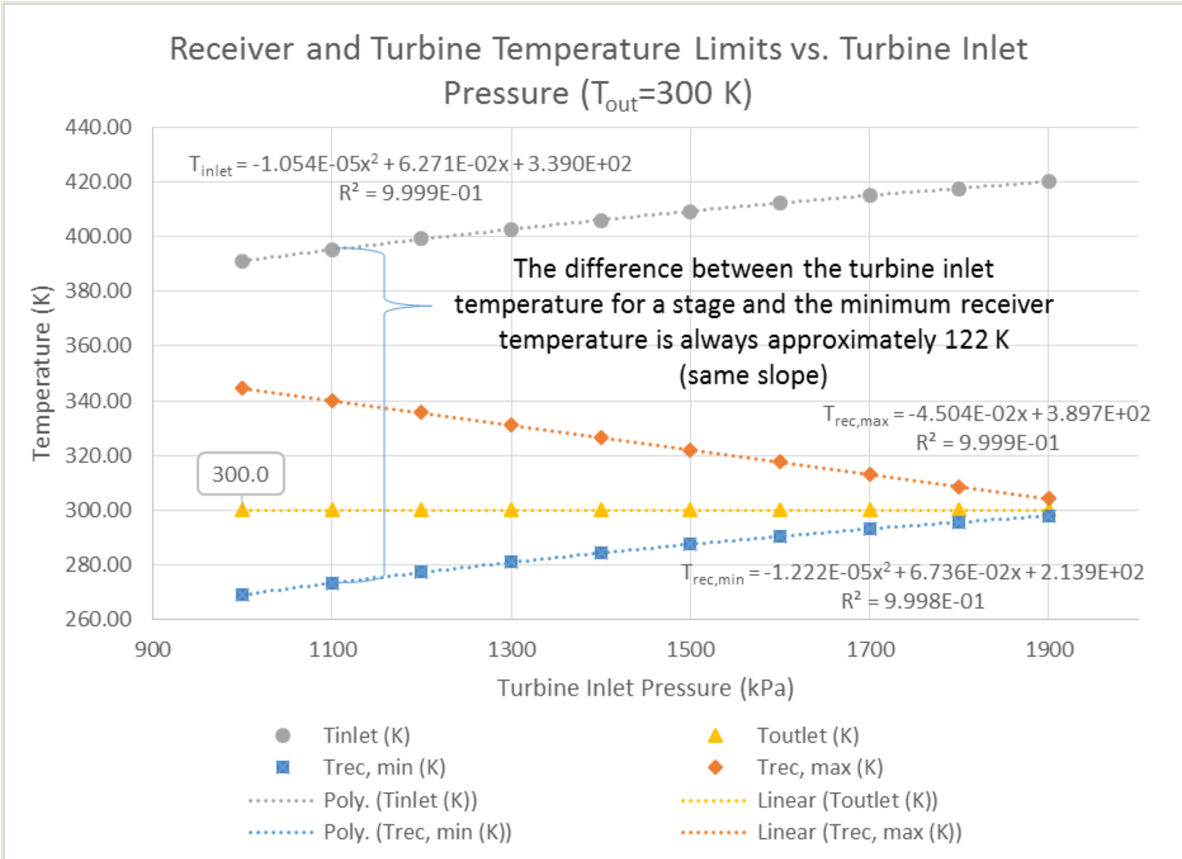


*Figure 40: Receiver and Turbine Temperature Limits vs. Turbine Inlet Pressure ($T_{out}$=300 K)*

Finally, most likely due to the increased inlet temperature requirement, is the negative trend in energy input return to increased inlet pressure. It seems that, at least for this case, the increase in power output is not enough to overcome the need for the required reheat energy.

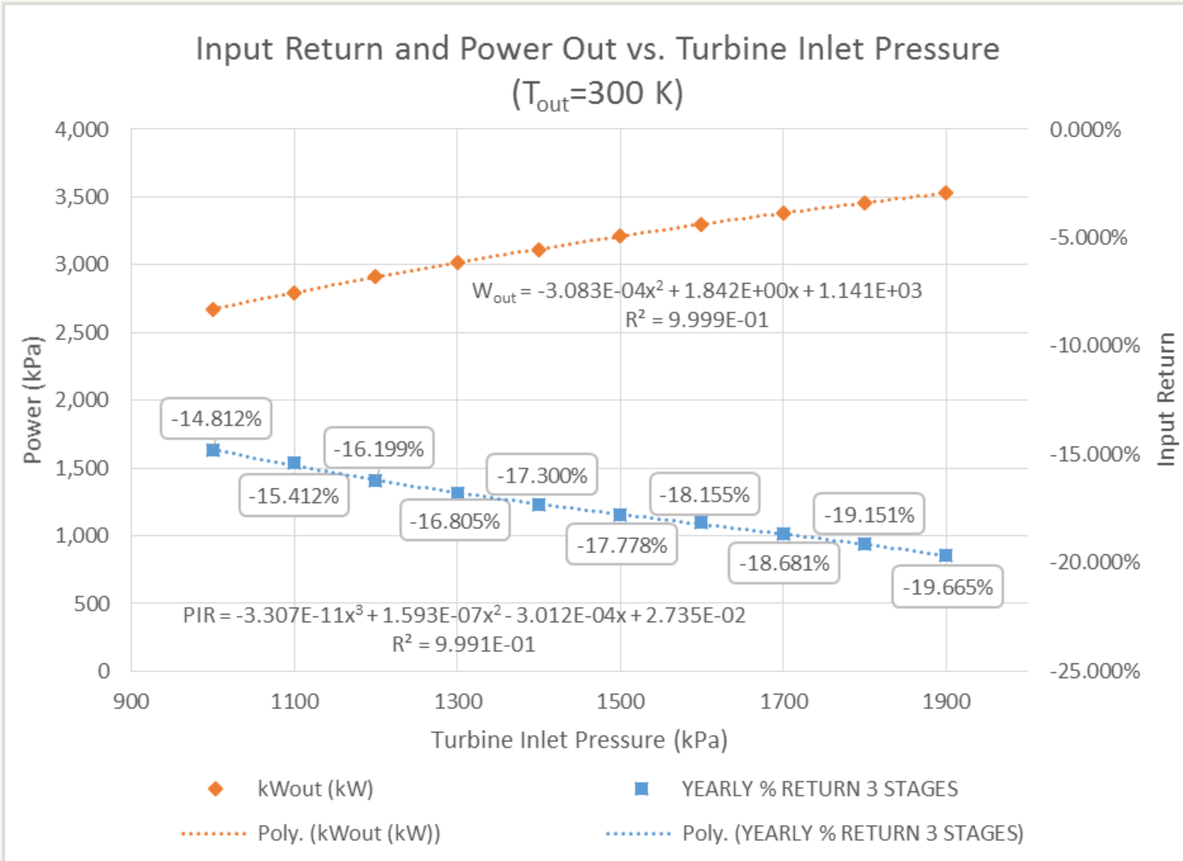


*Figure 41: Input Return and Power Out vs. Turbine Inlet Pressure ($T_{out}$=300 K)*

### 4.5 Storage (Compressor Outlet) Pressure

The third simulated system configuration set iterated changes in the storage pressure (compressor outlet pressure) over the range of 2000 *kPa* to 10,000 *kPa*, in increments of 1000 *kPa* (10 bar). The variable values for the set are listed in Table 4. Two things that should be noted about the data in the table: the receiver volumes were changed in a similar way to the volumes in the turbine inlet pressure configuration set and the expansion system mass flow rates changed as a result of the storage (compressor) pressure changes.

*Table 4: Storage Pressure Variations*

| Run | $Stages_{turb,run}$ | $P_{storage}$ (kPa) | $P_{turb,in}$ (kPa) | $T_{dp,in}$ (K) | $T_{turb,out}$ (K) | $\dot{m}_{turb}$(kg/s) | V ($m^3$) |
|---|---|---|---|---|---|---|---|
| 14 | 3 | 2000 | 1000 | 295 | 300 | 12.129 | 43270 |
| 15 | 3 | 3000 | 1000 | 295 | 300 | 10.466 | 37334 |
| 16 | 3 | 4000 | 1000 | 295 | 300 | 9.511 | 33937 |
| 17 | 3 | 5000 | 1000 | 295 | 300 | 8.868 | 31635 |
| 18 | 3 | 6000 | 1000 | 295 | 300 | 8.397 | 29932 |
| 19 | 3 | 7000 | 1000 | 295 | 300 | 8.028 | 28647 |
| 20 | 3 | 8000 | 1000 | 295 | 300 | 7.732 | 27585 |
| 21 | 3 | 9000 | 1000 | 295 | 300 | 7.487 | 26695 |
| 22 | 3 | 10000 | 1000 | 295 | 300 | 7.275 | 25953 |

The volume changes are shown in Figure 42. The minimum volumes were calculated in the sizing model, and the required volumes were found by separate iteration over 1 $m^3$ steps. The required and minimum volumes are decreasing with respect to the storage pressure, and have a positive linear relationship with each other. This makes sense, as the higher the pressure of the compressed air, given the same mass and temperature state values, the more dense the given parcel will be. This will take less space than the same amount of mass at a lower pressure. Also, since the same amount of energy is available for air compression, and higher pressures require more compression work for the same amount of mass, less mass is compressed, as shown in Figure 43. Total mass compressed and mass flow rate decrease as functions of storage pressure. If less, high pressure air is being stored, it is logical that less volume would be necessary.

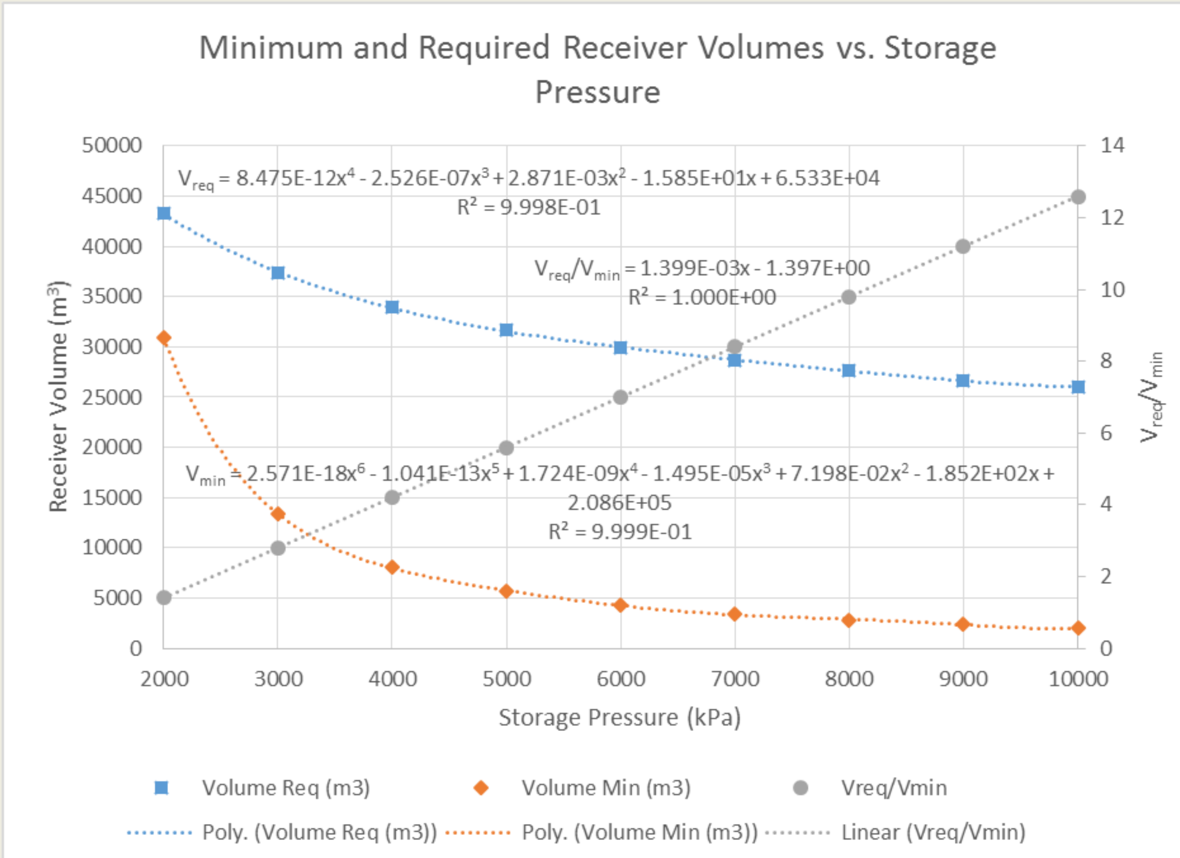


*Figure 42: Minimum and Required Receiver Volumes vs. Storage Pressure*

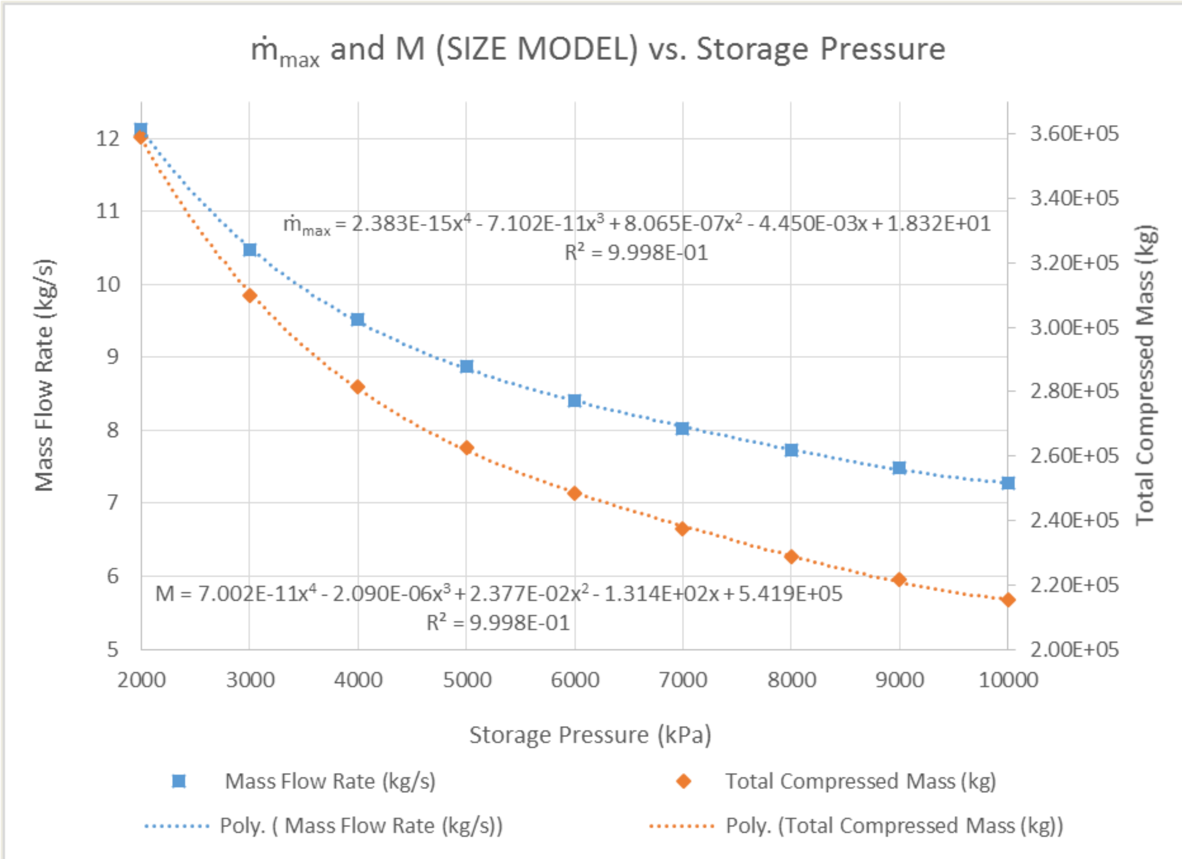


*Figure 43: $\dot{m}_{max}$ and Total Mass (SIZE MODEL) vs. Storage Pressure*

Finally, in Figure 44, the input return and energy output are shown as functions of storage pressure. The percentage of input return increases as storage pressure increases and the energy input decreases. The percentage return seems to be converging to -8 %, which is still a net loss for the overall system configuration. Both functional relationships may be a result of a decreasing amount of required reheat energy, which is likely a result of a lower mass flow rate.

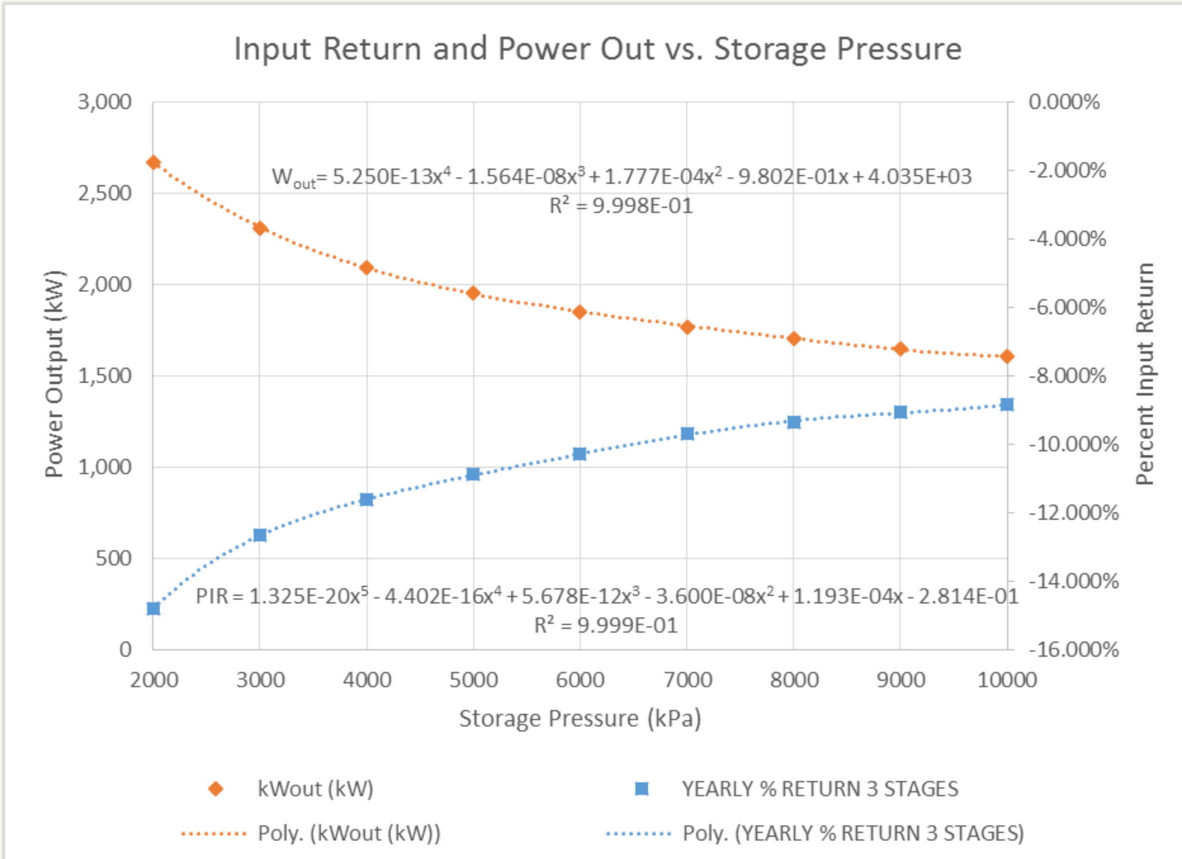


*Figure 44: Input Return and Power Out vs. Storage Pressure*

It should also be noted that in all of the storage pressure configurations, the maximum and minimum receiver temperatures and pressures were the same for all of the systems (345 *K*, 269 *K*, 2000 *kPa*, 1000 *kPa*). The inlet and outlet temperatures are also the same across all iterated systems (391 *K*, 300 *K*). This seems to be a result of the excess power data and the proportionality of the receiver volume and compressed mass to the compressor pressure. One would expect the pressure to rise above 10 bar higher than the turbine outlet pressure, but it never does. When the system is run with a receiver volume smaller than the required size for the given storage/compressor pressure, the receiver pressure rises above 2000 *kPa*. This behavior deserves further investigation. This may prove to be a method for determining the optimum compressor outlet pressure or expansion mass flow rate.

## 4.6 Storage and Turbine Inlet Pressures

The fourth system configuration set iterates both compressor outlet (storage) pressure and expansion system inlet pressure pairs, from 3000/2000 *kPa* to 10,000/9000 *kPa*. These eight pressure combinations all have a 10 bar gradient between the storage and inlet pressures. These changes are listed in Table 5 and it is no surprise that the receiver volumes and mass flow rates changed with increased compressor pressures.

*Table 5: Storage and Turbine Input Pressure Variations*

| Run | $Stages_{turb,run}$ | $P_{storage}$ (kPa) | $P_{turb,in}$ (kPa) | $T_{dp,in}$ (K) | $T_{turb,out}$ (K) | $\dot{m}_{turb}$(kg/s) | V ($m^3$) |
|---|---|---|---|---|---|---|---|
| 23 | 3 | 3000 | 2000 | 295 | 300 | 10.466 | 37364 |
| 24 | 3 | 4000 | 3000 | 295 | 300 | 9.511 | 33988 |
| 25 | 3 | 5000 | 4000 | 295 | 300 | 8.868 | 31739 |
| 26 | 3 | 6000 | 5000 | 295 | 300 | 8.395 | 30074 |
| 27 | 3 | 7000 | 6000 | 295 | 300 | 8.028 | 28777 |
| 28 | 3 | 8000 | 7000 | 295 | 300 | 7.731 | 27729 |
| 29 | 3 | 9000 | 8000 | 295 | 300 | 7.484 | 26876 |
| 30 | 3 | 10000 | 9000 | 295 | 300 | 7.274 | 26148 |

The required receiver volumes seem to follow a functional increase over the minimum calculated volume, by a factor of approximately 1.4 times. This is shown in Figure 45 below.

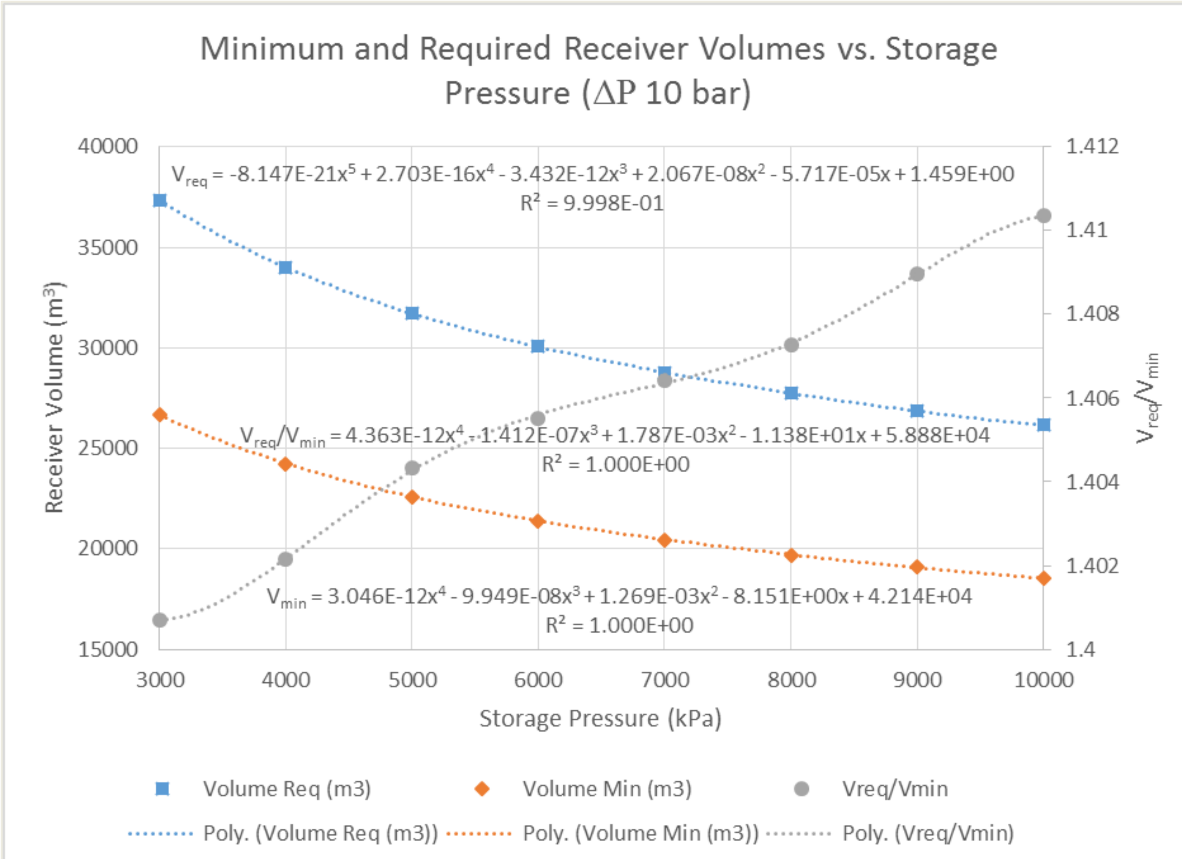


*Figure 45: Minimum and Required Receiver Volumes vs. Storage Pressure (ΔP 10 bar)*

As shown in Figure 46, the maximum receiver pressure increases linearly and one-to-one, according to the storage pressure, and the receiver temperatures decrease as both changed pressures increase. The reason for the temperature decrease may be due to the decreased mass flow rate through the compression system. Since the air is cooled to 300 K regardless of the output pressure, a lower mass flow rate would tend to affect the receiver temperature less than a higher flow rate. The increasing receiver pressure makes sense, because even though the storage pressure increases well above 2000 kPa, the 10 bar pressure gradient follows this increase. The receiver only ever undergoes a 10 bar decrease, which is the main difference between this configuration set and the storage pressure configuration set.

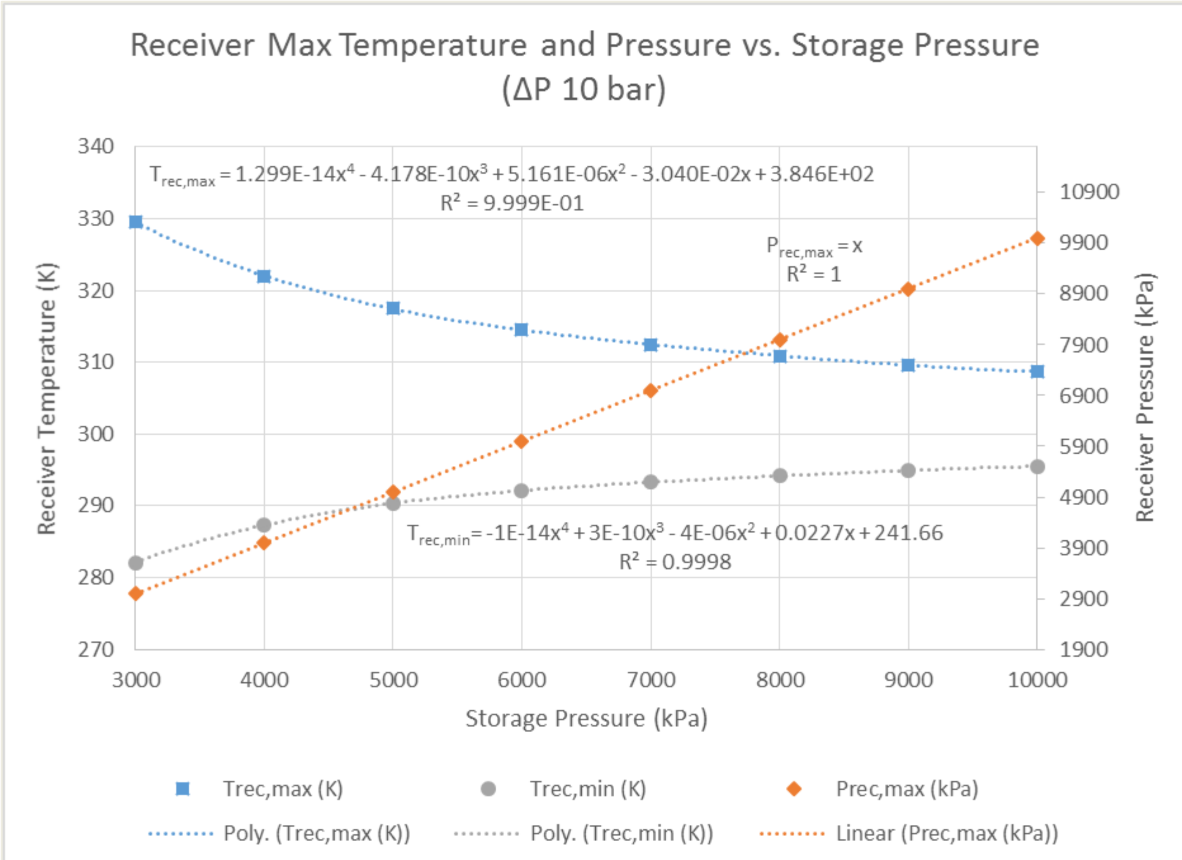


*Figure 46: Max Receiver Temperature and Pressure vs. Storage Pressure (ΔP 10 bar)*

The total compressed mass and the maximum compressor mass flow rate, both from the sizing model, decrease as a function of the increasing pressures, as shown in Figure 47. In fact, Figure 43 is an equivalent figure, save for the 2000 *kPa* storage pressure data point. This is because the mass flow rate and total compressed mass are characteristics of the compression system, not the receiver.

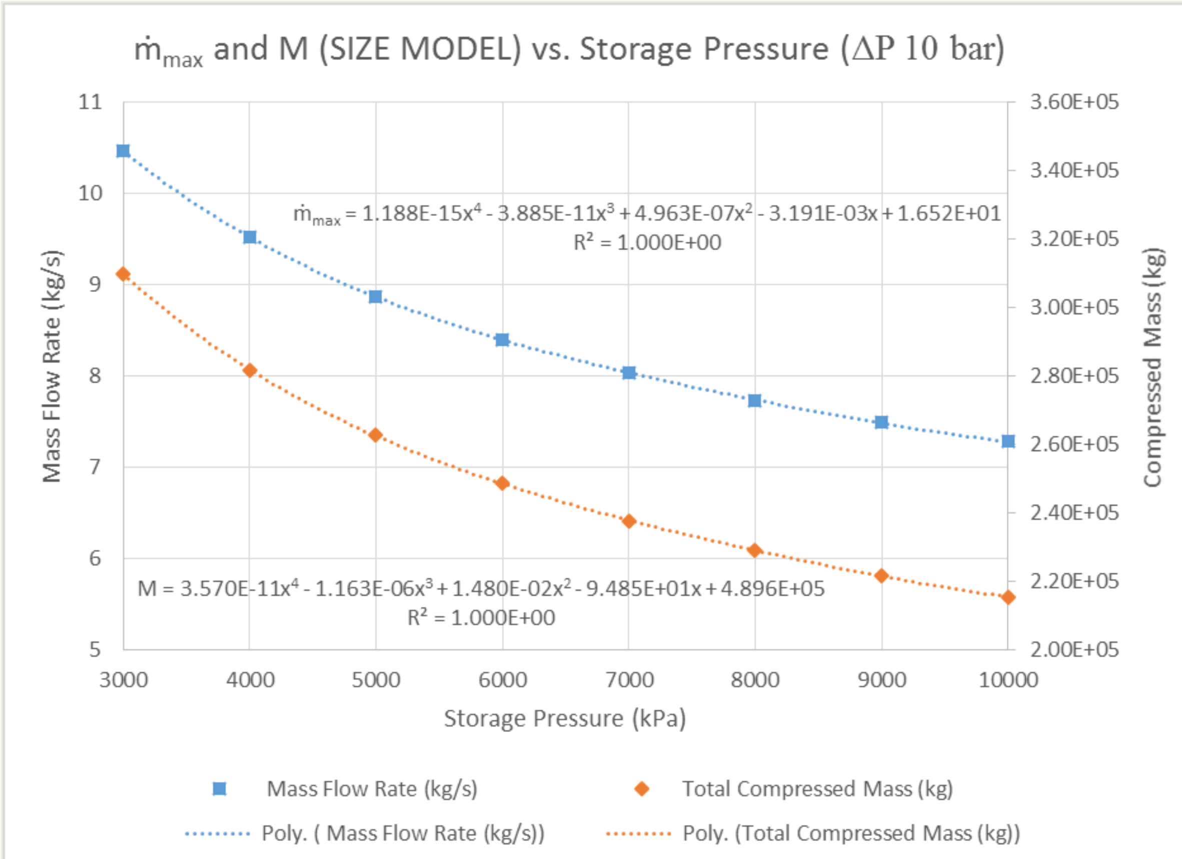


*Figure 47: $\dot{m}_{max}$ and M (SIZE MODEL) vs. Storage Pressure (ΔP 10 bar)*

Finally, the energy input return and the power output of the expansion system are shown in Figure 48. The power output seems to be converging to approximately 3525 *kW*. The decreasing input return, as was the case for the increasing turbine inlet pressure configuration set, is likely due to an increased need for reheat energy, as shown in Figure 49. As the turbine inlet pressure increases, the required inlet temperature increases as well. This leads to a dichotomy: less reheat energy should be needed for a lower mass flow rate through the expansion system, but increased turbine inlet pressure dictates higher inlet temperatures. Unlike the configuration set for turbine inlet pressure, this set may have a lower energy return limit of approximately -19.5 %. This is still relatively low. This convergence, if present, may be due to the decreasing expansion system mass flow rate.

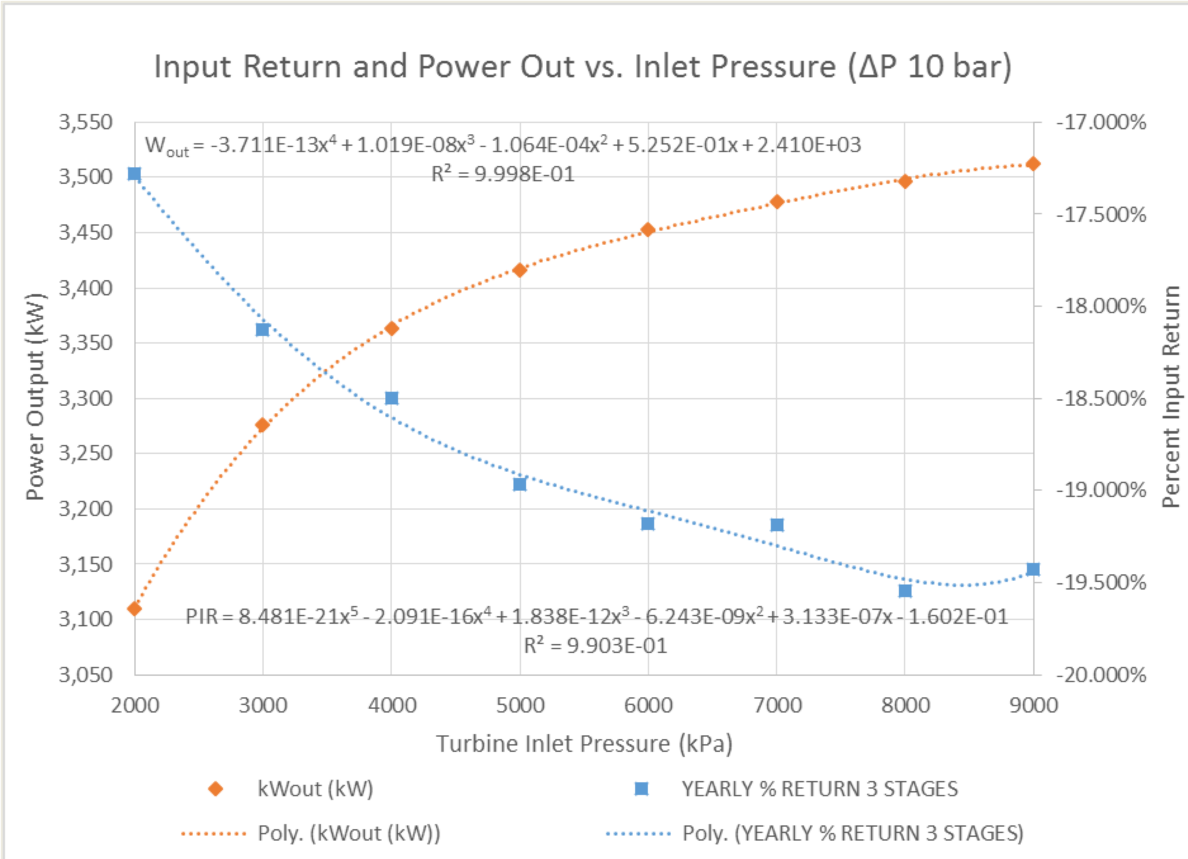


*Figure 48: Input Return and Power Out vs. Inlet Pressure (ΔP 10 bar)*

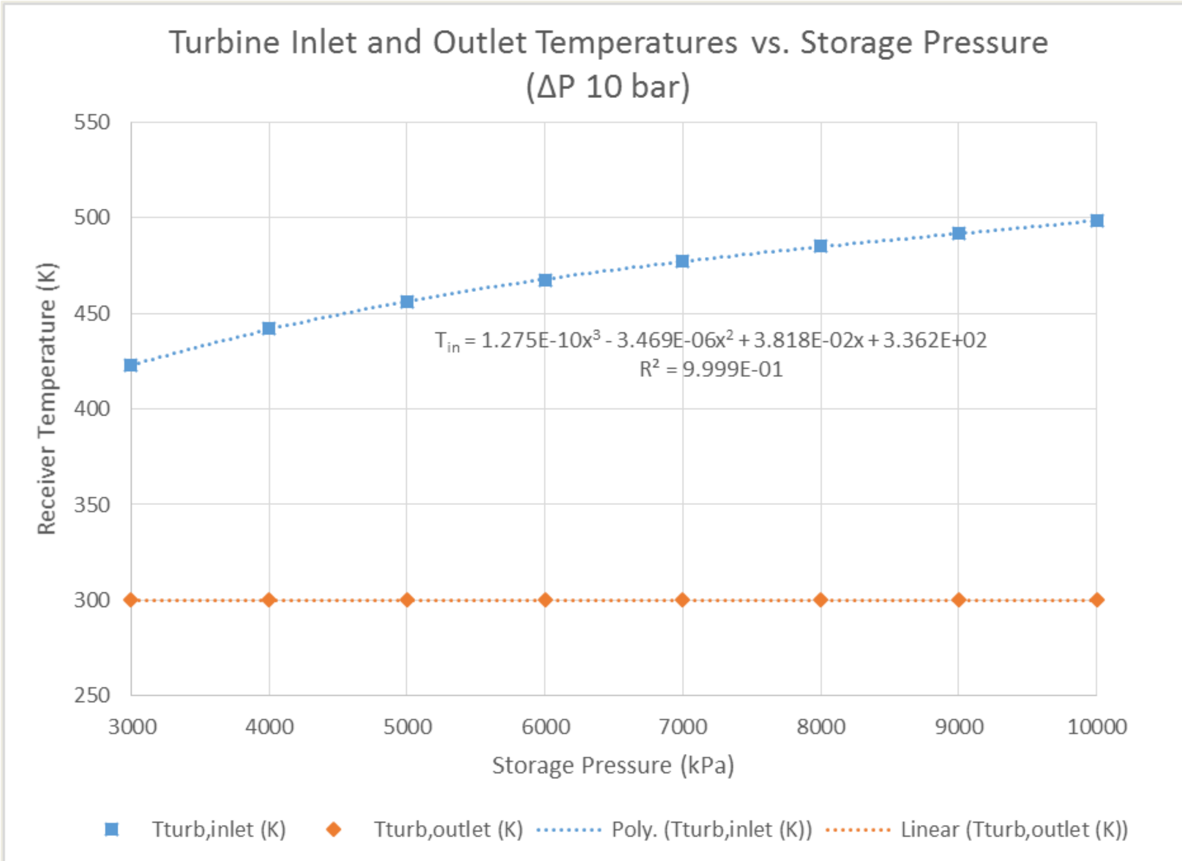


*Figure 49: Turbine Inlet and Outlet Temperatures vs. Storage Pressure (ΔP 10 bar)*

### 4.7 Turbine Mass Flow Rate

The fifth system configuration set iterates the expansion system mass flow rate; this is treated as an independent variable not linked to the compression system from the sizing model. The range of values over which the system is iterated is 1.1295 *kg/s* to 20.1295 kg/*s* in nine steps, as listed in Table 6. These values are based on the compressor mass flow rate of 12.1925 *kg/s.* This did not change the other specified variables, but it did affect the amount of energy that could be stored. The receiver volumes were intentionally held constant to show the effects of the mass flow rate on the system.

*Table 6: Turbine Mass Flow Rate Variations*

| Run | $Stages_{turb,run}$ | $P_{storage}$ (kPa) | $P_{turb,in}$ (kPa) | $T_{dp,in}$ (K) | $T_{turb,out}$ (K) | $\dot{m}_{turb}$(kg/s) | V ($m^3$) |
|---|---|---|---|---|---|---|---|
| 31 | 3 | 2000 | 1000 | 295 | 300 | 1.1295 | 43300 |
| 32 | 3 | 2000 | 1000 | 295 | 300 | 4.1295 | 43300 |
| 33 | 3 | 2000 | 1000 | 295 | 300 | 7.1250 | 43300 |
| 34 | 3 | 2000 | 1000 | 295 | 300 | 9.1295 | 43300 |
| 35 | 3 | 2000 | 1000 | 295 | 300 | 11.1295 | 43300 |
| 36 | 3 | 2000 | 1000 | 295 | 300 | 12.1295 | 43300 |
| 37 | 3 | 2000 | 1000 | 295 | 300 | 13.1295 | 43300 |
| 38 | 3 | 2000 | 1000 | 295 | 300 | 15.1295 | 43300 |
| 39 | 3 | 2000 | 1000 | 295 | 300 | 20.1295 | 43300 |

The amount of energy stored was related to the amount of air the compression system was able to compress. There is a threshold rate, shown in Figure 50, between 4.1295 *kg/s* and 9.1295 *kg/s*. Below this rate, not all of the excess energy can be used to compress air. The compression system is simply too small, and though it can compress the mass to the correct pressure, it cannot compress enough mass. This results in unused excess energy.

As can be seen in Figure 51, the total reheat energy does not change above this apparent threshold either, but the reheater size does. More power, with the same amount of energy over the course of the year, is needed as the mass flow rate increases. Although the overall reheat energy will not decrease, finding the threshold mass flow rate will reduce the size of the system, possibly increasing input return.

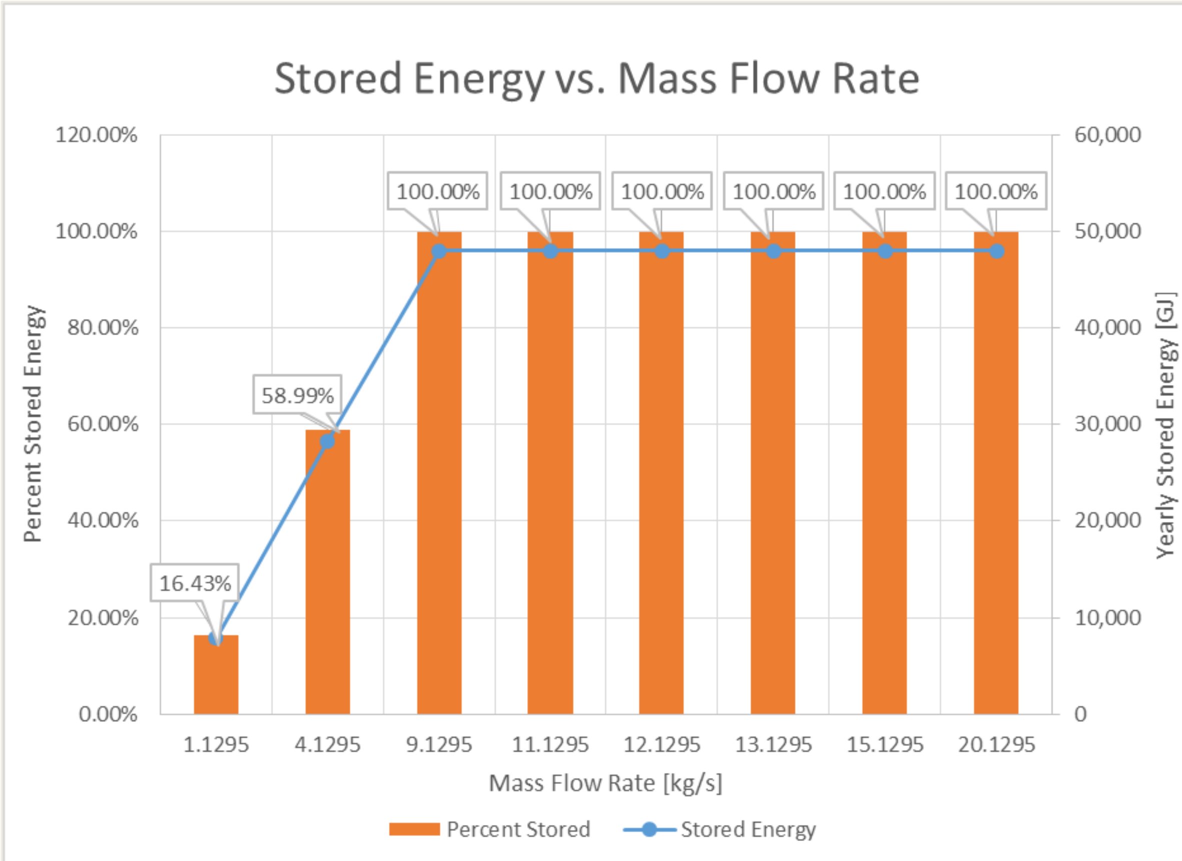


*Figure 50: Stored Energy vs. Mass Flow Rate*

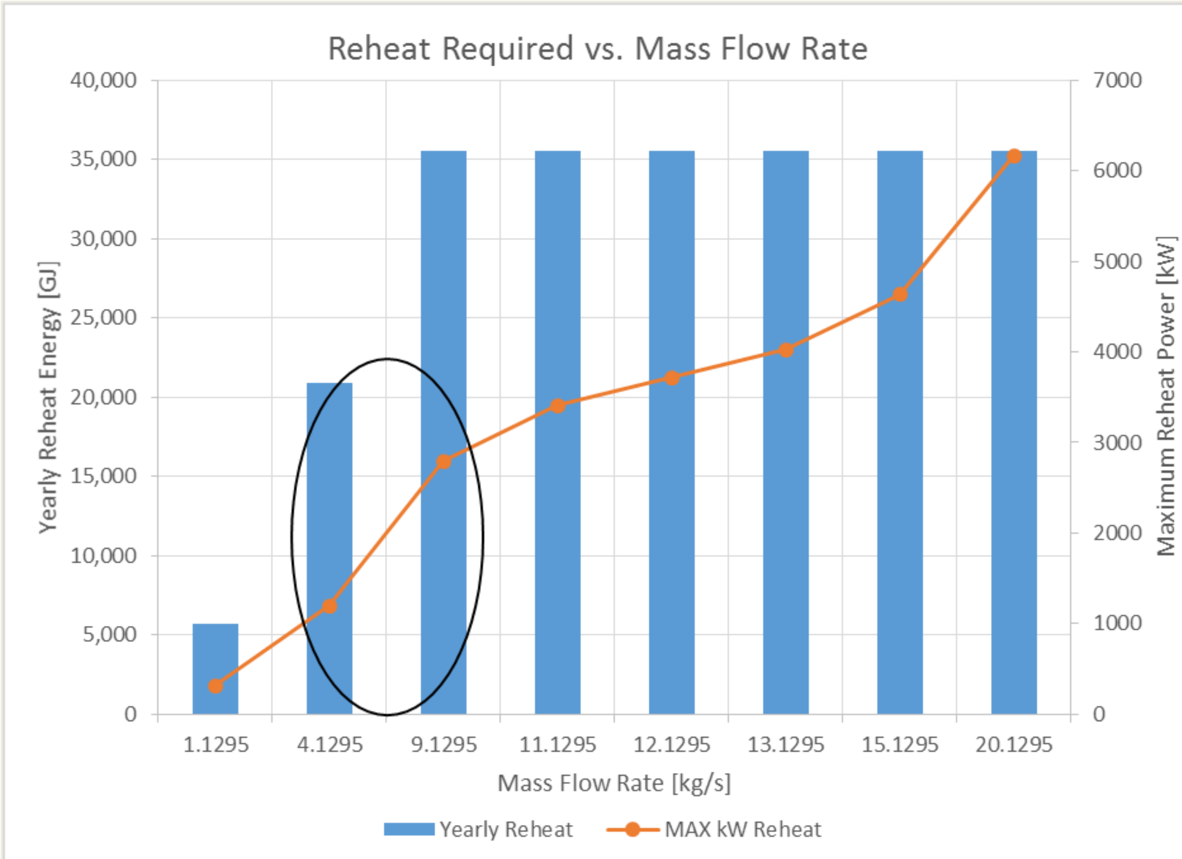


*Figure 51: Reheat Required vs. Mass Flow Rate*

After the mass flow rate was iterated over a step size of 0.0005 kg/s between 4.1295 kg/s and 9.1295 kg/s, it was determined that the threshold mass flow rate was 7.1250 kg/s for this system, shown in Figure 52. This corresponds to a reheater size of just below 2200 kW.

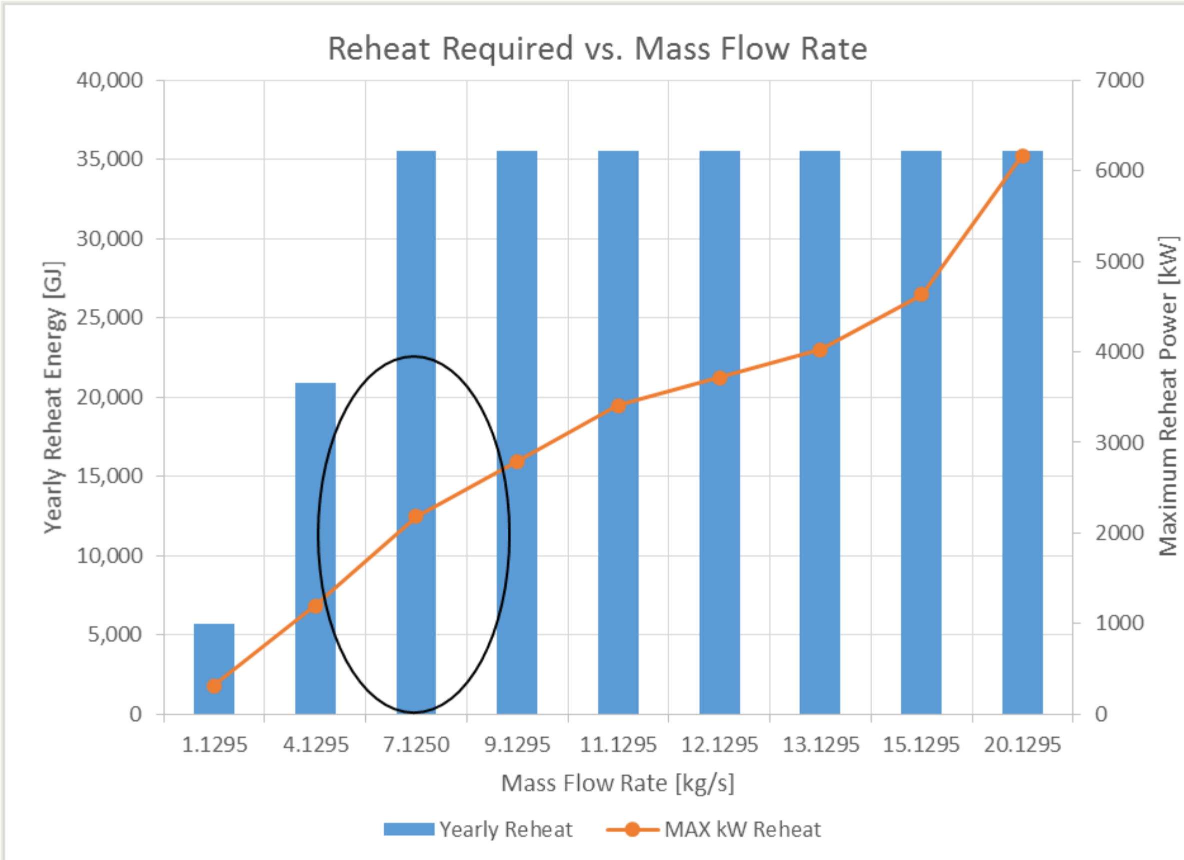


*Figure 52: Reheat Required vs. Mass Flow Rate (7.125 kg/s)*

All of the energy can be stored in the receiver, as is shown in Figure 53. This fact may be important in expansion system optimization, as it will drive the overall size of the expander powertrain.

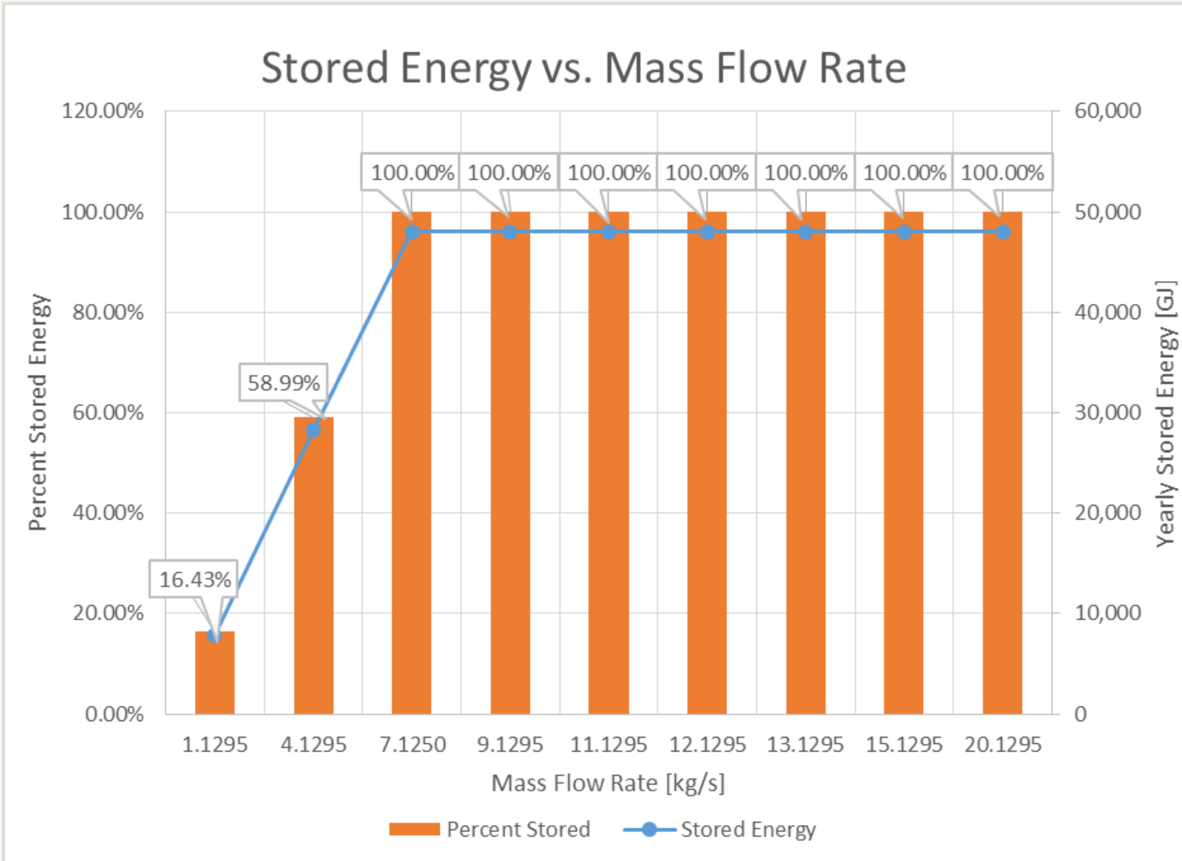


*Figure 53: Stored Energy vs. Mass Flow Rate (7.125 kg/s)*

### 4.8 Inlet Dew Point Temperature

The final factor that was considered was the dew point temperature of the compressor inlet air. This temperature was iterated over the range 260 $K$ to 300 $K$ in 10 $K$ increments, as listed in Table 7. The lower boundary of 260 $K$ was chosen because it corresponds to a very low absolute humidity ratio that, though accounted for with the data tables, is unlikely to be encountered. At this dew point temperature and atmospheric pressure, the air has almost no humidity. The upper boundary of 300 $K$ was chosen because if the dew point matches the dry bulb air temperature, the air is fully saturated with water. At this point, the maximum amount of water would be removed. Since the dew point temperature will only affect the compression system, assuming all other factors being the same, this configuration set iteration is a test of the humidity removal subroutine.

*Table 7: Dew Point Temperature Variations*

| Run | $Stages_{turb,run}$ | $P_{storage}$ (kPa) | $P_{turb,in}$ (kPa) | $T_{dp,in}$ (K) | $T_{turb,out}$ (K) | $\dot{m}_{turb}$(kg/s) | V ($m^3$) |
|---|---|---|---|---|---|---|---|
| 40 | 3 | 2000 | 1000 | 260 | 300 | 12.129 | 43300 |
| 41 | 3 | 2000 | 1000 | 270 | 300 | 12.129 | 43300 |
| 42 | 3 | 2000 | 1000 | 280 | 300 | 12.129 | 43300 |
| 43 | 3 | 2000 | 1000 | 290 | 300 | 12.129 | 43300 |
| 44 | 3 | 2000 | 1000 | 295 | 300 | 12.129 | 43300 |
| 45 | 3 | 2000 | 1000 | 300 | 300 | 12.129 | 43300 |

The results are shown in Figure 54, and are in line with what is expected. At 260 K, a negligible amount of water is removed per kg of air. This trend increases over the dew point temperature range. The condensate is removed during the compression and cooling processes, and if the compressed air at 300 K is cooled further, more water will condense out. Water is an important factor to consider, because in addition to the dangers of liquid water in the expansion system, there are corrosion concerns. This holds true for any compressed air system. If the quality of the air to be expanded is of the utmost concern (i.e. it will be used not only for reexpansion but for another process as well), further dehumidification of the compressed air may be necessary. This can be accomplished through further cooling or the use of desiccants.

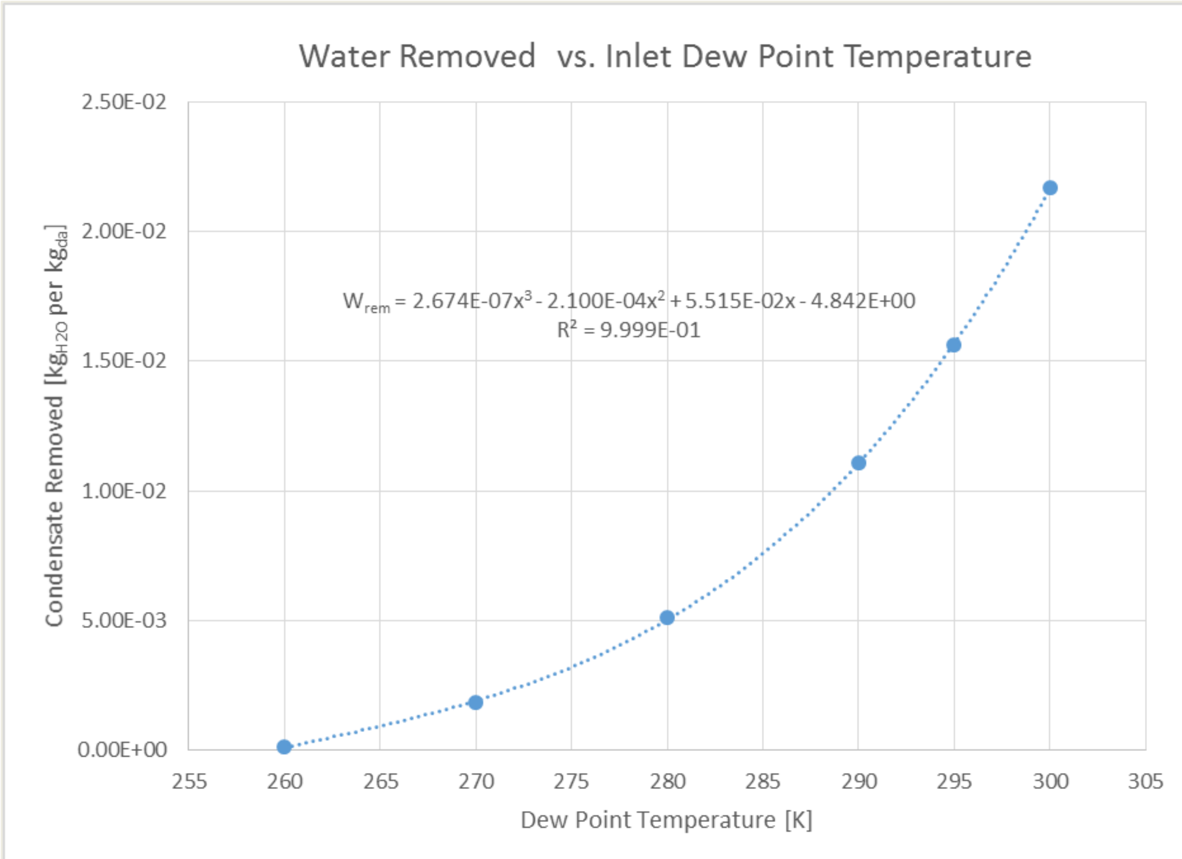


*Figure 54: Water Removed vs. Inlet Dew Point Temperature*

### 4.9 Final Model Plan

Now that the behavior of the simulated system has been observed and noted under various changes, an optimized model must be run. As expansion system output requirements rise, so do the requirements of the reheat system. The heat input translates to work output, but for most systems this is at a great loss. Most of the systems run, with the exception of the minimum reheat system with four or more stages, under a net deficit of energy. Not only do these systems not return any part of the excess energy stored, they require more energy in the form of heat to get anything out. The mass flow rate into the turbine has a threshold limit, and it appears that lower mass flow rates correspond to lower reheat values. Increased storage pressures also seem to correspond to better input return, but this is more likely a function of the lower mass flow rate.

With all of these variables and relevant behaviors, a model with the lowest possible reheat and mass flow will be simulated and iterated over multiple stages. This will determine the most effective system under this model structure. The simulation and its results are covered in the next chapter.

# CHAPTER 5

## Final Simulation, Discussion and Conclusions

The final simulation is intended to optimize the compressed air energy storage system, foremost as a mechanism to store and recuperate electrical energy. It may make more economic sense to have a system that uses magnitudes-greater amounts of reheat energy to maximize the production of work energy, such as through natural gas combustion. This will most definitely change the design parameters of the system. Certain variable values were kept constant, such as compressor inlet and outlet pressures and temperatures, intercooling temperature, turbine inlet pressure, and compressor inlet dew point temperature. Some of these variables are going to be environmentally-driven, and some will be driven by constraints of mechanical design, practicality or economics. Though it may make mathematic sense to compress the air to 100 bar or to have 15 expansion stages, economics and availability will guide the decision-making process. At the same time, the final simulation is based on the current model and the analysis performed; to remain realistic, this is not the end of the story. The model can undergo refinements, additions, and further analyses; it can be validated, implemented as a physical model, then studied some more. This is what has been determined here with the information available.

### 5.1 Final Simulation

The final system has been optimized for minimum reheat by manipulating the turbine outlet temperature, the number of expansion system stages and the mass flow rate through the expansion system. The variable values for the compression system and receiver are listed below in Table 8. These are, for the most part, the same values used in the analysis section.

*Table 8: Compressor Variable Values*

| $Stages_{comp}$ | $T_{comp,in}$ (K) | $T_{intercool}$ (K) | $T_{dp,in}$ (K) | $P_{comp,in}$ (kPa) | $P_{storage}$ (kPa) | V ($m^3$) |
|---|---|---|---|---|---|---|
| 3 | 300 | 300 | 295 | 101.325 | 2000 | 43270 |

The values for the expansion system variables are listed below in Table 9. The major points to note about the various ways the system was simulated are the number of stages and the mass flow rate. The number of stages varies from one to six, and the mass flow rate is set at the previously-determined minimum, 7.125 *kg/s*.

*Table 9: Expansion System Variable Values*

| Run | $Stages_{turb,run}$ | $T_{turb,out}$ (K) | $P_{turb,in}$ (kPa) | $P_{turb,outlet}$ (kPa) | $\dot{m}_{turb}$(kg/s) |
|---|---|---|---|---|---|
| 1 | 1 | 258.222 | 1000 | 101.325 | 7.125 |
| 2 | 2 | 258.222 | 1000 | 101.325 | 7.125 |
| 3 | 3 | 258.222 | 1000 | 101.325 | 7.125 |
| 4 | 4 | 258.222 | 1000 | 101.325 | 7.125 |
| 5 | 5 | 258.222 | 1000 | 101.325 | 7.125 |
| 6 | 6 | 258.222 | 1000 | 101.325 | 7.125 |

Figure 55 shows the effects of increasing the turbine/heat exchanger stage number on the temperature gradient over the turbine. By adding a second stage, the turbine inlet temperature falls by 171 *K*, and adding a third reduces the temperature another 46 *K*. Though the reheat temperature is being reduced, the number of times the air must be reheated increases. This fact may become important when determining the available reheat sources and their magnitudes: fuels will likely be capable of higher reheat temperatures than industrial waste heat or thermal storage.

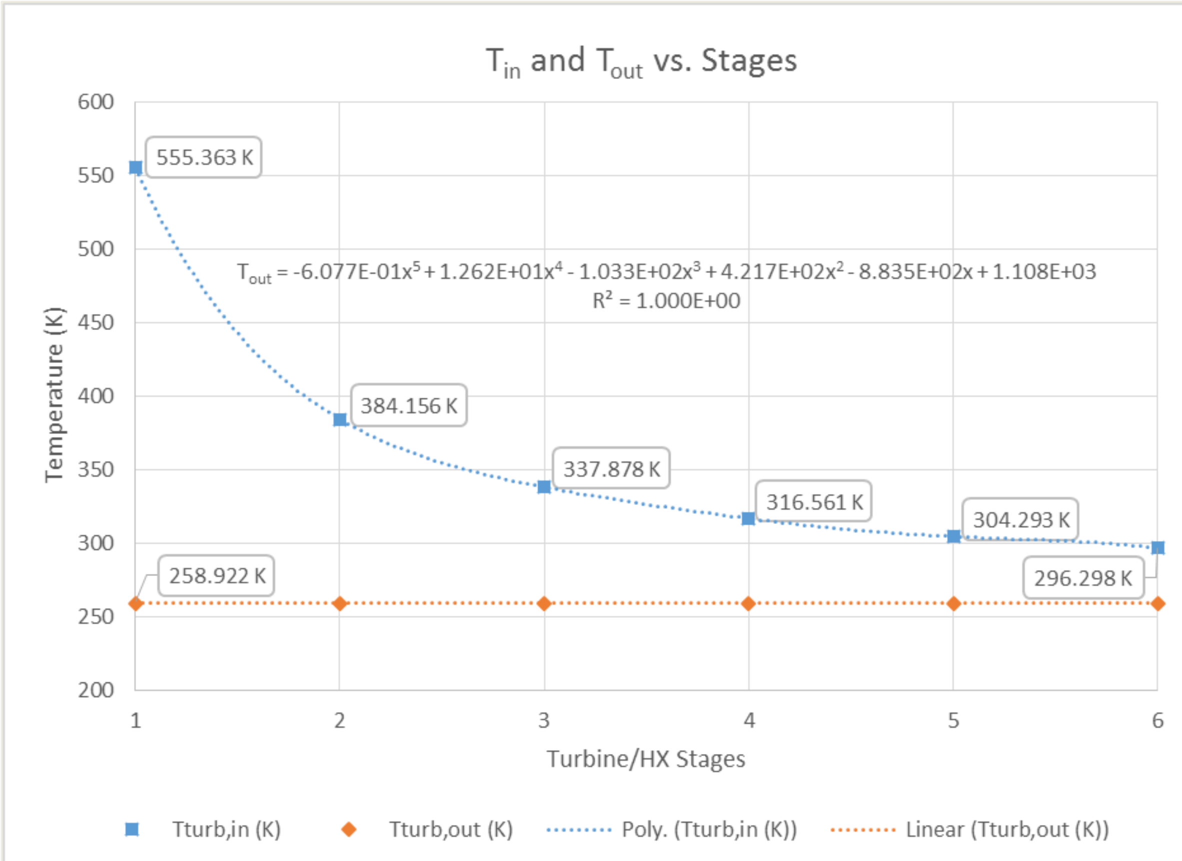


*Figure 55: $T_{in}$ and $T_{out}$ vs. Stages ($T_{out}$=$T_{dp}$)*

Figure 56 shows the power output of the expansion system, which is a static figure, and the necessary size of the reheat system, all as a function of stage number. The size of the reheat system is the maximum rate at which the inlet air at all turbines will need to be reheated; it does not indicate a static reheat power. The reheat is dependent, in part, on the temperature of the air in the receiver. For a single stage system, the reheat system must be sized for the minimum temperature of the air in the receiver. For multiple stages, the reheat system is a sum of its parts. Each turbine has a reheater/heat exchanger: the first reheater, depending on the temperature of the air leaving the receiver, will need to supply more or less energy than the subsequent units. If the air in the receiver is hotter than the required inlet temperature, no reheat will be provided to the first stage. If the air is colder, reheat will need to be supplied. Each subsequent stage will require steady power, but the first stage is variable. Therefore, the maximum size needed for reheat is based on the minimum temperature of the air in the receiver.

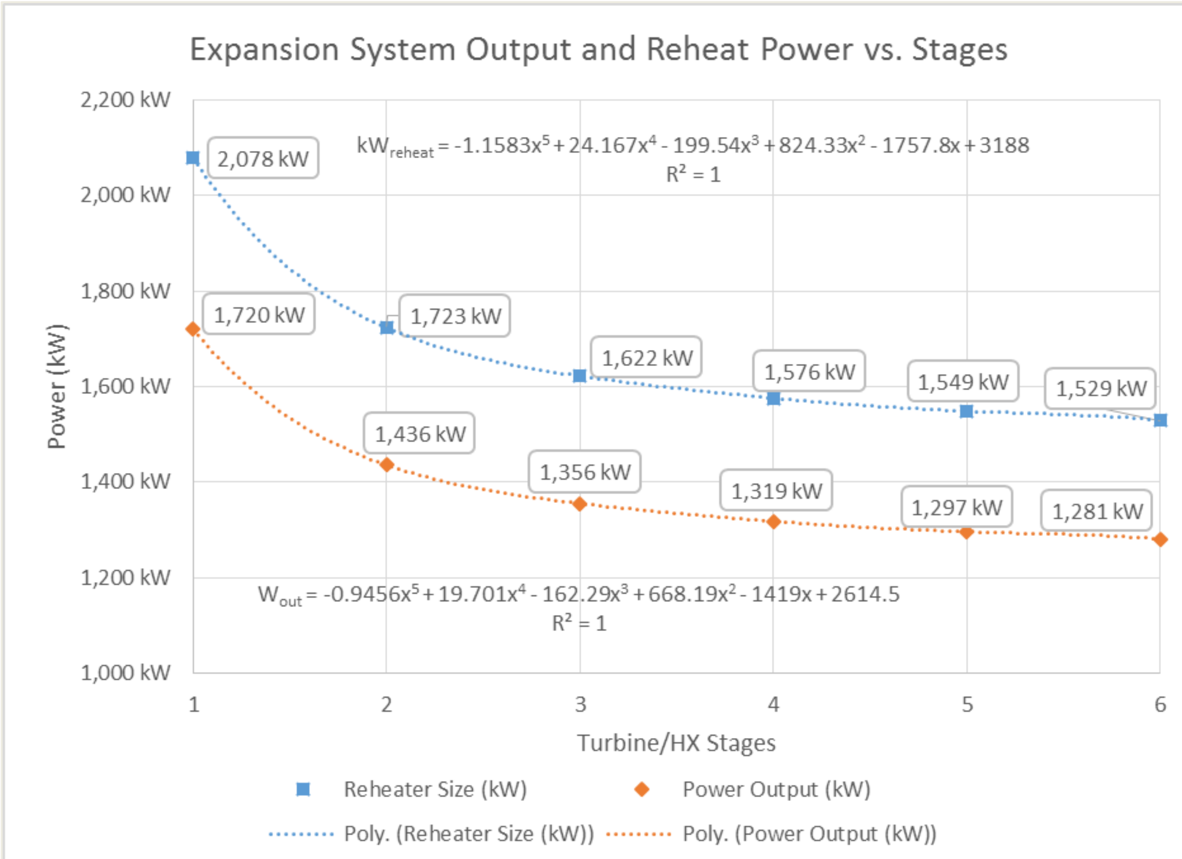


*Figure 56: Expansion System Output and Reheat Power vs. Stages*

The expansion system output, like the reheater system size, decreases with each additional stage. The mathematic difference between each of the two is not an accurate measure of performance, as one power is static and the other variable. Depending on the receiver air temperature, the reheat power may vary between a minimum number, which would be the sum of the power required to reheat the air entering the stages after the first, and the maximum size listed.

The final story is told in Figure 57, in which the overall system efficiency and the energy input return are displayed as functions of stage number. The overall efficiency decreases approximately 5.5% with the addition of 5 more stages, but the energy input return increase is just shy of 4.5%. The return never reaches or exceeds 0%, which indicates that, at least in the current configuration, the system requires more reheat energy than it produces.

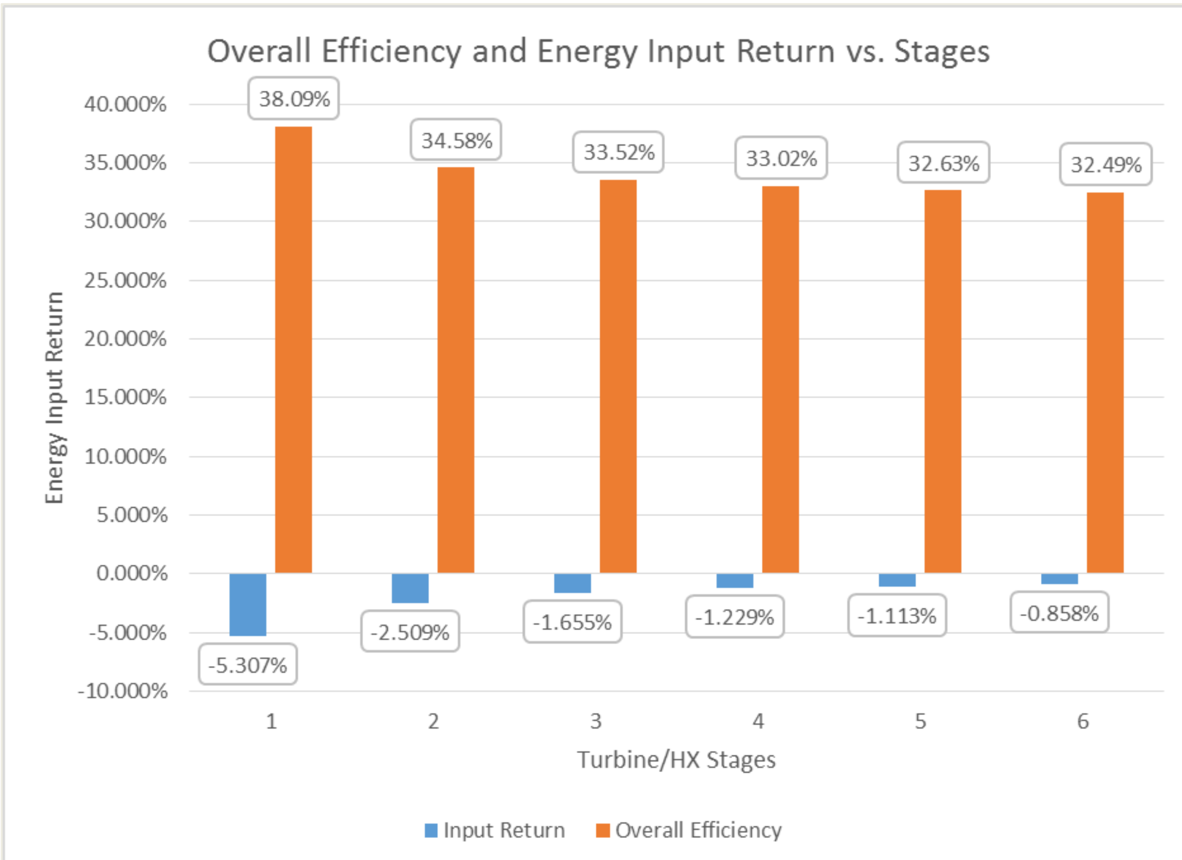


*Figure 57: Overall Efficiency and Energy Input Return vs. Stages*

The losses to overall system efficiency are greater, on a percent basis, than the gains to input return. Again, this will have to be considered, given available reheat energy sources. Tradeoffs in energy production may have to be made for lower-quality heat.

### 5.2 Discussion

Given the final model, the outlook seems grim for energy storage with this system. Even the six-stage expansion system, provided as an outlier example, cannot provide a viable storage system. This is, however, a matter of accounting. If merely half of the energy needed for reheat were able to be stored from the compression system and delivered to the expansion system, the entire picture changes. The results of halving the supplemental reheat are shown in Figure 58.

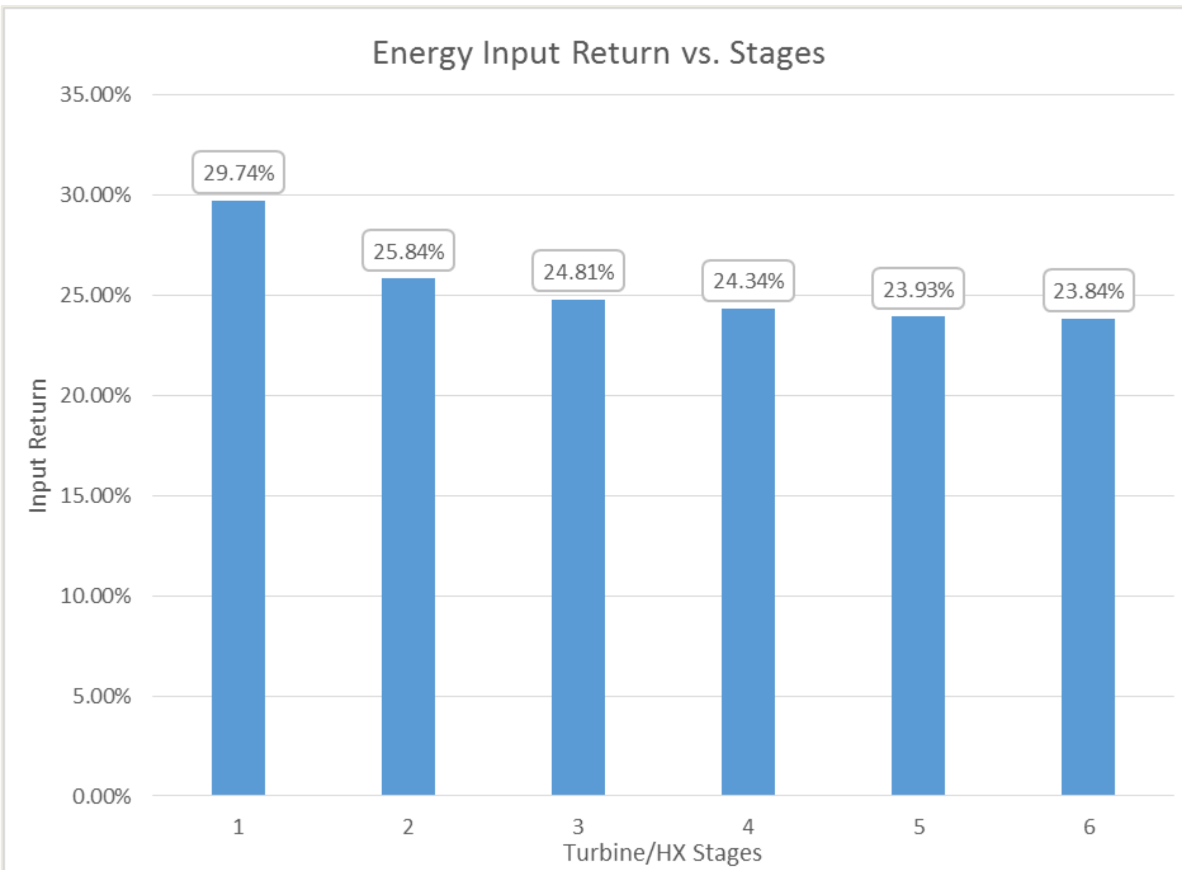


*Figure 58: Energy Input Return vs. Stages*

This is much better news than sub-zero returns. This change would increase the overall system efficiency as well, because some of the reheat would be supplied by rejected compression heat. This would reduce the amount of reheat added to the total input energy term in the denominator of equation (77) while keeping the numerator the same. As stated earlier, the available reheat may drive the number of expansion stages.

In addition to thermal storage, reheat energy might also be supplied by waste heat from either an industrial process or by the heat rejected through the condenser(s) of an electricity generation plant. Further, it remains to be seen what happens if the air leaving the compressor is not aftercooled. This effect was not modeled, and presumably this will require more storage. Not aftercooling should also reduce the amount of dehumidification, raising the dew point temperature of compressed air. The cost of a larger receiver may be worth any savings gained by reheat reduction. Depending on the magnitude of available energy, the outlet conditions may be able to be increased as well. This should increase the expansion system output, and may take care of any possible problems relating to water.

The outlined system is meant to serve as a boundary. Even if the necessary reheat is free and available, having a practical system that operates at the cusp of condensation, at temperatures 14° C below the freezing point of water, is very risky. It would be advisable to keep the air temperature a safe distance above these points throughout the storage and expansion process. If the output air has sufficient cooling capacity (amount, flow and temperature), it may also be used for process or condenser cooling.

Finally, assuming the system can be built so that all of the reheat can be supplied through thermal storage and "free" waste heat as shown in Figure 59, we still run into the issue of the receiver. The receiver in question is 43,270 m$^3$, which is 1.528 million cubic feet or 11.431 million gallons. This volume could be split up into multiple tanks, but the issue of space remains. The real estate is still required to keep a tank that is just under the magnitude of 17 ½ Olympic-sized swimming pools. This would look like a rectangular tank that is 50 m long, 25 m wide and 35 m high (164 ft x 82 ft x 115 ft). Given the potential returns, this may or may not be a proposition worth considering.

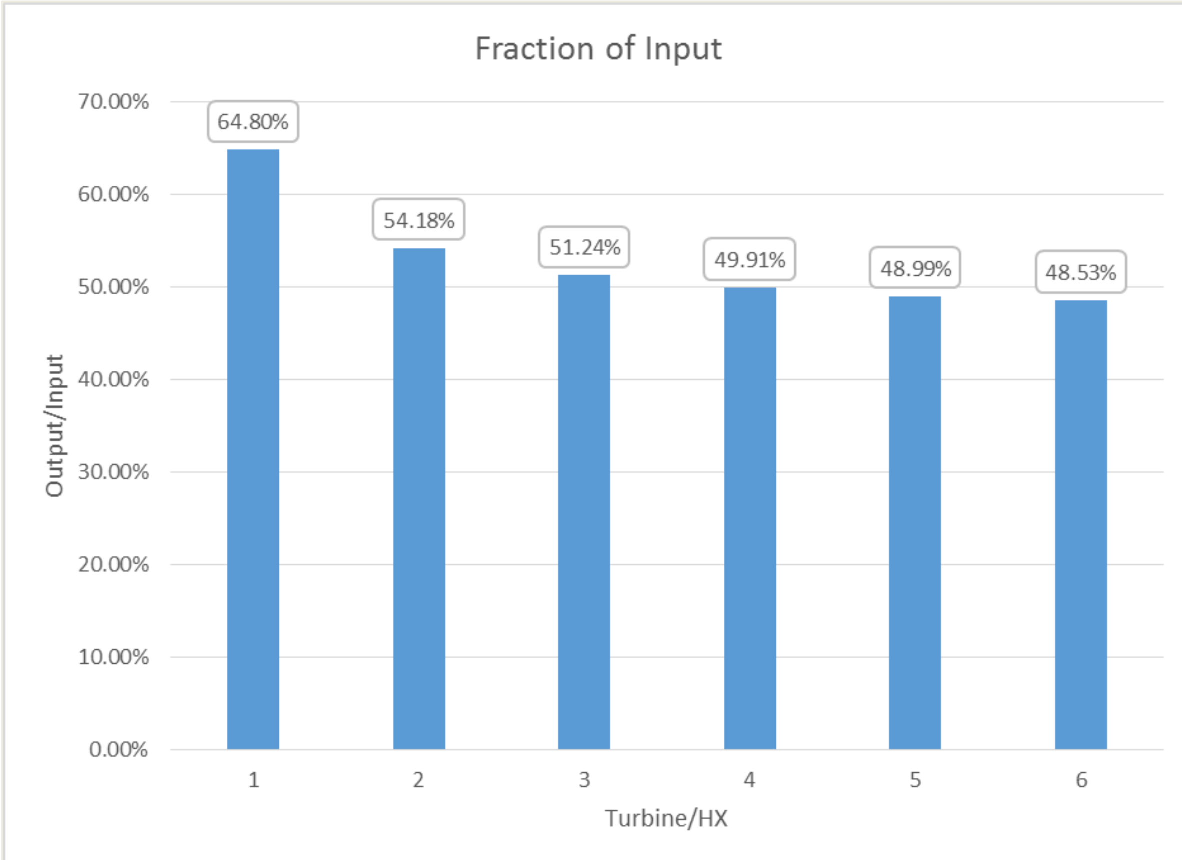


*Figure 59: Fraction of Input (Neglecting Reheat Energy)*

### 5.3 Conclusions

In the end, compressed air energy storage seems to have marginal returns at best. Depending on the availability of secondary reheat energy, the system (as simulated) may return just under 65% of the input energy. This return requires a (single stage) turbine inlet temperature of 555 *K*; the temperature for two stages requires a temperature that is 11°C above the boiling point of water. Only at three stages does the magnitude of reheat temperature (65°C) become reasonable. This is for a model that does not include losses other than compressor heat rejection and the isentropic compressor and turbine efficiencies. In fact, the upper level of any storage system's output return should be the input energy – process losses; this is the basis of the system efficiency. With the simulated systems, since the input energy is largely rejected as heat from the compression process before the air is stored, the return is inevitably low. The enthalpy of the compressed air is entirely temperature-dependent, and the compressor outlet air is cooled to the input temperature condition. This heat removal is necessary to minimize storage vessel size and compression work, but may be counterproductive for a storage facility. Compounding the effects of the heat rejection is the fact that the air must be reheated in order to prevent condensation in the expansion system. If the air was allowed to expand unheated through the expansion system, even given completely dry air, the system would still produce energy. There is a pressure gradient, and there would still be mass flow and an enthalpy change across each turbine. The expansion system output temperature would likely be very low, and power output would be low. This deserves looking into, but given the amount of space and machinery required for a CAES system, another more energy dense form of storage may be warranted.

To summarize, any energy storage system will have space limitations, but the Achilles Heel of compressed air storage is that so much energy is removed during compression. It is removed to minimize the input work of compression as well as to minimize the storage needed. This energy is rejected but has to be added back in order to extract useful work at any reasonable magnitude. It is in this way that energy storage with compressed air is almost entirely dependent on the way its heat is stored. If compression heat is not stored, reheat energy is not freely available, or the compressed air is not used in some critical process (i.e. combustion turbine), compressed air storage is not a worthwhile technology for energy storage.

### 5.4 Future Work

These conclusions are the result of the specific model used for simulation and analysis performed. The set of variables subject to manipulation in the model may be incomplete, and some major variable or quantity may have been overlooked. Further study is the only course through which more answers will be found. The reheat system is the source of many questions. It is only through the study and design of an effective thermal storage or waste heat delivery system that compressed air energy storage will start to become worthwhile. In addition, the effects of not aftercooling the compressed air should be studied. This model should also be run to simulate a real world system, such as those in McIntosh, AL or Huntorf, Germany. The model would have to be augmented to include fuel as a reheat source and the pressure effects of combustion, at a minimum. Finally, the model should be augmented and run to determine how much power can be recovered given unheated expansion of completely dried air.

# REFERENCES

[1] “Pacific Gas & Electric - Tariffs.” [Online]. Available: http://www.pge.com/nots/rates/2000_static.shtml#topic1. [Accessed: 10-Dec-2014].

[2] L. Grigsby, Ed., *The Electric Power Engineering Handbook*. CRC Press, 2000.

[3] “Hydropower Facts, Hydropower Information- National Geographic.” [Online]. Available: http://environment.nationalgeographic.com/environment/global-warming/hydropower-profile/. [Accessed: 10-Dec-2014].

[4] “Getting salmon past daunting Willamette Basin dams could have a big price tag -- and a big payoff | OregonLive.com.” [Online]. Available: http://www.oregonlive.com/environment/index.ssf/2011/09/detroit_dam_to_restore_wild_ru.html. [Accessed: 10-Dec-2014].

[5] “What is U.S. electricity generation by energy source? - FAQ - U.S. Energy Information Administration (EIA).” [Online]. Available: http://www.eia.gov/tools/faqs/faq.cfm?id=427&t=3. [Accessed: 11-Dec-2014].

[6] D. R. Biggar and M. R. Hesamzadeh, Eds., “Efficient Short-Term Operation of an Electricity Industry with no Network Constraints,” in *The Economics of Electricity Markets*, John Wiley & Sons Ltd, 2014, pp. 93–118.

[7] Y. A. Çengel and M. A. Boles, *Thermodynamics: an engineering approach*. Boston: McGraw-Hill Higher Education, 2008.

[8] “Power From the Sun :: Chapter 12.” [Online]. Available: http://www.powerfromthesun.net/Book/chapter12/chapter12.html. [Accessed: 11-Dec-2014].

[9] G. Chen, *Nanoscale energy transport and conversion: a parallel treatment of electrons, molecules, phonons, and photons*. Oxford ; New York: Oxford University Press, 2005.

[10] “Coal | Clean Energy | US EPA.” [Online]. Available: http://www.epa.gov/cleanenergy/energy-and-you/affect/coal.html. [Accessed: 10-Dec-2014].

[11] “IEA - Access to Electricity.” [Online]. Available: http://www.worldenergyoutlook.org/resources/energydevelopment/accesstoelectricity/. [Accessed: 10-Dec-2014].

[12] D. R. Biggar and M. R. Hesamzadeh, Eds., “Achieving Efficient Use of Generation and Load Resources using a Market Mechanism in an Industry with no Network Constraints,” in *The Economics of Electricity Markets*, John Wiley & Sons Ltd, 2014, pp. 119–138.

[13] J. Casazza and F. Delea, *Understanding electric power systems an overview of the technology, the marketplace, and government regulation*. Piscataway, NJ; Hoboken, N.J.: IEEE Press ; Wiley, 2010.

[14] “AEP.com - Smith Mountain Dam.” [Online]. Available: http://www.aep.com/environment/conservation/hydro/smithmtn.aspx. [Accessed: 10-Dec-2014].

[15] “Pump Storage - The Hydro Equipment Association (HEA).” [Online]. Available: http://www.thehea.org/hydropower/special-focus/pump-storage-power-plants/. [Accessed: 11-Dec-2014].

[16] P. Breeze, “10 - Storage technologies,” in *Power Generation Technologies*, P. Breeze, Ed. Oxford: Newnes, 2005, pp. 134–152.

[17] A. Ter-Gazarian and Institution of Engineering and Technology, *Energy storage for power systems*, 2nd ed. London: Institution of Engineering and Technology, 2011.

[18] M. R. Patel, *Shipboard propulsion, power electronics, and ocean energy*. Boca Raton, FL: Taylor & Francis, 2012.

[19] L. K. Wang, *Handbook of Advanced Industrial and Hazardous Wastes Treatment*. Boca Raton, FL, USA: CRC Press, 2009.

[20] “Compressed Air History.” [Online]. Available: http://www.ecompressedair.com/library-pages/compressed-air-history.aspx. [Accessed: 10-Dec-2014].

[21] “DOE Global Energy Storage Database.” [Online]. Available: http://www.energystorageexchange.org/projects. [Accessed: 11-Dec-2014].

[22] D. L. V. Katz and E. R. Lady, *Compressed air storage for electric power generation*. Ann Arbor, Mich.: [s.n.], 1976.

[23] T. Giampaolo, *Compressor handbook: principles and practice*. Lilburn, GA : Boca Raton, FL: Fairmont Press ; Distributed by Taylor & Francis, 2010.

[24] “Types of Air Compressors.” [Online]. Available: http://constructionmanuals.tpub.com/14050/css/14050_196.htm. [Accessed: 11-Dec-2014].

[25] “Screw Compressor.” [Online]. Available: http://www.jagweb.com/aj6eng/supercharging_article.html. [Accessed: 11-Dec-2014].

[26] “STC-SX - Siemens.” [Online]. Available: http://www.energy.siemens.com/mx/en/compression-expansion/product-lines/single-shaft-horizontal-split/stc-sx.htm. [Accessed: 11-Dec-2014].

[27] “Site | GE Oil & Gas.” [Online]. Available: http://www.geoilandgas.com/search/site/roots%20api?f%5B0%5D=bundle%3Aproduct. [Accessed: 10-Dec-2014].

[28] “OPRA Turbines | Kinsley Group.” [Online]. Available: http://www.kinsley-group.com/Opra-turbines. [Accessed: 11-Dec-2014].

[29] “Generator Set CX400 (12.9 MWe) | Centrax Gas Turbines.” [Online]. Available: http://www.centraxgt.com/products/generator-set-cx400-129-mwe. [Accessed: 11-Dec-2014].

[30] M. Schobeiri, *Turbomachinery flow physics and dynamic performance*, 2nd & enhanced ed. New York: Springer, 2012.

[31] M. Raju and S. K. Khaitan, “Modeling and simulation of compressed air storage in caverns: A case study of the Huntorf plant,” *Appl. Energy*, vol. 89, no. 1, pp. 474–481, 2012.

[32] R. H. Howell, *Principles of heating ventilating and air conditioning: a textbook with design data based on the 2013 ASHRAE handbook fundamentals*, 7th edition. Atlanta, GA: ASHRAE, 2013.

[33] G. E. P. Box, *Empirical model-building and response surfaces*. New York: Wiley, 1987.

[34] “NIST Thermodynamic Properties of Air.” [Online]. Available: http://www.nist.gov/data/PDFfiles/jpcrd581.pdf. [Accessed: 11-Dec-2014].

# APPENDICES

# Appendix A

## NIST Constant Pressure Specific Heat of Air Equations and Coefficients[33]

$$c_p = R_{gas}\sum_{i=1}^{4} N_i T^{i-1} + \frac{N_5}{T^{1.5}} + \frac{N_6 u^2 e^u}{\left(e^u - 1\right)^2} + \frac{N_7 v^2 e^v}{\left(e^v - 1\right)^2} + \frac{\left(2/3\right) N_8 w^2 e^{-w}}{\left(\left(2/3\right) e^{-w} + 1\right)^2}$$

$$u = \frac{N_9}{T} \ , v = \frac{N_{10}}{T} \ , \ w = \frac{N_{11}}{T}$$

N coefficient values:

1) 3.490 888 032
2) $2.395\ 525\ 583 \times 10^{-6}$
3) $7.172\ 111\ 248 \times 10^{-9}$
4) $-3.115\ 413\ 101 \times 10^{-13}$
5) 0.223 806 688
6) 0.791 309 509
7) 0.212 236 768
8) 0.197 938 904
9) 3364.011
10) 2242.45
11) 11 580.4

# Appendix B

## Fortran Code Samples (All Rights Reserved)

```
!------------------------------------------------------------------------------------------------------------------------------------------------------------------------

PROGRAM outlet_300K

IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-------------------------------VARIABLES---------------------------------------

REAL:: stages
REAL:: P_in,P_out,P_low,T_in,T_tank,eta_percent,eta_t_percent,T_wb
INTEGER :: i,j,n
CHARACTER(len=20) :: fname

!-----------------------------------------------------------------------------

stages=3
P_in=101.325

P_out=2000

P_low=1000

T_in=300
T_tank=300

eta_percent=80

fname='PeakDay.dat'
n=24

eta_t_percent=80

T_wb=295

CALL comp_work_check(stages, P_in, P_out, P_low, T_in, T_tank, eta_percent, fname, n, eta_t_percent, T_wb)

!-------------------------------------------------------------------------------

WRITE(*,*)'All done hon!'

END PROGRAM outlet_300K

!------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE comp_work_check(stages, P_in, P_out, P_low, T_in, T_tank, eta_percent, fname, n, eta_t_percent, T_wb)!, tstages)

!This program, through use of subroutines, calculates the work and power required to compress air,
!the volume required for the compressed air and excess power parameters given,
!and the change in enthaply / air reheat necessary to recapture the input power via expansion

!Subroutines include:

!specific_heat_calc (T_in, R_gas, cp, cv, k)
                                                                                                    CALC SPECIFIC HEATS & RATIO K
!spec_comp_work (stages, rc, P_in, P_out, T_in, T_out, cp, cv, eta_c, w_in, k)
                                                            CALC COMPRESSOR WORK
!enter_w_dot_xs (fname,n)

                                                                                                    If file does not exist, call entry subroutine
!volume_choice (V_m, V_g, P_low, P_out, T_tank, w_in, fname, n, del_m, W_dot_sum, R_gas, Comp_powmax)CALC TANK VOLUME
!tank_temp (T_in, T_tank, V_tank, T_out, P_low, P_out, R_gas, m_discharged, T_fill, T_empty, n) CALC AIR TEMP IN TANK
!expander (P_low, P_amb, T_amb, del_m, W_dot_sum, T_tank, k, a, b, c, d, e, R_gas, m_dot_xpd, kWreq)  CALC TURBINE WORK
OUTPUT, REHEAT

IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!--------------------------FILE OPERATION UNITS-------------------------------

!AIR DATA -- UNITS (9,13)
!results300K -- UNITS (8,10,12)
!EXCESS DATA FILE (7,11)
!MASS RATE IN (15,16,17)

!------------------------------VARIABLES--------------------------------------

!REAL :: a,b,c,d,e                                  !coefficients for air specific heat equation KEPT FOR EXPANDER
SUBROUTINE
REAL,INTENT(IN) :: stages
                              !number of stages [unitless] (HERE, SPEC_COMP_WORK)
REAL :: rc
                           !overall compression ratio [unitless] (HERE, SPEC_COMP_WORK)
REAL :: rp
                           !stage compression ratio [unitless] (HERE, SPEC_COMP_WORK)
REAL,INTENT(IN) :: P_in
                         !inlet air pressure (ambient pressure) [kPa] (HERE, SPEC_COMP_WORK)
REAL,INTENT(IN) :: P_out
     !final air pressure (max storage pressure)[kPa] (HERE, SPEC_COMP_WORK, VOLUME_CHOICE, TANK_TEMP)
REAL,INTENT(IN) :: P_low
       !working air pressure (low storage pressure) [kPa] (HERE, VOLUME_CHOICE, TANK_TEMP, EXPANDER)
REAL :: del_m
      !mass differential between max/high and base/low pressure storage states [kg] (VOLUME_CHOICE, EXPANDER)
REAL,INTENT(IN) :: T_in
       !inlet air temperature (per stage) [K] (HERE, SPECIFIC_HEAT_CALC, SPEC_COMP_WORK, TANK_TEMP)
REAL :: T_out
                 !outlet air temperature (per stage) [K] (HERE, SPEC_COMP_WORK, TANK_TEMP)
REAL,INTENT(IN) :: T_tank
              !steady-state receiver temperature [K] (HERE, VOLUME_CHOICE, TANK_TEMP, EXPANDER)
```

```
REAL :: cp
                    !constant pressure specific heat of air at given temperature and pressure [kJ/kg-K]
!(HERE, SPECIFIC_HEAT_CALC, SPEC_COMP_WORK)
REAL :: cv
                    !constant volume specific heat of air at given temperature and pressure [kJ/kg-K]
!(HERE, SPECIFIC_HEAT_CALC, SPEC_COMP_WORK)
REAL :: k
            !cp/cv specific heat ratio (HERE, SPECIFIC_HEAT_CALC, SPEC_COMP_WORK, EXPANDER)
REAL :: R_gas
                    !gas constant for air (SPECIFIC_HEAT_CALC, TANK_TEMP, EXPANDER)
REAL :: R_u
                                !gas constant universale (HERE)
REAL,INTENT(IN) :: eta_percent                  !isentropic compressor efficiency <percentage> [unitless] (HERE)
REAL :: eta_c
                    !isentropic compressor efficiency [unitless] (HERE, SPEC_COMP_WORK)
REAL :: w_in
            !specific work out for given parameters [kJ/kg] (HERE, SPEC_COMP_WORK, VOLUME_CHOICE)
REAL :: W_dot_sum                               !Sum of all excess power values (VOLUME_CHOICE, EXPANDER)
REAL :: V_m
                        !receiver volume in cubic meters (VOLUME_CHOICE)
REAL :: V_g
                        !receiver volume in gallons (VOLUME_CHOICE)
REAL :: Comp_powmax                             !max power rating [kW] (within 20%) of the compressor system
(VOLUME_CHOICE)
REAL :: m_dot_max                                       !Maximum cmpressor mass flow rate [kg/s]

CHARACTER(len=20),INTENT(IN) :: fname!data file with excess power data (HERE, ENTER_W_DOT_XS, VOLUME_CHOICE)
INTEGER :: ioerr                                                !for file operations (HERE)
LOGICAL :: xst
                                !for file operations (HERE)
INTEGER :: i,m
            !number of data points in data file (HERE, ENTER_W_DOT_XS, VOLUME_CHOICE, TANK_TEMP)
INTEGER, INTENT(IN) :: n

REAL :: T_amb
                    !Ambient temperature at turbine outlet, same as T_in (HERE, EXPANDER)
REAL :: P_amb
                    !Ambient pressure at turbine outlet, same as P_in (HERE, EXPANDER)
REAL :: m_dot_xpd                                   !Turbine design flow rate [kg/s] (HERE, EXPANDER)
!REAL :: kWreq
                    !Power output at given mass flow required to recup energy expended
REAL, INTENT(IN) :: eta_t_percent
!REAL, INTENT(IN) :: tstages

REAL, ALLOCATABLE :: Ttab(:)        !Table reference temperature [K] (HERE)
REAL, ALLOCATABLE :: htab(:)        !Table reference enthalpy [kJ/kg] (HERE)
REAL, ALLOCATABLE :: Prtab(:)       !Table reference refpressure [kPa] (HERE)
REAL, ALLOCATABLE :: utab(:)        !Table reference internal energy [kJ/kg] (HERE)
REAL, ALLOCATABLE :: vrtab(:)       !Table reference refspecific volume [m3/kg] (HERE)
REAL, ALLOCATABLE :: sotab(:)   !Table reference zero-pressure entropy [kJ/kg-K] (HERE)

REAL :: h_comp_hot                              !specific enthalpy available per intercooled/aftercooled stage [kJ/kg]
(HERE)
REAL :: comp_int_quot_hot       !interpolation quotient for compression properties (HERE)
REAL :: h_comp_cold                             !specific enthalpy available per intercooled/aftercooled stage [kJ/kg]
(HERE)
REAL :: comp_int_quot_cold      !interpolation quotient for compression properties (HERE)

REAL,INTENT(IN) :: T_wb
            !Dewpoint temperature of the compressor inlet air --> Initially conflated wet bulb and dewpoint temps
REAL :: T_min
                    !Minimum temperature the turbine outlet air can get before condensation occurs
REAL :: W_rem
                        !Fraction of water removed per kg of air compressed
```

```
REAL :: V_tank_gal                                                  !Receiver volume [GALLONS]specified by user (TANK_TEMP)
REAL :: V_tank
                                        !Receiver volume [m**3] specified by user (HERE, TANK_TEMP)
REAL :: m_discharged                                                               !Base fill mass at T_tank [kg] (TANK_TEMP)
REAL :: T_fill
                                       !Temperature [K] after a period of filling the receiver (TANK_TEMP)
REAL :: T_empty
                                      !Temperature [K] after a period of emptying the receiver (TANK_TEMP)

!--------------------------READ IN AIR PROPERTY DATA----------------------------

OPEN(UNIT=9,FILE='air_data_table_T_h_Pr_u_vr_so.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)!Open & Read INPUT
FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

105 FORMAT(8X,F8.0,4X,F9.2,4X,F10.4,4X,F9.3,4X,F10.3,4X,F12.5)

m=121

ALLOCATE(Ttab(m+1),htab(m+1),Prtab(m+1),utab(m+1),vrtab(m+1),sotab(m+1))

DO i=1,m

READ(9,*)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)

END DO

WRITE(*,*)''
WRITE(*,*)''
WRITE(*,*)'                TABLE OF IDEAL GAS PROPERTIES FOR AIR (WARK)'
WRITE(*,*)''
WRITE(*,*)'           TEMP [K]    h [kJ/kg]     Pr     u [kJ/kg]     vr     so [kJ/kg-K]'
WRITE(*,*)'    ========    =========    ==========    =========    ==========    ============'
DO i=1,m

WRITE(*,105)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)
END DO
WRITE(*,*)''

CLOSE(9)

!--------------------------USER INPUTS VARIABLES--------------------------------

R_u=8.3144621
R_gas=R_u/28.97


                                                        !(Ru/Mair)

100 FORMAT(A,2X,F10.3,1X,A,2X,F10.3,1X,A,2X,F10.3,1X,A)

WRITE(*,'(A)')'  This program will calculate the isentropic compression work required for staged air compression,',&
&'                    the minimum storage receiver volume required given a data set of excess power values,',&
&'          and the reheat necessary to increase the enthalpy of the air in order to have ambient turbine outlet conditions.',&
&'                 The energy output with the given reheat is calculated to determine the roundtrip cycle efficiency.'
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Enter Integer Value for Number of Stages:'
!READ(*,*) stages



                                                        !STAGES
WRITE(*,'(A,2X,F5.0,2X,A)')'          You entered',stages,'number of compression stages'
```

```
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Enter the Ambient Air Pressure in Units of kPa:'
!READ(*,*) P_in

                                                                        !INLET (ATM) PRESSURE
WRITE(*,100)'       You entered',P_in,'kPa or',P_in/6.89475729,'psi or',P_in/100,'bar for the ambient air pressure'
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Enter the Final (Target) Air Pressure in Units of kPa:'
!READ(*,*) P_out

                                                                        !STORAGE (HIGH) PRESSURE
WRITE(*,100)'You entered',P_out,'kPa or',P_out/6.89475729,'psi or',P_out/100,'bar for the final compressor outlet pressure'
WRITE(*,*)''

rc=(P_out/P_in)

WRITE(*,'(A,2X,F10.3)')'              The overall compression ratio is for the final pressure is:',rc              !COMPRESSION RATIO
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Enter the Working Pressure for the system in Units of kPa:'
!READ(*,*)P_low

                                                                        !WORKING (LOW) PRESSURE
WRITE(*,100)'You entered',P_low,'kPa or',P_low/6.89475729,'psi or',P_low/100,'bar for the initial turbine inlet pressure'
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Enter the Compressor Inlet Temperature for Each Stage [K]:'
!READ(*,*) T_in

                                                                        !INLET TEMPERATURE
WRITE(*,100)'You entered',T_in,'K or',T_in*1.8,'R or',(T_in*1.8)-459.67,'F as the inlet temp for each compression stage'
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Please enter the DEWPOINT TEMPERATURE (integer between 253 K and 313 K) for the
compressor inlet air:'
!READ(*,*)T_wb
WRITE(*,100)'       You entered',T_wb,'K or',T_wb*1.8,'R or',(T_wb*1.8)-459.67,'F as the dewpoint inlet temp'
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Enter the Steady State Receiver Temperature [K] at the Working Pressure:'
!READ(*,*)T_tank

                              !DISCHARGED STORAGE TANK TEMPERATURE
WRITE(*,100)'You entered',T_tank,'K or',T_tank*1.8,'R or',(T_in*1.8)-459.67,'F as the discharged receiver temperature'
WRITE(*,*)''

!WRITE(*,'(A,2X)',ADVANCE='NO')'Enter the Isentropic Compressor Efficiency in terms of PERCENT:'
!READ(*,*) eta_percent

                                !COMPRESSOR EFFICIENCY (DECIMAL)
WRITE(*,100)'                 You entered',eta_percent,'% as the Isentropic Compressor Efficiency'
```

```
WRITE(*,*)''

eta_c=eta_percent/100


                                        !COMPRESSOR EFFICIENCY (%)

!WRITE(*,'(A,/A,2X)',ADVANCE='NO')'Enter the Constant PRESSURE Specific Heat of Air for the',&
!
                                        &'Given Inlet Temperature and Pressure in Units of kJ/kg-K:'
!READ(*,*) cp



                                                                        !INLET Cp
!WRITE(*,100)'You entered',cp,'kJ/kg-K'

!WRITE(*,'(A,/A,2X)',ADVANCE='NO')'Enter the Constant VOLUME Specific Heat of Air for the',&
!
                                        &'Given Inlet Temperature and Pressure in Units of kJ/kg-K:'
!READ(*,*) cv



                                                                        !INLET Cv
!WRITE(*,100)'You entered',cv,'kJ/kg-K'

!-----------------------------CALL SUBROUTINE-----------------------------------

WRITE(*,*)''
WRITE(*,*)'                                        <<CALL specific_heat_calc SUBROUTINE>>'
WRITE(*,*)''

CALL specific_heat_calc (T_in, R_gas, cp, cv, k)        !CALC cp, cv, & k

!--------------------------------------------------------------------------------
!KEEP THESE COEFFICIENTS FOR DEBUGGING--USED IN EXPANDER SUBROUTINE (OLD METHOD --- USE
specific_heat_calc above now)

!a=3.653

                                        !Assign enthalpy equation coefficients
!b=-1.334E-03
!c=3.291E-06
                                        !Why is 'c' blue?  It is declared REAL, not CHARACTER...
!d=-1.91E-09
!e=2.75E-13

!cp=(R_gas)*((a)+(b*T_in)+((c)*(T_in**2))+((d)*(T_in**3))+((e)*(T_in**4)))

WRITE(*,100)'The Constant PRESSURE (Isobaric) Specific Heat of Air for the given inlet temperature is',cp,'kJ/kg-K'
WRITE(*,*)''

!cv=cp-R_gas

WRITE(*,100)'The Constant VOLUME (Isochoric) Specific Heat of Air for the given inlet temperature is',cv,'kJ/kg-K'
WRITE(*,*)''

WRITE(*,100)'                    The Ratio of Specific Heats, k, for the air at the inlet is',k
WRITE(*,*)''

!-----------------------------CALL SUBROUTINE-----------------------------------

WRITE(*,*)''
```

```
WRITE(*,*)'                                                               <<CALL spec_comp_work SUBROUTINE>>'
WRITE(*,*)''

CALL spec_comp_work (stages, rc, rp, P_in, P_out, T_in, T_out, cp, cv, eta_c, w_in, k, R_gas)          !CALC COMPRESSOR WORK

!------------------------LOOK UP ENTHALPY PER STAGE-----------------------------

110 FORMAT(A,2X,F5.0,1X,F6.2,1X,/A,2X,F5.0,1X,F6.2)

!WRITE(*,*)'Here are the inlet temps for the enthalpy calcs!'

Ttab(0)=0
htab(0)=0
Prtab(0)=0
utab(0)=0
vrtab(0)=0
sotab(0)=0

DO i=1,m

comp_int_quot_cold=(T_in-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
h_comp_cold=htab(i-1)+(comp_int_quot_cold*(htab(i)-htab(i-1)))

!WRITE(*,110)'The low temp/enthalpy for the inlet is'&
!&,Ttab(i-1),htab(i-1),'and the high temp/enthalpy for the inlet is',Ttab(i),htab(i)

IF (Ttab(i) > T_in)EXIT

END DO

!WRITE(*,*)'Here are the outlet temps for the enthalpy calcs!'

DO i=1,m

comp_int_quot_hot=(T_out-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
h_comp_hot=htab(i-1)+(comp_int_quot_hot*(htab(i)-htab(i-1)))

!WRITE(*,110)'The low temp/enthalpy for the outlet is'&
!&,Ttab(i-1),htab(i-1),'and the high temp/enthalpy for the outlet is',Ttab(i),htab(i)

IF (Ttab(i) > T_out)EXIT
END DO

!---------------------------------ANSWER----------------------------------------

WRITE(*,100)'                The air will be compressed',rp,'times per compression stage [stage ratio]'
WRITE(*,*)''

WRITE(*,100)'                 The given combination will require:',w_in,'kJ of energy per kg of air'
WRITE(*,*)''

WRITE(*,'(A,/A,2X,F7.3,1X,A,/A)')'     The outlet temperature for each stage, if cooled to',&
&'                                    the initial inlet temperature for each stage, is:',T_out,'[K]'&
&,'','                     <THIS IS THE VALUE INCLUDING THE ISENTROPIC COMP EFFICIENCY.>'

WRITE(*,100)'                The enthalpy per kilogram available from intercooling and aftercooling'
WRITE(*,100)'                     the compressed air from',T_out,'K to a temperature of',T_in,'K'
WRITE(*,100)'is',(h_comp_hot-h_comp_cold),'kJ/kg per stage and',(h_comp_hot-h_comp_cold)*stages,'kJ/kg over the process.'
WRITE(*,100)''
WRITE(*,*)'         NOTE: this "total specific enthalpy" number is the total amount of energy removed from the air'
WRITE(*,*)'         and should not be confused with another specific enthalpy associated with a higher temperature'
WRITE(*,*)''

!THIS 'TOTAL SPECIFIC ENTHALPY' NUMBER/CONCEPT MAY BE MISLEADING! THE COOLING FLUID ACCEPTING THE
HEAT
```

```
!WILL HAVE A HIGHER MAGNITUDE OF ENERGY IF MORE ENERGY IS REMOVED FROM THE AIR, BUT THESE
!DISCRETE AMOUNTS WILL NOT BE ADDITIVE (I.E. INCREASE TEMPERATURE) FOR OTHER USES, I.E. PREHEATING
WATER.
!THIS WILL SIMPLY BE A GREATER AMOUNT OF LOWER QUALITY HEAT AVAILABLE FOR USES AT THE COOLING
FLUID TEMP

!--------------------------CHECK DATA FILE EXISTENCE----------------------------

!WRITE(*,'(A,2X)',ADVANCE='NO')'Please enter a file name that includes the suffix ".dat" you would like to use/create:'
!READ(*,*)fname



                                                                  !(OPEN/CREATE) EXCESS POWER DATA INPUT FILE
INQUIRE(FILE=fname,EXIST=xst,IOSTAT=ioerr)
IF(ioerr /= 0)STOP '                    The file operation INQUIRE did not complete successfully'
IF(.NOT. xst)THEN
WRITE(*,*)'                        This file does not exist!  Whatchoo gotta prob"m or sum"n?'     !CALL enter_w_dot_xs (fname,n)
                                             !If file does not exist, call entry subroutine
ELSE
WRITE(*,'(A,2X)')'       This file already exists. Please enter number of hours for which the file has data:'        !If it exists, continue
!                                                       READ(*,*)n
END IF
WRITE(*,*)''
WRITE(*,*)n,'hours'

!-----------------------------CALL SUBROUTINE-----------------------------------

WRITE(*,*)''
WRITE(*,*)'                                                                      <<CALL volume_choice SUBROUTINE>>'
WRITE(*,*)''

CALL volume_choice (V_m, V_g, P_low, P_out, T_tank, w_in, fname, n, del_m, W_dot_sum, R_gas, Comp_powmax, m_dot_max)
                                                  !CALC TANK VOLUME

!WRITE(*,*)'                                        DEBUG'

!--------------------APPEND results300K.DAT FILE WITH DATA--------------------------

OPEN(UNIT=10,FILE='results300K.dat',STATUS='OLD',ACTION='WRITE',POSITION='APPEND',IOSTAT=ioerr)!Open & Append
OUTPUT FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

WRITE(10,*)'-------------------------------------------------------------------------------------------------------------------'
WRITE(10,100)'The Compressor Inlet Temperature for each compression stage (with intercooling) is:',T_in,'K or',T_in*1.8,'R'
WRITE(10,*)''
WRITE(10,100)'The Steady State Receiver Temperature is',T_tank,'K or',T_tank*1.8,'R'
WRITE(10,*)''
WRITE(10,100)'There are',stages,'Compressor Stages'
WRITE(10,*)''
WRITE(10,100)'The Isentropic Compressor Efficiency is',eta_percent,'%'
WRITE(10,*)''
WRITE(10,100)'The Ambient Air Pressure is:',P_in,'kPa or',P_in/6.89475729,'psi'
WRITE(10,*)''
WRITE(10,100)'The Maximum (Storage) Pressure for the system is:',P_out,'kPa or',P_out/6.89475729,'psi'
WRITE(10,*)''
WRITE(10,100)'The Working (Expansion) Pressure for the system is',P_low,'kPa or',P_low/6.89475729,'psi'
WRITE(10,*)''
WRITE(10,100)'The overall compression ratio is:',rc,'and the stage compression ratio is',rp
WRITE(10,*)''
WRITE(10,100)'The Constant PRESSURE Specific Heat of Air for the given inlet temperature is',cp,'kJ/kg-K'
WRITE(10,*)''
WRITE(10,100)'The Constant VOLUME Specific Heat of Air for the given inlet temperature is',cv,'kJ/kg-K'
WRITE(10,*)''
WRITE(10,100)'The Total Specific Compression Work is',w_in,'kJ/kg'
```

```
WRITE(10,*)''
WRITE(10,100)'The Heat Per kg Removed per stage by intercooling and aftercooling is',(h_comp_hot-h_comp_cold),'kJ/kg'
WRITE(10,*)'----------------------------------------------------------------------------------------------------'
WRITE(10,*)''

WRITE(*,'(F15.3,A)')V_m,'m3 is the minimum receiver volume calculated for the PEAK DAY.'
WRITE(*,'(A,2X)',ADVANCE='NO')'Please enter the desired receiver size in m3:'
READ(*,*)V_tank
!V_tank=V_m !tank_gal/264.172

!WRITE(*,*)'                                        DEBUG'

WRITE(*,100)'The receiver volume being used is:',V_tank,'m3, instead of the minimum:',V_m,'m3'
WRITE(*,*)''

WRITE(10,100)'The receiver volume being used is:',V_tank,'m3, instead of the minimum:',V_m,'m3'
WRITE(10,*)''
!---------------------------CALL SUBROUTINE---------------------------------

WRITE(*,*)''
WRITE(*,*)'                                                                      <<CALL humid_rat SUBROUTINE>>'
WRITE(*,*)''

CALL humid_rat (rc, P_out, T_in, T_wb, T_min, W_rem)

!WRITE(*,*)''
!WRITE(*,*)'                                        ',T_wb,'K'
!WRITE(*,*)'                                        ',T_min,'K'
!WRITE(*,*)''
!---------------------------CALL SUBROUTINE---------------------------------

WRITE(*,*)''
WRITE(*,*)'                                                                      <<CALL tank_temp SUBROUTINE>>'
WRITE(*,*)''

CALL tank_temp (T_in, T_tank, V_tank, T_out, P_low, P_out, R_gas, n, m_discharged, T_fill, T_empty) !CALC TANK TEMP (FILL)

!T_amb=T_in
!P_amb=P_in

WRITE(10,*)''
WRITE(10,*)T_min,'K is the minimum temperature (uncompressed) the air can be cooled to before condensation occurs'
WRITE(10,*)''
WRITE(10,*)W_rem,'kg water per kg dry air is removed during compression/cooling'
WRITE(10,*)''
WRITE(10,*)'If no air is removed from the receiver during the specified period,'
WRITE(10,*)'the final receiver temperature is:',T_fill,'K,'
WRITE(10,*)''
WRITE(10,*)'----------------------------------------------------------------------------------------------------'
WRITE(10,*)''

CLOSE(10)

m_dot_xpd=m_dot_max

!---------------------------CALL SUBROUTINE---------------------------------

WRITE(*,*)''
WRITE(*,*)'                                                                      <<CALL expander SUBROUTINE>>'
WRITE(*,*)''

CALL expander (P_low, P_in, T_in, del_m, W_dot_sum, T_tank, k, R_gas, m_dot_xpd, eta_t_percent)      !CALC TURBINE WORK
OUTPUT, REHEAT

!WRITE(*,*)'k=',k                                                                                      !Debug
```

```
!---------------------------------ANSWER----------------------------------------

WRITE(*,*)'                    Calculated data is logged in file "results300K.dat" in your home directory.'                    !OUTPUT FILE

DEALLOCATE(Ttab,htab,Prtab,utab,vrtab,sotab)

CALL real_run (V_tank, R_gas, T_in, h_comp_hot, h_comp_cold, stages, m_discharged, m_dot_xpd, w_in,&
& P_low, P_in, del_m, W_dot_sum, T_tank, k, eta_t_percent, P_out)

END SUBROUTINE comp_work_check

!------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE specific_heat_calc (T_in, R_gas, cp, cv, k)

!This subroutine calculates the specific heat [kJ/kg-K] for air

IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-------------------------Input/Output Variables-------------------------------

REAL, INTENT(IN) :: T_in
                                        !inlet air temperature (per stage) [K]
REAL, INTENT(IN) :: R_gas
                                                !gas constant for air
REAL, INTENT(OUT) ::cp
                                                            !constant pressure specific heat [kJ/kg-K]
REAL, INTENT(OUT) ::cv
                                                            !constant volume specific heat [kJ/kg-K]
REAL, INTENT(OUT) ::k
                                                                   !ratio of specific heats [unitless]

!----------------------------Local Variables----------------------------------

REAL :: N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11        !equation coefficients
REAL :: u,v,w

                                          !calculated equation coefficients
REAL :: EQ1,EQ2,EQ3,EQ4,EQ5
                                          !calculated equation coefficients

!-----------------------------------------------------------------------------

N1=3.490888032
N2=2.395525583E-6
N3=7.172111248E-9
N4=-3.115413101E-15
N5=0.223806688
N6=0.791309509
N7=0.212236768
N8=0.197938904
N9=3364.011
N10=2242.45
N11=11580.4

u=N9/T_in
v=N10/T_in
w=N11/T_in

EQ1=N1+(N2*T_in)+(N3*(T_in**2))+(N4*(T_in**3))
EQ2=N5/(T_in**1.5)
EQ3=(N6*(u**2)*(EXP(u)))/((EXP(u)-1)**2)
EQ4=(N7*(v**2)*(EXP(v)))/((EXP(v)-1)**2)
EQ5=((2/3)*N8*(w**2)*(EXP(-w)))/(((2/3)*(EXP(-w))+1)**2)

cp=R_gas*(EQ1+EQ2+EQ3+EQ4+EQ5)
cv=cp-R_gas
k=cp/cv
```

```
END SUBROUTINE specific_heat_calc
!----------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!----------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE spec_comp_work (stages, rc, rp, P_in, P_out, T_in, T_out, cp, cv, eta_c, w_in, k, R_gas)

!This subroutine calculates the ideal specific compressor work [kJ/kg] for
!isentropic intercooled staged air compression

IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!--------------------------Input/Output Variables-------------------------------

REAL, INTENT(IN) :: stages                                              !number of stages
REAL, INTENT(IN) :: rc
                                        !overall compression ratio
REAL, INTENT(OUT) :: rp
                              !compression ratio per stage, internal variable
REAL, INTENT(IN) :: P_in                                    !inlet (ambient) air pressure [kPa]
REAL, INTENT(IN) :: P_out                                   !final (target) air pressure [kPa]
REAL, INTENT(IN) :: T_in                                    !inlet air temperature (per stage) [K]
REAL, INTENT(OUT) :: T_out                                  !outlet air temperature (per stage) [K]
REAL, INTENT(IN) :: cp
                              !constant pressure specific heat of air at given
!temperature and pressure [kJ/kg-K]
REAL, INTENT(IN) :: cv
                              !constant volume specific heat of air at given
!temperature and pressure [kJ/kg-K]
REAL, INTENT(IN) :: eta_c                                   !isentropic compressor efficiency
REAL, INTENT(OUT) :: w_in                        !specific work out for given parameters [kJ/kg]
REAL, INTENT(IN) :: k
                    !cp/cv specific heat ratio       MAY WANT TO MAKE k INTENT IN
REAL, INTENT(IN) :: R_gas                 !gas constant for dry air, internal variable [kJ/kg-K]

!-----------------------------Local Variables-----------------------------------

REAL :: P_int
                                              !pressure rise per stage, internal variable [kPa]
!REAL :: l
                                                                      !k-1, internal variable
REAL :: w_i
                                                                  !specific stage work [kJ/kg]
REAL :: kout
                                              !Calculated ratio of specific heats for outlet temp
REAL :: kavg, kavgold
                                        !Average ratio values

!--------------------------------Equations--------------------------------------

rp = rc**(1/stages)
                              !Calculate compression ratio per stage (rp)

P_int = ((P_in**(stages-1))*P_out)**(1/stages)!Calculate pressure rise per stage (p_int)

!R_gas = 8.31446/28.970
                                        !Calculate R_gas

!k = cp/cv
                                                                      !Calculate k and l
```

```
!l = k-1

kout=k

                                        !Initialize specific heat ratios
kavg=k
kavgold=2
T_out = T_in*(rp**((k-1)/k))                                    !Initialize the outlet temp then write it FSAG

WRITE(*,*)'                        DEBUG: Initialized T_out',T_out
WRITE(*,*)''
WRITE(*,*)'                               kavgold                                 kavg
                                            T_in
                                          T_out
                                           w_i'
WRITE(*,*)''

DO WHILE(ABS(kavg-kavgold)>0.0000001)

w_i = (((kavg*R_gas*T_in)/(kavg-1))*((rp**((kavg-1)/kavg))-1))/eta_c                !Calculate specific stage work w_i

T_out=T_in*(rp**((kavg-1)/kavg))
                                                                      !Calculate hot expansion outlet temp

!                       T_out = T_in+((w_i)*((kavg-1)/(kavg*R_gas)))
                                                                                             !As a check

CALL specific_heat_calc (T_out, R_gas, cp, cv, kout)              !Call the specific heat SUB for the outlet temp

kavgold=kavg
                                          !The 'ol switcheroo

kavg=(k+kout)/2
                                          !Average the values

WRITE(*,*)'                         ',kavgold,kavg,T_in,T_out,w_i

END DO

w_in = (w_i*stages)                                                        !Calculate total specific work w_in

!T_out = T_in*(rp**(l/k))

WRITE(*,*)''
WRITE(*,*)'DEBUG: T_out, DTs way:', T_in*(rp**((kavg-1)/(eta_c*kavg)))!T_in+((w_i/eta_c)*((kavg-1)/(kavg*R_gas))) !Calculate
outlet air temperature T_out

WRITE(*,*)''
WRITE(*,*)'           Total work IN is',w_in,'T_out is',T_in+(((T_in*(rp**((kavg-1)/(kavg))))-T_in)/eta_c)
WRITE(*,*)''

END SUBROUTINE spec_comp_work

!-------------------------------------------------------------------------------------------------------------------------
```

```
!---------------------------------------------------------------------------------------------------------------------------------------------------------------

!PROGRAM enter_w_dot_xs
                                        !To run as a program instead of a subroutine
SUBROUTINE enter_w_dot_xs (fname,n)                              !To run as a subroutine instead of a program

!This subroutine takes user input for hourly excess power and exports them to a file in n X 2 matrix form [hour | power]

IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-------------------------INPUT/OUTPUT VARIABLES--------------------------------

INTEGER, INTENT(OUT) :: n
                                                        !Index
CHARACTER(len=20)::fname !,answer, answer2!Input file name, joke variables (delete excl point to activate)

!----------------------------LOCAL VARIABLES------------------------------------

INTEGER :: i,m,ioerr
                                                                              !Indices, error variable
REAL, ALLOCATABLE :: W_dot_xs(:,:)                                            !Excess work matrix

!---------------------------------JOKES-----------------------------------------

!!(delete exclamation points to activate)

!WRITE(*,'(A,2X)',ADVANCE='NO')'Would you like to play a game?'
!READ(*,'(A)')answer
!IF(answer=='thermonuclear warfare'.OR.answer=='Thermonuclear Warfare')THEN
!22                  WRITE(*,*)'Are you sure you would not rather play a nice game of chess?'
!                                   READ(*,'(A)')answer2
!                       IF(answer2=='no'.OR.answer2=='NO'.OR.answer2=='No')THEN
!                                                                                     GO TO 22
!                                            END IF
!END IF
!WRITE(*,*)'Ha Ha Ha. Just kidding!! Hit ENTER to continue'
!READ(*,*)

!-------------------------------END JOKES---------------------------------------

!WRITE(*,'(A,2X)',ADVANCE='NO')'Please enter a file name that includes the suffix ".dat" you would like to create:'
!READ(*,*)fname

OPEN(UNIT=11,FILE=fname,STATUS='REPLACE',ACTION='WRITE',IOSTAT=ioerr)              !Open the file
IF(ioerr /= 0)STOP'There was an error with your data file.  Please check the filename and start again.'

WRITE(*,'(A,2X)',ADVANCE='NO')'Enter the number of hours you will be inputting data for:'
READ(*,*)n

m=2

                                                                              !CHOOSE n X (2) matrix
ALLOCATE(W_dot_xs(n,m))
                                             !Allocate to n X 2 matrix

WRITE(*,*)'Your first hourly excess power value must be a NON-ZERO positive number!'
```

```
DO i=1,n

                                   !ENTER DATA FOR EACH HOUR >> WRITE TO SCREEN AND FILE
WRITE(*,'(A,2X,I3,A,2X)',ADVANCE='NO')'Enter the EXCESS POWER in kW for hour',i,':'
READ(*,*)W_dot_xs(i,m)
WRITE(*,'(A,2X,F10.3,2X,A)')'You entered',W_dot_xs(i,m),'kW'
WRITE(11,'(I3,4X,F10.3)')i,W_dot_xs(i,m)
END DO

WRITE(*,'(A,4X,A,/A,4X,A)')'HOUR','EXCESS POWER [kW]','====','================='

DO i=1,n

                                     !WRITE DATA TABLE TO SCREEN WITH Kewl LABELS ABOVE
WRITE(*,'(I3,4X,F10.3)')i,W_dot_xs(i,m)
END DO

WRITE(*,'(A,/A25,A)')'The data is stored in the data file',fname,'in your home directory.'

DEALLOCATE(W_dot_xs)
                                                                                                                                                  !Give up the matrix
CLOSE(11)

                                                                                                  !Close the file

END SUBROUTINE enter_w_dot_xs
                                                              !If run as a subroutine instead of a program
!END PROGRAM enter_w_dot_xs                                          !If run as a program instead of a subroutine

!------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE volume_choice (V_m, V_g, P_low, P_out, T_tank, w_in, fname, n, del_m, W_dot_sum, R_gas, Comp_powmax,
m_dot_max)

IMPLICIT NONE

!This subroutine calculates the minimum required reciever volume m**3 for a compressed
!air energy storage system, based on working pressure, maximum reciever pressure,
!steady-state receiver temperature, and specific isentropic compressor work.

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-------------------------INPUT/OUTPUT VARIABLES--------------------------------

REAL, INTENT(OUT) :: V_m                                          !Tank Volume [m**3]
REAL, INTENT(OUT):: V_g
                                                    !Tank Volume [gal]
REAL, INTENT(IN) :: P_low                                         !Working (base/low) Pressure [kPa]
REAL, INTENT(IN) :: P_out                                         !Storage Pressure (max/high) [kPa]
REAL, INTENT(IN) :: T_tank                      !Storage Tank Temperature [K] at Discharged Steady State
REAL, INTENT(IN) :: w_in                                 !Specific Work required for compression [kJ/kg]
REAL, INTENT(OUT) :: del_m             !Mass differential between max/high and base/low pressure states [kg]
REAL, INTENT(OUT) :: W_dot_sum                                    !Sum of all excess power values
REAL, INTENT(IN) :: R_gas                                         !Gas constant [kJ/kg-K]
REAL, INTENT(OUT) :: Comp_powmax          !Max compressor power [kW]
INTEGER, INTENT(IN) :: n                                          !Number of hours for which there are data
CHARACTER(20), INTENT(IN) ::fname
                                             !File name of excess power input file
REAL, INTENT(OUT) :: m_dot_max                                    !Maximum compressor mass flow rate [kg/s]

!----------------------------LOCAL VARIABLES----------------------------------

INTEGER :: i,ioerr,m
                                        !Index, file error variable, matrix size variables
REAL :: ro_low
                                       !Gas density at working pressure and tank temperature [kg/m**3]
REAL :: ro_high
                                       !Gas density at storage pressure and tank temperature [kg/m**3]
REAL :: m_low
                                       !Mass of air in tank at working (low) pressure [kg]
REAL :: m_high
                                       !Mass of air in tank at storage (high) pressure [kg]
REAL, ALLOCATABLE :: m_dot(:)                                     !Hourly mass flow rate [kg/s]
REAL, ALLOCATABLE :: m_comp(:)                                    !Mass compressed per hour [kg]
REAL, ALLOCATABLE :: W_dot(:)                                     !Compresser power per hour [kW]
REAL, ALLOCATABLE :: foe(:)                        !Hourly fraction of excess power [unitless/decimal]
REAL, ALLOCATABLE :: W_dot_xs(:,:)            !Hourly Excess Power [kW]
REAL :: SUMMER
                                                                  !Dummy variable for calculating volume

!-----------------------------------------------------------------------------

!R_gas = 8.31446/28.970
                                                    !Calculate R_gas

V_m = 1

                                                    !Initialize V_m [m**3]
```

```
ro_low = P_low/(R_gas*T_tank)                                   !Calculate densities
ro_high = P_out/(R_gas*T_tank)

!---------------FILE OPERATIONS-------------------------------------------------

!These first lines are to require user input for file name pointer.
!If used, these lines may need to be included in the main program.
!The 24 hour restriction is not included in the main program as number of data points is chosen

!WRITE(*,'(A,/A)')'Data file must consist of 24 excess power values corresponding','to each hour, beginning with 12:00AM (0hr) and
!ending with 11:00PM (23hr)'
!WRITE(*,'(A,2X)',ADVANCE='NO')'Please enter the file name of the .dat file you would like to open:'
!READ(*,*)fname

OPEN(UNIT=7,FILE=fname,STATUS='OLD',ACTION='READ',IOSTAT=ioerr)
                                                                !Open INPUT FILE
!OPEN(UNIT=7,FILE='excess_power.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)

OPEN(UNIT=8,FILE='results300K.dat',STATUS='REPLACE',ACTION='WRITE',IOSTAT=ioerr)     !Open/Create OUTPUT FILE

IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

!WRITE(*,'(A,2X)',ADVANCE='NO')'Please enter the number of hours for which your file has data:'
!READ(*,*)n

m=2

                                                                !CHOOSE n X (2) matrix

ALLOCATE(W_dot_xs(n,m))
                                                                !Allocate n X m matrix
ALLOCATE(m_dot(n))
                                                                !Allocate n sized list
ALLOCATE(m_comp(n))
                                                                !Allocate n sized list
ALLOCATE(W_dot(n))
                                                                !Allocate n sized list
ALLOCATE(foe(n))
                                                                !Allocate n sized list

DO i=1,n
READ(7,*)W_dot_xs(i,:)
                                        !Read in excess power values
END DO

!------------------------------------------------------------------------------

SUMMER=0

                                !Dummy variable to sum the excess power values

DO i=1,n

                                        !For all 24 hourly values
W_dot_sum = SUMMER+W_dot_xs(i,m)        !Sums 2nd column values, 1-24
SUMMER = W_dot_sum
                                        !Dummy variable switch before iteration
END DO

                                                !*Should* output the correct sum (kWh)

DO i=1,n

                                        !For all 24 hourly values
```

```
foe(i) = W_dot_xs(i,m)/W_dot_sum    !Calculates per hour percentage of excess power (energy)
END DO

                                        !*Should* output the correct percentages

DO i=1,n

                        !CHANGED INDEX FROM m TO n, BECAUSE m=2

W_dot(i)=0
                                        !Initialize W_dot(i) numbers to minimum

END DO

DO WHILE(W_dot(1) < W_dot_xs(1,m))          !Check first value for relative size, continue or exit do loop
!First excess power value MUST be non-zero, else do loop will not run
DO i=1,n
m_low = V_m*ro_low                                        !Calculate minimum mass
m_high = V_m*ro_high                                      !Calculate maximum mass
del_m = m_high-m_low                                      !Calculate mass differential
m_comp(i) = foe(i)*del_m          !Calculate hourly values for compressed mass [kg/h]
m_dot(i) = m_comp(i)/3600          !Calculate constant (hourly) mass flow rates [kg/s]
W_dot(i) = w_in*m_dot(i)      !Calculate power required for hourly compression [kW = kJ/kg*kg/s]
END DO
V_m=V_m+1

END DO

V_g=V_m*264.172

                              !Convert m**3 to gallons

102 FORMAT(A,I3,4X,F16.3,4X,F16.3,4X,F14.3,4X,F17.3,4X,F15.3)

WRITE(8,*)'Compressor Outlet Pressure:',P_out,'Turbine Inlet Pressure:',P_low
WRITE(8,*)''
WRITE(8,*)'HOUR    EXCESS WORK [KWh]    MASS FLOW [KG/S]    COMP MASS [KG]    PERCENT OF EXCESS    COMP
POWER [kW]'
WRITE(8,*)'====    =================    ================    ==============    =================
==============='

DO i=1,n
WRITE(8,102)'',i,W_dot_xs(i,m),m_dot(i),m_comp(i),(foe(i)*100),W_dot(i)                  !Print to file
END DO

Comp_powmax=MAXVAL(W_dot_xs)

m_dot_max=MAXVAL(m_dot)

WRITE(8,*)''
WRITE(8,'(A,2X,F10.3,2X,A)')'The highest available excess power is:',Comp_powmax,'kW'
WRITE(8,*)'The compressor must have a power rating at or below 120% the maximum available power.'
WRITE(8,*)''
WRITE(8,'(A,1X,F7.0,1X,A,1X,F7.3,1X,A)')'The following calculated volumes assume steady state conditions'&
&'for the receiver as',T_tank,'K and',P_out/100,'bar'
WRITE(8,*)'This volume only gives an estimate of the receiver size based on the given parameters'
WRITE(8,*)'and is based on an isothermal tank assumption, i.e. Tfinal=Tinitial'
WRITE(8,*)'In practice, the receiver temperature will fluctuate during fill and empty cycles'
WRITE(8,*)'Fill cycles will increase temperatures and pressures,'
WRITE(8,*)'while empty cycles will reduce temperatures and pressures.'
WRITE(8,*)''
WRITE(8,'(A,2X,ES10.3,2X,A)')'The minimum size for the receiver in cubic meters, for the given conditions, is:',V_m,'[m3]'
WRITE(8,*)''
WRITE(8,'(A,2X,ES10.3,2X,A)')'The minimum size for the receiver in gallons, for the given conditions, is:',V_g,'[gal]'
WRITE(8,*)''
```

```
WRITE(8,'(A,2X,ES10.3,2X,A)')'The mass in the receiver at the working pressure, for the given conditions, is:',m_low,'[kg]'
WRITE(8,*)''
WRITE(8,'(A,2X,ES10.3,2X,A)')'The mass compressed for the given period and conditions, is:',del_m,'[kg]'
WRITE(8,*)''
WRITE(8,'(A,2X,F10.3,2X,A)')'The SUM of the excess energy for the given day is:',SUMMER,'[kWh]'  !SUMMER = W_dot_sum
WRITE(8,*)''

CLOSE(7)
                                                        !Close the file
CLOSE(8)
                                                        !Close the file

WRITE(*,*)'                    Compressor Outlet Pressure:',P_out,'Turbine Inlet Pressure',P_low
WRITE(*,*)''
WRITE(*,*)'HOUR    EXCESS WORK [KWh]    MASS FLOW [KG/S]    COMP MASS [KG]    PERCENT OF EXCESS    COMP
POWER [kW]'
WRITE(*,*)'  ====    =================    ================    ==============    =================
==============='

DO i=1,n
WRITE(*,102)'                    ',i,W_dot_xs(i,m),m_dot(i),m_comp(i),(foe(i)*100),W_dot(i)                    !Print to screen
END DO

WRITE(*,*)''
WRITE(*,'(A,2X,F10.3,2X,A)')'       The highest available excess power is:',Comp_powmax,'kW'
WRITE(*,*)'          The compressor must have a power rating at or below 120% the maximum available power.'

OPEN(UNIT=15,FILE='massflow.dat',STATUS='REPLACE',ACTION='WRITE',IOSTAT=ioerr)         !Open/Create OUTPUT FILE

IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

DO i=1,n
WRITE(15,*)i,m_dot(i),m_comp(i)                                         !Export mass rate data to file "massflow.dat"
END DO

CLOSE(15)

DEALLOCATE(W_dot_xs)
                                                                                    !Give up the matrix
DEALLOCATE(m_dot)
                                                                                    !Give up the matrix
DEALLOCATE(m_comp)
                                                                                    !Give up the matrix
DEALLOCATE(W_dot)
                                                                                    !Give up the matrix
DEALLOCATE(foe)
                                                      !Give up the matrix

WRITE(*,*)''
WRITE(*,'(A,1X,F7.0,1X,A,1X,F7.3,1X,A)')'The following calculated volumes assume steady state conditions'&
&'for the receiver as',T_tank,'K and',P_out/100,'bar'
WRITE(*,*)'              This volume only gives an estimate of the receiver size based on the given parameters'
WRITE(*,*)'                        and is based on an isothermal tank assumption, i.e. Tfinal=Tinitial'
WRITE(*,*)'                 In practice, the receiver temperature will fluctuate during fill and empty cycles'
WRITE(*,*)'                              Fill cycles will increase temperatures and pressures,'
WRITE(*,*)'                          while empty cycles will reduce temperatures and pressures.'
WRITE(*,*)''
WRITE(*,'(A,2X,ES10.3,2X,A)')' The minimum size for the receiver in cubic meters is:',V_m,'[m3]'
WRITE(*,*)''
WRITE(*,'(A,2X,ES10.3,2X,A)')'      The minimum size for the receiver in gallons is:',V_g,'[gal]'
WRITE(*,*)''
WRITE(*,'(A,2X,ES10.3,2X,A)')'The mass in the receiver at the working pressure, for the given conditions, is:',m_low,'[kg]'
```

```
WRITE(*,*)''
WRITE(*,'(A,2X,ES10.3,2X,A)')'The mass compressed for the given period and conditions, is:',del_m,'[kg]'
WRITE(*,*)''
WRITE(*,'(A,2X,F10.3,2X,A)')'The SUM of the excess energy for the given period is:',SUMMER,'[kWh]'
WRITE(*,*)''

END SUBROUTINE volume_choice

!-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!---------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE humid_rat (rc, P_out, T_in, T_wb, T_min, W_rem)

IMPLICIT NONE

!This subroutine calculates the humidity ratio of, the pressure dewpoint temperature for, and the moisture content
!removed from the compressed air.

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-----------------------------------------Input/Output Variables-------------------------------------------------------

REAL, INTENT(IN) :: rc                                                                                   !Overall compression ratio
REAL, INTENT(IN) :: P_out                        !Compressor output pressure [kPa]
REAL, INTENT(IN) :: T_wb               !Dewpoint temperature of the compressor inlet air [K]
REAL, INTENT(IN) :: T_in              !Compressor inlet/intercooler/aftercooler temperature [K]
REAL, INTENT(OUT) :: T_min!Minimum temperature the turbine outlet air can get before condensation occurs [K]
REAL, INTENT(OUT) :: W_rem    !Fraction of water removed per kg of air compressed [unitless]

!---------------------------------------------Local Variables---------------------------------------------------------

REAL :: Ws_in
                                 !Humidity ratio of the compressor inlet air (ambient) [unitless]
REAL :: Ws_comp
                                !Humidity ratio of the compressed air (final, aftercooled) [unitless]
REAL :: Pw_in
                                  !Vapor pressure of the compressor inlet air (ambient) [kPa]
REAL :: Pw_cool
                                 !Vapor pressure at the intercooler/aftercooler temperature [kPa]
REAL :: Pw_comp
                               !Vapor pressure of the cooled compressed air at the final pressure [kPa]
REAL :: T_min_quot                                                 !Interpolation quotient for the minimum expansion temperature
REAL,ALLOCATABLE :: Trat(:), Tvp(:)!Temperature vectors for humidity ratio and vapor pressure arrays
REAL,ALLOCATABLE :: Wrat(:), Pw(:)!Humidity ratio and vapor pressure vectors for said arrays
INTEGER :: i,j,k,l,ioerr1,ioerr2                  !Variables for counting and file operations

!--------------------------------------------------------------------------------------------------------------------

OPEN(UNIT=76,FILE='Temp_Humid_Rat.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr1)
                                                                                                           !Open INPUT FILE

IF(ioerr1 /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

k=61
!Humidity Ratio Data Size
ALLOCATE(Trat(k+1))
                                                                                                          !Allocate k sized list
ALLOCATE(Wrat(k+1))
                                                                                                          !Allocate k sized list

123 FORMAT(32X,F8.0,4X,ES13.4)

WRITE(*,*)'                    PARTIAL TABLE OF MOIST AIR PROPERTIES (ASHRAE)'
WRITE(*,*)''
WRITE(*,*)'
                                                                                          TEMP [K]    Ws [kgw/kgda]'
WRITE(*,*)'
                                                                                        ========    ============='
```

```
!WRITE(*,*)'                                                                          Trat
                                                                                   Wrat'

DO i=1,k
READ(76,*)Trat(i),Wrat(i)
                                   !Read in humidity temperatures and data
WRITE(*,123)Trat(i),Wrat(i)
END DO

CLOSE(76)

OPEN(UNIT=77,FILE='Temp_Vapor_Press.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr2)
                                                                   !Open INPUT FILE

IF(ioerr2 /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

l=121

                                                               !Vapor Pressure Data Size

ALLOCATE(Tvp(l))
                                                                   !Allocate l sized list
ALLOCATE(Pw(l))

                                        !Allocate l sized list

WRITE(*,*)''
WRITE(*,*)'                    PARTIAL TABLE OF WATER PROPERTIES (ASHRAE)'
WRITE(*,*)''
WRITE(*,*)'
                                                               TEMP [K]    Pvapor [kPa]'
WRITE(*,*)'
                                                             ========    ============'

!WRITE(*,*)''
!WRITE(*,*)'                                                                          Tvp
                                                                                    Pv'

DO j=1,l
READ(77,*)Tvp(j),Pw(j)
                                     !Read in vapor pressure values
WRITE(*,123)Tvp(j),Pw(j)
END DO

CLOSE(77)

DO i=1,k
IF(Trat(i)>T_wb)EXIT
Ws_in = Wrat(i)
END DO

!WRITE(*,*)'                              ',Ws_in

DO j=1,l
IF(Tvp(j)>T_wb)EXIT
Pw_in = Pw(j)
END DO

!WRITE(*,*)'                              ',Pw_in

DO j=1,l
IF(Tvp(j)>T_in)EXIT
Pw_cool = Pw(j)
END DO
```

```
!WRITE(*,*)'                                                              ',Pw_cool

Pw_comp=rc*Pw_cool

!WRITE(*,*)'                                                              ',Pw_comp

Ws_comp=(0.62198*(Pw_cool/((P_out)-Pw_cool)))

!WRITE(*,*)'                                                              ',Ws_comp

Trat(0)=0
Wrat(0)=0

DO i=1,k

T_min_quot=(Ws_comp-Wrat(i-1))/(Wrat(i)-Wrat(i-1))
T_min=Trat(i-1)+(T_min_quot*(Trat(i)-Trat(i-1)))

IF (Wrat(i) > Ws_comp)EXIT
END DO


WRITE(*,*)''
WRITE(*,*)'                        ',T_wb,'K is the dewpoint temperature of the air at the compressor inlet'
WRITE(*,*)''
WRITE(*,*)'',T_min,'K is the minimum temperature (uncompressed) the air can be cooled to before condensation occurs'

W_rem =  Ws_in - Ws_comp

WRITE(*,*)''
WRITE(*,*)'                                           ',W_rem,'kg water per kg dry air removed'

DEALLOCATE(Trat,Wrat,Tvp,Pw)

END SUBROUTINE humid_rat

!----------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE tank_temp (T_in, T_tank, V_tank, T_out, P_low, P_out, R_gas, n, m_discharged, T_fill, T_empty)

IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-------------------------Input/Output Variables-------------------------------

REAL, INTENT(IN) :: T_in
                              !Compressor outlet stream temp [K] when aftercooled at final pressure [P_out]
REAL, INTENT(IN) :: T_tank
                              !Steady state receiver temp [K] when discharged to working pressure [P_low]
REAL, INTENT(IN) :: V_tank
                                                  !Receiver volume [m**3] specified by user
REAL, INTENT(IN) :: T_out
                                                    !Compressed air outlet temperature [K]
REAL, INTENT(IN) :: P_low
                                             !Working air pressure (low storage pressure) [kPa]
REAL, INTENT(IN) :: P_out
                                            !Maximum air pressure (high storage pressure) [kPa]
REAL, INTENT(IN) :: R_gas
                                                        !Gas constant for air [kJ/kg-K]
INTEGER, INTENT(IN) :: n
                                                   !Number of hours in the modeled period
REAL, INTENT(OUT) :: m_discharged                                                        !Base fill mass at T_tank [kg]
REAL, INTENT(OUT) :: T_fill
                                           !Temperature [K] after a period of filling the receiver
REAL, INTENT(OUT) :: T_empty
                              !Temperature [K] after a period of emptying the receiver (NOT USED...YET)


!----------------------------Local Variables----------------------------------

REAL :: V_tank_gal
                                                                                    !Receiver volume [GALLONS] specified by user
REAL, ALLOCATABLE :: Ttab(:)
                                                !Air table temperature [K] points (VECTOR)
REAL, ALLOCATABLE :: utab(:)
                                        !Air table specific internal energy [kJ/kg] points (VECTOR)
REAL, ALLOCATABLE :: htab(:)
                                           !Air table specific enthalpy [kJ/kg] points (VECTOR)
REAL, ALLOCATABLE :: Prtab(:)
                       !UNUSED HERE, ONLY FOR READING IN Table reference refpressure [kPa] (VECTOR)
REAL, ALLOCATABLE :: vrtab(:)
                 !UNUSED HERE, ONLY FOR READING IN Table reference refspecific volume [m3/kg] (VECTOR)
REAL, ALLOCATABLE :: sotab(:)
                    !UNUSED HERE, ONLY FOR READING IN Table reference zero-pressure entropy [kJ/kg-K] "
REAL, ALLOCATABLE :: m_dot(:)
                                   !Mass flow rate [kg/s] from the compressor to the tank (VECTOR)
REAL, ALLOCATABLE :: m_comp_h(:)                                  !Hourly mass amount [kg/h] compressed and sent to the tank
(VECTOR)
REAL, ALLOCATABLE :: m_comp_s(:)                             !Per second mass amount compressed and sent to the tank (VECTOR)
REAL, ALLOCATABLE :: ucv(:)
                                                !Specific internal energy [kJ/kg] of air in tank
REAL, ALLOCATABLE :: mcv(:)
                                                         !Total air mass [kg] in tank
```

```
REAL, ALLOCATABLE :: Tcv(:)
                                        !Transient temperature [K] of the tank
REAL, ALLOCATABLE :: Tcv_quot(:)                  !Interpolation quotient vector for transient tank temperatures
REAL :: hfill

                                        !Enthalpy [kJ/kg] of the inlet air stream
INTEGER, ALLOCATABLE :: hour(:)
                                        !Index vector (VECTOR)
INTEGER :: i,j,m,p,q,ioerr
                                        !Index variables
REAL :: tank_int_quot
                                        !Interpolation quotient for initial control volume properties @ T_tank
REAL :: fill_int_quot
                                        !Interpolation quotient for compressor inlet stream properties @ T_in

!-------------------------------------------------------------------------------

OPEN(UNIT=13,FILE='air_data_table_T_h_Pr_u_vr_so.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)!Open & Read INPUT
FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

105 FORMAT(F8.0,4X,F9.2,4X,F10.4,4X,F9.3,4X,F10.3,4X,F12.5)

m=121

ALLOCATE(Ttab(m+1),htab(m+1),Prtab(m+1),utab(m+1),vrtab(m+1),sotab(m+1))

DO i=1,m


                                        !Read in air property data

READ(13,*)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)

END DO

CLOSE(13)

WRITE(*,*)'                    T_tank @ Steady State:',T_tank
WRITE(*,*)'                         Gas Constant:',R_gas
WRITE(*,*)'                      Receiver Volume:',V_tank
WRITE(*,*)'                Base/Working Pressure:',P_low
!WRITE(*,*)n

m_discharged=(V_tank*P_low)/(T_tank*R_gas)

                                        !Mass of tank at P_low

WRITE(*,'(A,2X,F10.3,1X,A)')'  The mass in the tank at steady state conditions is',m_discharged,'kg'
WRITE(*,*)''

!-----READ IN MASS RATE DATA TO DETERMINE THE TRANSIENT TANK TEMPERATURE--------

OPEN(UNIT=16,FILE='massflow.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)          !Open & Read INPUT FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

ALLOCATE(hour(n), m_dot(n), m_comp_h(n))

DO i=1,n
READ(16,*)hour(i), m_dot(i), m_comp_h(i)
                                        !Import mass rate data from file "massflow.dat"
END DO

!DO i=1,n
```

```
!                                        WRITE(*,*)hour(i), m_dot(i), m_comp_h(i)
                                                        !Write out mass rate data from file "massflow.dat"
!END DO

CLOSE(16)

p=n*3600


                                                        !p is the number of seconds in the fill/empty cycle

ALLOCATE(ucv(p),mcv(p),m_comp_s(p),Tcv(p),Tcv_quot(p))
                                        !Allocate the vectors

DO i=1,n
q=1+(3600*(i-1))
DO j=q,(3600*i)
m_comp_s(j)=m_comp_h(i)/3600
                                                !Split mass change per hour [kg/h] into per min [kg/s] values
END DO
END DO

OPEN(UNIT=17,FILE='massflow_sec.dat',STATUS='REPLACE',ACTION='WRITE',IOSTAT=ioerr)!Open & Write to OUTPUT FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

WRITE(17,*)1,m_comp_s(1)

                                                                !Output first per second mass to file

DO i=60,p,60


                                                        !Output per second mass (one per minute) to file
WRITE(17,*)i,m_comp_s(i)
END DO

CLOSE(17)

mcv(1)=m_discharged


                                        !The first mass is the lowest

Ttab(0)=0
htab(0)=0
Prtab(0)=0
utab(0)=0
vrtab(0)=0
sotab(0)=0

DO i=1,m


                                                        !CALC initial specific internal energy @ T_tank

tank_int_quot=(T_tank-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
ucv(1)=utab(i-1)+(tank_int_quot*(utab(i)-utab(i-1)))

IF (Ttab(i) > T_tank)EXIT

END DO

DO i=1,m
```

```
                                              !CALC inlet stream specific enthalpy @ T_in

fill_int_quot=(T_in-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
hfill=htab(i-1)+(fill_int_quot*(htab(i)-htab(i-1)))

IF (Ttab(i) > T_in)EXIT

END DO

!WRITE(*,*)m_discharged,ucv(1),hfill

                        !OUTPUT to screen

DO i=2,p


                                              !CALC specific internal energies during fill
mcv(i)=mcv(i-1)+m_comp_s(i-1)
ucv(i)=((mcv(i-1)*ucv(i-1))+(m_comp_s(i-1)*hfill))/(mcv(i))
END DO

DO i=1,p


                                              !Find temperatures associated with ucvs
DO j=1,m
Tcv_quot(i)=(ucv(i)-utab(j-1))/(utab(j)-utab(j-1))
Tcv(i)=Ttab(j-1)+(Tcv_quot(i)*(Ttab(j)-Ttab(j-1)))
IF (utab(j) > ucv(i))EXIT
END DO
END DO

WRITE(*,*)'        Control Volume Mass | Control Volume Temp | Control Volume Spec Internal Energy'
!-XXXXXXXXXXXXXXXXXX-------XXXXXXXXXX-----------------XXXXXXXXXXXXX------------

WRITE(*,'(8X,F15.3,A,12X,F7.3,A,17X,F10.3,A,12X)')mcv(1),' kg',Tcv(1),' K',ucv(1),' kJ/kg'
DO i=3600,p,3600


                                              !OUTPUT selected points to screen
WRITE(*,'(8X,F15.3,A,12X,F7.3,A,17X,F10.3,A,12X)')mcv(i),' kg',Tcv(i),' K',ucv(i),' kJ/kg'
END DO

T_fill=Tcv(p)

WRITE(*,*)''
WRITE(*,*)'            If no air is removed from the receiver during the specified period,'
WRITE(*,*)'                 the final receiver temperature is:',T_fill,'K,'
WRITE(*,*)'                the total internal energy is',(ucv(p)*mcv(p)),'kJ'
WRITE(*,*)'                 and total mass at',P_out,'kPa is',mcv(p),'kg'
WRITE(*,*)''

!----------------------------OUTPUTS FOR DEBUGGING!-----------------------------

!WRITE(*,*)''
!WRITE(*,*)''
!WRITE(*,*)'                TABLE OF IDEAL GAS PROPERTIES FOR AIR (WARK)'
!WRITE(*,*)''
!WRITE(*,*)'            TEMP [K]   h [kJ/kg]     Pr     u [kJ/kg]     vr     so [kJ/kg-K]'
!WRITE(*,*)'     ========   =========   ==========   =========   ==========   ============'

!DO i=1,m

!                WRITE(*,105)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)
```

```
!END DO
!WRITE(*,*)''
CLOSE(13)

!OUTPUTS FOR DEBUGGING!
!WRITE(*,*)'This is the tank_temp SUBROUTINE, here is the SS receiver temp and the comp outlet temp'
!WRITE(*,*)T_tank
!WRITE(*,*)T_out
!DO i=1,m
!                    WRITE(*,*)Ttab(i),utab(i),htab(i)
!END DO
DEALLOCATE(Ttab, htab, Prtab, utab, vrtab, sotab, hour, m_dot, m_comp_h, ucv, mcv, m_comp_s, Tcv, Tcv_quot)
END SUBROUTINE tank_temp
!--------------------------------------------------------------------------------------------------------------------------------------------
```

```
!------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE expander (P_low, P_in, T_in, del_m, W_dot_sum, T_tank, k, R_gas, m_dot_xpd, eta_t_percent)

!This subroutine calculates the reheat necessary for expansion, given the number of
!turbine/heat exchanger couples


IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-------------------------INPUT/OUTPUT VARIABLES-------------------------------

REAL, INTENT(IN) :: P_low                                    !Working (low) pressure [kPa]
REAL, INTENT(IN) :: P_in                                     !Ambient (atmospheric) pressure [kPa]
REAL, INTENT(IN) :: T_in                                     !Ambient (atmospheric) temperature [K]
REAL, INTENT(IN) :: del_m          !Mass differential between max/high and base/low pressure states [kg]
REAL, INTENT(IN) :: W_dot_sum                                !Sum of all excess power values
REAL, INTENT(IN) :: T_tank                                   !Steady-state receiver temperature [K]
REAL, INTENT(IN) :: k
                                   !Specific heat ratio Cp/Cv
!REAL, INTENT(IN) :: a,b,c,d,e                               !Coefficients for enthalpy equation
REAL, INTENT(IN) :: R_gas                                    !Gas constant [kJ/kg-K]
REAL, INTENT(IN) :: m_dot_xpd                                !Expansion mass flow rate [kg/s]
REAL, INTENT(IN) :: eta_t_percent          !Isentropic turbine efficiency [Percent]
!REAL, INTENT(OUT) :: kJpsreq       !Energy output at given mass flow required to recup energy expended

!----------------------------LOCAL VARIABLES----------------------------------

INTEGER :: ioerr
                                   !Error variable for file I/O operations
REAL :: tstages
                                                             !Turbine/Heat Exchanger couple (stage) number
REAL :: T_hxpd
                                                             !Hot expansion Temperature (turbine inlet) [K]
REAL :: T_h_out
                                                             !Hot expansion Temperature (turbine outlet) [K]
REAL :: TIME
                                                             !Time period for expansion [h]
REAL :: t_xpd
                                                             !Time period for expansion [s]
REAL :: del_h_hot
                             !Specific enthalpy change under heated conditions [kJ/kg]
REAL :: W_hot
                                                             !Energy out hot [kJ]
REAL :: eta_t
                                                             !Isentropic turbine efficiency [Decimal]
REAL :: W_out_tstage
                             !Work output required per turbine/heat exchanger couple set
REAL, ALLOCATABLE :: P_int(:)                                !Intermediate turbine output pressure [kPa]
REAL, ALLOCATABLE :: P_max(:)                                !Dummy variable for iterations
REAL, ALLOCATABLE :: Ttab(:),htab(:),Prtab(:),utab(:),vrtab(:),sotab(:)       !Table variables
REAL :: trp
                                                             !Turbine stage ratio for expansion
REAL :: turb_int_quot_in                                     !Interpolation quotient to find inlet enthalpy
REAL :: turb_int_quot_out                                    !Interpolation quotient to find outlet enthalpy
REAL :: h_turb_in
                                   !Turbine inlet enthalpy
```

```
REAL :: h_turb_out
                                        !Turbine outlet enthalpy
INTEGER :: i,n,m
                                      !Indices for iterations, allocation
REAL :: cp,cv
                                                            !Specific heats (isobaric and isochoric)
REAL :: kin
                                                       !Calculated ratio of specific heats for inlet temp
REAL :: kout
                                                      !Calculated ratio of specific heats for outlet temp
REAL :: kavg, kavgold
                                         !Average ratio values
REAL :: kJpsreq
                                              !Specific Energy output per stage at given mass flow required to recup
energy expended
REAL :: Pr_quot_out, Pr_out
REAL :: Pr_quot_in, Pr_in
REAL :: T_sin
                                                                  !Inlet temp given eta_t=1

!-----------------------------------------------------------------------------

!!VERSION REWRITTEN FOR STATIC TURBINE FLOW AND DYNAMIC EXPANSION PERIOD

!MAY NEED TO BE OPTIMIZED FOR REHEAT AVAILABLITY >>>> FIND STAGES NEEDED (REHEAT TEMPS ARE VERY
HIGH!!!)

!THIS VERSION USES THE ISENTROPIC COMPRESSION/EXPANSION EQUATIONS TO FIND THE INLET AND OUTLET
TEMPERATURES,
!THEN MATCHES THESE TEMPERATURES WITH TABLULAR ENTHALPY DATA

!THE k VALUES USED ARE AVERAGED BETWEEN THE INLET AND OUTLET CONDITIONS (ITERATIVELY)

!-------------------------------FILE OPERATIONS-------------------------------

OPEN(UNIT=12,FILE='results300K.dat',STATUS='OLD',ACTION='WRITE',POSITION='APPEND',IOSTAT=ioerr)!Open & Append
OUTPUT FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

103 FORMAT(A,2X,F10.3,1X,A)
!104 FORMAT(A,/A,2X,F10.3,1X,A)


OPEN(UNIT=9,FILE='air_data_table_T_h_Pr_u_vr_so.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)!Open & Read INPUT
FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

105 FORMAT(F8.0,4X,F9.2,4X,F10.4,4X,F9.3,4X,F10.3,4X,F12.5)

m=121

ALLOCATE(Ttab(m+1),htab(m+1),Prtab(m+1),utab(m+1),vrtab(m+1),sotab(m+1))

DO i=1,m

READ(9,*)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)

END DO

CLOSE(9)

!-----------------------------------------------------------------------------

DO n=1,6
```

```
WRITE(*,'(A,2X,F5.2,1X,A)')'     You entered',eta_t_percent,'% as the Isentropic Turbine Efficiency'
WRITE(*,*)''

eta_t=eta_t_percent/100

tstages=REAL(n)

WRITE(*,'(A,2X,I3,1X,A)')'                  You entered',n,'Turbine/Heat Exchanger Couple Sets'
WRITE(*,*)''

trp=(P_in/P_low)**(1/tstages)                                          !Calculate Stage Ratio -- Assumes exit pressure is ambient pressure

WRITE(*,*)'                                         The turbine stage ratio is',trp
WRITE(*,*)''

ALLOCATE(P_int(n))
                                                  !Allocate intermediate pressures vector

WRITE(*,'(A,2X,F7.3,1X,A)')'You entered',m_dot_xpd,'kg/s as the design flow rate of the turbine system'
WRITE(*,*)''

t_xpd=del_m/m_dot_xpd
                                  !Calculate maximum time for expansion [seconds] @ constant tank temp

TIME=t_xpd/3600
                                                                                         !Convert time for expansion to hours


ALLOCATE(P_max(n+1))
                                                    !Allocate dummy variable vector

P_max(1)=P_low
                                                                                                        !Initialize P_max(1)

DO i=1,n
                                                                                    !Calculate and write turbine outlet pressures

P_int(i)=P_max(i)*trp
WRITE(*,'(A,2X,I3,2X,A,2X,F7.3)')'            The outlet pressure of turbine',i,'is',P_int(i)
WRITE(*,*)''
WRITE(12,'(A,2X,I3,2X,A,2X,F15.3)')'The outlet pressure of turbine',i,'is',P_int(i)
P_max(i+1)=P_int(i)

END DO

T_h_out=300

Ttab(0)=0
htab(0)=0
Prtab(0)=0
utab(0)=0
vrtab(0)=0
sotab(0)=0

DO i=1,m

Pr_quot_out=(T_h_out-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
Pr_out=Prtab(i-1)+(Pr_quot_out*(Prtab(i)-Prtab(i-1)))

IF (Ttab(i) > T_h_out)EXIT
END DO

WRITE(*,*)''
WRITE(*,*)'                                        Pr_out is',Pr_out,'at',T_h_out,'K'
WRITE(*,*)''
```

```
Pr_in=Pr_out/trp

DO i=1,m

Pr_quot_in=(Pr_in-Prtab(i-1))/(Prtab(i)-Prtab(i-1))
T_sin=Ttab(i-1)+(Pr_quot_in*(Ttab(i)-Ttab(i-1)))

IF(Prtab(i) > Pr_in)EXIT

END DO

WRITE(*,*)''
WRITE(*,*)'                    Pr_in is',Pr_in,'at',T_sin,'K'
WRITE(*,*)''


CALL specific_heat_calc (T_h_out, R_gas, cp, cv, kout)
                    !Call the specific heat SUB again for the outlet temp

CALL specific_heat_calc (T_sin, R_gas, cp, cv, kin)
                    !Call the specific heat SUB for the inlet temp

kavg=(kin+kout)/2                                        !Average the values

kJpsreq=(((kavg*R_gas)/(kavg-1))*((trp**((kavg-1)/kavg))-1))*T_sin        ![kJ/kg]

T_hxpd=T_h_out+((T_sin-T_h_out)/eta_t)

WRITE(*,*)''

WRITE(*,*)'    DEBUG: The power out per stage should be approximately',ABS(kJpsreq*m_dot_xpd),'kW'

WRITE(*,*)'DEBUG: The total power out for the system should be approximately',ABS(kJpsreq*m_dot_xpd*tstages),'kW'

WRITE(*,*)'DEBUG: The total energy out for the system should be approximately',ABS(kJpsreq*m_dot_xpd*tstages*TIME),'kWh'

WRITE(*,*)''

!------------------------LOOK UP ENTHALPY PER STAGE-----------------------------

110 FORMAT(A,2X,F5.0,1X,F6.2,1X,/A,2X,F5.0,1X,F6.2)

!WRITE(*,*)'Here are the inlet temps for the enthalpy calcs!'

DO i=1,m

turb_int_quot_in=(T_hxpd-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
h_turb_in=htab(i-1)+(turb_int_quot_in*(htab(i)-htab(i-1)))

!                  WRITE(*,110)'The low temp/enthalpy for the outlet is'&
!              &,Ttab(i-1),htab(i-1),'and the high temp/enthalpy for the outlet is',Ttab(i),htab(i)

IF (Ttab(i) > T_hxpd)EXIT
END DO

WRITE(*,*)'                    turb_int_quot_in     h_turb_in'
WRITE(*,*)''
WRITE(*,*)'                    ',turb_int_quot_in,h_turb_in

!WRITE(*,*)'Here are the outlet temps for the enthalpy calcs!'

DO i=1,m

turb_int_quot_out=(T_h_out-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
```

```
h_turb_out=htab(i-1)+(turb_int_quot_out*(htab(i)-htab(i-1)))

!                    WRITE(*,110)'The low temp/enthalpy for the inlet is'&
!                 &,Ttab(i-1),htab(i-1),'and the high temp/enthalpy for the inlet is',Ttab(i),htab(i)

IF (Ttab(i) > T_h_out)EXIT

END DO

WRITE(*,*)''
WRITE(*,*)'                    turb_int_quot_out    h_turb_out'
WRITE(*,*)''
WRITE(*,*)'                    ',turb_int_quot_out,h_turb_out
WRITE(*,*)''

!--------------------------------------------------------------------------------

!del_h_hot = eta_t*R_gas*(((a)*(T_h_out-T_hxpd))+((b/2)*((T_h_out**2)-(T_hxpd**2)))+((c/3)*((T_h_out**3)-&
!        &(T_hxpd**3)))+((d/4)*((T_h_out**4)-(T_hxpd**4)))+((e/5)*((T_h_out**5)-(T_hxpd**5))))

del_h_hot=h_turb_in-h_turb_out                          !Calculate the enthalpy change for each stage

W_hot=m_dot_xpd*del_h_hot*TIME*eta_t                    !Expansion energy out per stage [kg/s * kJ/kg * h * eta = kWh]

!I WOULD LIKE TO CHANGE THE BELOW EQUATIONS TO BE BASED ON TABLE DATA  >>> DID IT, FOUND LESS PRECISE
BUT SAME APPROXIMATE ANSWER

!FIND CHANGE IN ENTHALPY NEEDED PER STAGE, THEN MAX ENTHALPY NEEDED, THEN

!CORRESPOND THE TEMPERATURE TO THE CHANGE IN ENTHALPY FROM THE STEADY (COLD) STATE

!DO WHILE(ABS(W_hot) < ABS(W_out_tstage))                       !Change in enthalpy equation

!                    T_h_out=T_hxpd*((P_int(1)/P_low)**((k-1)/k))  !Calculate hot isentropic expansion outlet
temperature

!      del_h_hot = eta_t*R_gas*(((a)*(T_h_out-T_hxpd))+((b/2)*((T_h_out**2)-(T_hxpd**2)))+((c/3)*((T_h_out**3)-&
!        &(T_hxpd**3)))+((d/4)*((T_h_out**4)-(T_hxpd**4)))+((e/5)*((T_h_out**5)-(T_hxpd**5))))
!Calculate change in enthalpy for hot expansion (IG) FOR EACH TURBINE

!                    W_hot=m_dot_xpd*del_h_hot*TIME
                    !Calculate energy out from hot expansion

!                    IF(ABS(W_hot) >= ABS(W_out_tstage))EXIT!Exit when goal is met, otherwise inlet temp
increases

!                    T_hxpd=T_hxpd+1

                    !Switch with the dummy variable

!WRITE(*,'(A,2X,F10.3)')'The hot inlet temperature is now',T_hxpd
                    !TO CHECK REHEAT ITERATION
!WRITE(*,'(A,2X,F10.3,1X,A,2X,F10.3,1X,A)')'The total energy out is',W_hot,'kJ'&
!

                    &'and the total enthalpy change is',del_h_hot,'kJ/kg'
!END DO

!del_h_hot=del_h_hot*tstages
          !Adjust change in enthalpy for number of stages <FOR ALL TURBINES>

W_hot=W_hot*tstages
                    !Adjust work output for number of stages <FOR ALL TURBINES>
```

```
!----------------------------------------------------------------------------------

WRITE(12,103)'The Isentropic Turbine Efficiency is:',eta_t_percent,'%'
WRITE(12,*)''

WRITE(*,103)'                    The expansion process may occur for a period up to:',TIME,'hours'
WRITE(*,*)''
WRITE(12,103)'The expansion process may occur for a period up to:',TIME,'hours'
WRITE(12,*)''

WRITE(*,103)'            at the designated mass flow rate for the expansion process of:',m_dot_xpd,'[kg/s]'
WRITE(*,*)''
WRITE(12,103)'at the designated mass flow rate for the expansion process of:',m_dot_xpd,'[kg/s]'
WRITE(12,*)''

WRITE(*,103)'           This will be a volumetric flow rate of',(T_in*R_gas*m_dot_xpd)/P_in,'m3/s free air'
WRITE(*,*)''
WRITE(12,103)'This will be a volumetric flow rate of',(T_in*R_gas*m_dot_xpd)/P_in,'m3/s free air'
WRITE(12,*)''

WRITE(*,103)'                  The change in enthalpy per stage for hot expansion is:',del_h_hot,'[kJ/kg]'
WRITE(*,*)''
WRITE(12,103)'The change in enthalpy per stage for hot expansion is:',del_h_hot,'[kJ/kg]'
WRITE(12,*)''

WRITE(*,103)'                    The energy produced from hot expansion is',ABS(W_hot),'[kWh]'
WRITE(*,*)''
WRITE(12,103)'The energy produced from hot expansion is',ABS(W_hot),'[kWh]'
WRITE(12,*)''

WRITE(*,103)'                   This will be',(ABS(W_hot)*100)/W_dot_sum,'% of the input energy'
WRITE(*,*)''
WRITE(12,103)'This will be',(ABS(W_hot)*100)/W_dot_sum,'% of the input energy'
WRITE(12,*)''

WRITE(*,103)'    The hot expansion air will have to be reheated to',T_hxpd,'[K] before each stage in order for the',&
&'                              outlet temperature and pressure to return to ambient conditions'
WRITE(*,*)''
WRITE(12,103)'The hot expansion air will have to be reheated to',T_hxpd,'[K] before each stage in order for the',&
&'outlet temperature and pressure to return to ambient conditions'
WRITE(12,*)''

WRITE(*,103)'       The (outlet) temperature of the expanded air for hot isentropic expansion is:',T_h_out,'[K]'
WRITE(*,*)''
WRITE(12,103)'The (outlet) temperature of the expanded air for hot isentropic expansion is:',T_h_out,'[K]'
WRITE(12,*)''

WRITE(*,103)'               In units that real people use, this is a reheat temp of',(T_hxpd*1.8)-459.67,'F'
WRITE(*,*)''
WRITE(12,103)'In units that real people use, this is a reheat temp of',(T_hxpd*1.8)-459.67,'F'
WRITE(12,*)''

WRITE(*,103)'                           and an a outlet temp of',(T_h_out*1.8)-459.67,'F'
WRITE(*,*)''
WRITE(12,103)'and an a outlet temp of',(T_h_out*1.8)-459.67,'F'
WRITE(12,*)''

WRITE(*,103)'                        This is a temperature gradient of',(T_hxpd-T_h_out),'K'
WRITE(*,*)''
WRITE(12,103)'This is a temperature gradient of',(T_hxpd-T_h_out),'K'
WRITE(12,*)''

WRITE(*,103)'                                   or',((T_hxpd-T_h_out)*1.8),'F'
WRITE(*,*)''
WRITE(12,103)'or',((T_hxpd-T_h_out)*1.8),'F'
```

```
WRITE(12,*)''
!CLOSE(12)
DEALLOCATE(P_int,P_max)
END DO
DEALLOCATE(Ttab,htab,Prtab,utab,vrtab,sotab)
CLOSE(12)
END SUBROUTINE expander
!-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE real_run (V_tank, R_gas, T_in, h_comp_hot, h_comp_cold, stages, m_discharged, m_dot_xpd, w_in,&
& P_low, P_in, del_m, W_dot_sum, T_tank, k, eta_t_percent, P_out)

IMPLICIT NONE

!This subroutine takes the sizing parameters determined from the other subroutines
!and runs the model in fill and empty cycles, depending on excess power data inputs
!The turbines will only run at the specified design flow rate [kg/s]

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!-------------------------Input/Output Variables-------------------------------

REAL, INTENT(IN) :: V_tank                                   !Receiver volume [m3]
REAL, INTENT(IN) :: R_gas                                    !Gas constant [kJ/kg-K]
REAL, INTENT(IN) :: T_in                                     !Inlet air temperature (per stage) [K]
REAL, INTENT(IN) :: h_comp_hot                               !Enthalpy of the air at each compression stage outlet
REAL, INTENT(IN) :: h_comp_cold                              !Enthalpy of the air at each compression stage inlet
REAL, INTENT(IN) :: stages                                   !Number of compression stages
REAL, INTENT(IN) :: m_discharged              !Base fill mass at T_tank [kg]
REAL, INTENT(IN) :: m_dot_xpd                                !Expansion mass flow rate [kg/s]
REAL, INTENT(IN) :: w_in                                     !Energy per kilogram of air required for compression [kJ/kg]
REAL, INTENT(IN) :: P_low                                    !working air pressure (low storage pressure) [kPa]
REAL, INTENT(IN) :: P_in                                     !inlet air pressure (ambient pressure) [kPa]
REAL, INTENT(IN) :: del_m                                    !mass differential between max/high and base/low pressure storage
states [kg]
REAL, INTENT(IN) :: W_dot_sum                                !Sum of all excess power values
REAL, INTENT(IN) :: T_tank                                   !steady-state receiver temperature
REAL, INTENT(IN) :: k
                                              !cp/cv specific heat ratio
REAL, INTENT(IN) :: eta_t_percent             !Isentropic turbine efficiency
!REAL, INTENT(IN) :: T_min                                   !Minimum temperature the turbine outlet air can get before condensation
occurs
REAL, INTENT(IN) :: P_out                                    !Compressor Outlet Pressure

!----------------------------Local Variables----------------------------------

REAL, ALLOCATABLE :: Ttab(:)
                                    !Air table temperature [K] points (VECTOR)
REAL, ALLOCATABLE :: utab(:)
                              !Air table specific internal energy [kJ/kg] points (VECTOR)
REAL, ALLOCATABLE :: htab(:)
                                 !Air table specific enthalpy [kJ/kg] points (VECTOR)
REAL, ALLOCATABLE :: Prtab(:)
                  !UNUSED HERE, ONLY FOR READING IN Table reference refpressure [kPa] (VECTOR)
REAL, ALLOCATABLE :: vrtab(:)
              !UNUSED HERE, ONLY FOR READING IN Table reference refspecific volume [m3/kg] (VECTOR)
REAL, ALLOCATABLE :: sotab(:)
                 !UNUSED HERE, ONLY FOR READING IN Table reference zero-pressure entropy [kJ/kg-K]
REAL, ALLOCATABLE :: f(:),g(:)
                                  !Minute number and minute-averaged excess power
REAL, ALLOCATABLE :: kJin(:),kgsin(:),kgsout(:) !Excess power in, mass in, mass out
REAL, ALLOCATABLE :: Vs(:),Hout(:)                           !Volumetric flow rate in (free air), total enthalpy out (heat removed)
REAL, ALLOCATABLE :: Trec(:), Prec(:), Prrec(:), m_tank(:)!Receiver temp, pressure and mass
REAL, ALLOCATABLE :: u_tank(:), hempty(:)!Internal energy for the receiver, enthalpy of the outlet stream
REAL, ALLOCATABLE :: Trec_quot(:)                            !Receiver property quotient vector
```

```
REAL, ALLOCATABLE :: kJSUM(:)
                                        !kJ produced during that minute
INTEGER :: i, j, m, n, s, v, ioerr                              !Integers for counting and file operations
REAL :: tstages

                                        !Number of stages
REAL :: kWout

                                        !Wattage out for the turbine drivetrain
REAL :: T_hxpd

                                        !Turbine inlet
REAL :: Pr_tank

                                        !Relative pressure of the "evacuated tank"
REAL :: fill_int_quot
                                                                !Inlet stream property quotient
REAL :: hfill

                                        !Inlet stream enthalpy
REAL:: del_h_hot
                                                  !Specific enthalpy change under heated conditions [kJ/kg]
REAL:: h_turb_in
                                                                       !Turbine inlet enthalpy
REAL, ALLOCATABLE :: kJREHEAT(:)                        !Reheat energy needed for expansion process
REAL, ALLOCATABLE :: kWREHEAT(:)                 !Average per minute power needed for expansion process
REAL :: kJinPOT

                                        !Total available energy for storage

!-------------------------------------------------------------------------------

!Read in property data

OPEN(UNIT=56,FILE='air_data_table_T_h_Pr_u_vr_so.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)!Open & Read INPUT
FILE
IF(ioerr /= 0)STOP'The file operation 1 did not complete successfully. Please check file(s) and start again.'

m=121

ALLOCATE(Ttab(m+1),htab(m+1),Prtab(m+1),utab(m+1),vrtab(m+1),sotab(m+1))

DO i=1,m



                                        !Read in air property data

READ(56,*)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)
!                          WRITE(*,*)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)

END DO

CLOSE(56)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!USER INPUT!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

WRITE(*,'(A,2X)')'                 Enter the INTEGER number of turbine stages you wish to run:'
READ(*,*)tstages



                                        !Choose you turbine stages!
```

```
!Open file and read in data (per hour)

OPEN(UNIT=10,FILE='dayta_year.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)                    !Open & Read INPUT FILE
IF(ioerr /= 0)STOP'The file operation 2 did not complete successfully. Please check file(s) and start again.'

n=8760

                    !525600 DEBUG USE 8760 !Per hour, because 31,536,000 data points is a bit much for an Excel vector
!Next time it may be a good idea to create a 365 (day) X 86400 (seconds/day) array
s=525600
                                                                     !31536000 DEBUG USE 525600 !For the per minute data

ALLOCATE(f(n), g(n))
                                                  !Allocate per hour property vectors to read in

ALLOCATE(kJin(s), kgsin(s+1), kgsout(s+1), Vs(s), Hout(s))                                  !Allocate per minute property vectors

ALLOCATE(Trec(s), Prrec(s), Prec(s), m_tank(s), Trec_quot(s), kJSUM(s))                          !Allocate per minute properties

DO i=1,n

READ(10,*)f(i),g(i)                                          !f is the yearly hour, g is the energy value [kWh] for the hour

END DO

CLOSE(10)

DO i=1,n,24

WRITE(*,*)'                                                  ',f(i),g(i)
                                             !f is the yearly hour, g is the energy value [kWh] for the hour

END DO

DO i=1,n

DO j=1,60

kJin(((i-1)*60)+j)=g(i)*60
                      !Convert hourly data (kWh) into average per minute data (kJ/m) by extension 8760 X 60 = 525,600

END DO

END DO

CALL turbine (P_low, P_in, T_in, del_m, W_dot_sum, T_tank, k, R_gas, &
& m_dot_xpd, eta_t_percent, tstages, kWout, T_hxpd, del_h_hot, h_turb_in)!Calculate power output [kJ/s or kW] based on reheat & mass
removed

WRITE(*,*)''
WRITE(*,*)'                                          ',kWout,T_hxpd,w_in

DO i=1,s


                  !Get the masses per second in or out, the total mass in the reciever per second, volumetric flow, and heat out

IF(kJin(i) > 0)THEN
                                                                                                   !COMPRESSION

kgsin(i)=(kJin(i))/w_in                        !DEBUG kJin(i)/w_in !Calc mass flow rate [kg/m <for this minute>] and mass
compressed (will be positive)
kgsout(i)=0

                                                   !No mass out here, because g(i) is positive
```

```
!----------------------------> !I know that air cannot be compressed beginning to end in one second; it's a model
!                                                         Vs(i)=(kgsin(i)*R_gas*T_in)/P_in

                          !Calc volumetric flow rate [m3/s (free air)]
!                                          Hout(i)=kgsin(i)*(h_comp_hot-h_comp_cold)*stages
                            !Calc total heat out per second [kW]


ELSE IF(kJin(i) == 0)THEN

                                  !EXPANSION

kgsin(i)=0

                          !No mass in here, because g(i) is not positive
kgsout(i)=m_dot_xpd*60
                                          !The mass out per second is the design flow rate

END IF

END DO

kJinPOT=SUM(kJin)

                                           !Potential excess energy available for storage

!m_tank(0)=m_discharged

!DO i=1,s


                              !ADD IT UP, ADD IT UP

!                          IF (m_tank(i-1) <= m_discharged) THEN
                                      !Turbine stops when Tank mass is at low point (P=P_low)

!                                                                           kgsout(i)=0

!                                  END IF

!                          m_tank(i)=m_tank(i-1)+kgsin(i)-kgsout(i)
                          !ADD or SUBTRACT MASS from the tank

!END DO

!WRITE(*,*)''
!WRITE(*,*)'                  ',m_discharged, m_tank(0),T_tank
!WRITE(*,*)''
!WRITE(*,*)'                                                                                    i
                                  kgsin(i)                                        kgsout(i)
                                  m_tank(i)                                        kJin(i)
                             m_tank(i-1)+kgsin(i)-kgsout(i)'

!DO i=1,s,600

!                               WRITE(*,*)'', i, kgsin(i), kgsout(i), m_tank(i), kJin(i), m_tank(i-1)+kgsin(i)-
kgsout(i)

!END DO

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!Good To Go!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

ALLOCATE(u_tank(s+1), hempty(s+1), kJREHEAT(s), kWREHEAT(s))

!NEED T_tank  ----  GOT IT TO IMPORT, IS 300K!!
```

```
m_tank(1)=m_discharged

Ttab(0)=0
htab(0)=0
utab(0)=0
Prtab(0)=0
utab(0)=0
vrtab(0)=0
sotab(0)=0
u_tank(0)=0
hempty(0)=0
kgsin(0)=0
kgsout(0)=0
hempty(1)=0
Trec(1)=T_tank
Prec(1)=P_low
kJSUM(1)=0

DO i=1,m

                                                    !CALC initial specific internal energy @ T_tank
!WRITE(*,*)i
Trec_quot(1)=(T_tank-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
!WRITE(*,*)i+1
u_tank(1)=utab(i-1)+(Trec_quot(1)*(utab(i)-utab(i-1)))
!WRITE(*,*)i+2
Pr_tank=Prtab(i-1)+(Trec_quot(1)*(Prtab(i)-Prtab(i-1)))
!WRITE(*,*)i+3
IF (Ttab(i) > T_tank) EXIT

                                    !Exit on high interpolation limit
!WRITE(*,*)i+4
END DO

WRITE(*,*)'                    Trec_quot(1):',Trec_quot(1),'u_tank(1):',u_tank(1),'Pr_tank:',Pr_tank

DO i=1,m

                                                    !CALC inlet stream specific enthalpy @ T_in

fill_int_quot=(T_in-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
hfill=htab(i-1)+(fill_int_quot*(htab(i)-htab(i-1)))

IF (Ttab(i) > T_in)EXIT

END DO

WRITE(*,*)'                              fill_int_quot:',fill_int_quot,'hfill:',hfill

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!Work On This!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

DO i=2,s

                                            !Find temperatures & pressures associated with u_tanks

IF(Prec(i-1)<=(P_low*1.001).AND.kgsin(i)==0)THEN
                        !If the pressure is at or below the working pressure AND

kgsout(i)=0

                                                    !There is no mass inflow, stop the mass outflow
```

```
ELSEIF(Prec(i-1)>=P_out.AND.kgsout(i)==0)THEN
                                  !If the receiver pressure reaches or exceeds the stream

kgsin(i)=0

                                                                  !pressure, shut it down nah!
kJin(i)=0

END IF

m_tank(i)=m_tank(i-1)+kgsin(i)-kgsout(i)                    !ADD or SUBTRACT MASS from the tank

IF(m_tank(i)<=m_discharged)THEN

m_tank(i)=m_tank(i-1)
                                                   !If mass goes below MIN threshold, shut it down nah!

END IF

u_tank(i)=((m_tank(i-1)*u_tank(i-1))+((kgsin(i)*hfill)-(kgsout(i)*hempty(i-1))))/(m_tank(i))          !Calc internal energies

!       New Specific Internal Energy = Old Internal Energy + Inlet Enthalpy - Outlet Enthalpy ALL BY Current Tank Mass

DO j=1,m

Trec_quot(i)=(u_tank(i)-utab(j-1))/(utab(j)-utab(j-1))                              !Quotient
Trec(i)=Ttab(j-1)+(Trec_quot(i)*(Ttab(j)-Ttab(j-1)))                                !Temp
Prrec(i)=Prtab(j-1)+(Trec_quot(i)*(Prtab(j)-Prtab(j-1)))      !Ref Pressure
Prec(i)=(m_tank(i)*R_gas*Trec(i))/V_tank
                                              !Pressure
!                                                            Prec(i)=P_low*(Prrec(i)/Pr_tank)

                                               !Pressure (This causes the mass to blow up 10X over one year, as if the
tank had INF volume)
hempty(i)=htab(j-1)+(Trec_quot(i)*(htab(j)-htab(j-1)))                  !Exit stream enthalpy if extracting

IF (utab(j) > u_tank(i))EXIT !THEN

!IF (Prec(i)<=P_low) THEN
!                                       Prec(i)=Prec(i-1)
                         !STOP EXTRACTING WHEN Receiver Pressure = Working Pressure
!                                       hempty(i)=0
                                                                          !No Inlet Stream
!                                       kgsout(i)=0
                                                                            !No mass out
!END IF
!EXIT

!END IF

END DO

!!!NEED TO FIGURE OUT THE MASS OUT PROBLEM WHEN NOT EXTRACTING!!!  DID IT!!!

IF (kgsout(i)>0)THEN
                                               !If you are extracting at a set flowrate, you are producing at a set power

(kJ/s)

kJSUM(i)=(kWout*60)
                              !Work energy out is power*timestep (here timestep is 60s)!!!MAKE SURE THIS SHIT IS RIGHT,

JOHN BOY!!!

ELSE
```

```
kJSUM(i)=0
                                                            !No mass out, no extraction, no work energy out

END IF

!IF (Trec(i)>T_hxpd)THEN

!                                   kJREHEAT(i)=0
!ELSE

kJREHEAT(i)=(((tstages-1)*(del_h_hot))+(h_turb_in-hempty(i-1)))*kgsout(i)
                                   !Per minute energy in for reheat

!END IF

kWREHEAT(i)=kJREHEAT(i)/60


                                   !Average wattage in to reheat

END DO

OPEN(UNIT=51,FILE='WHOLE_YEAR_300K.DAT',STATUS='REPLACE',ACTION='WRITE',IOSTAT=ioerr)!Open/Create OUTPUT
FILE

IF(ioerr /= 0)STOP'The file operation 3 did not complete successfully. Please check file(s) and start again.'

WRITE(51,*)'Max Pressure:',MAXVAL(Prec),'Min Pressure:',MINVAL(Prec),'Max Temperature:',MAXVAL(Trec),'Min
Temperature:',MINVAL(Trec)

!!WRITE(*,*)'u_tank(i)                    =((                    m_tank(i-1)
                                          *                      u_tank(i-1)
                                          )+(                    (kgsin(i)'&
!!                                                                        &'
                                          *                             hfill
                                          )-(                     kgsout(i)
                                          *                     hempty(i-1)
                                          )))/                  (m_tank(i))'
!!WRITE(*,*)'                           m_tank(i)                     Trec(i)
                                          Prec(i)                    u_tank(i)
                                         hempty(i)'

!!WRITE(51,*)''
!!WRITE(51,*)'                                                       u_tank(1)
                                          =((                     m_tank(0)
                                          *                       u_tank(0)
                                          )+(                   kgsin(1)',&
!!                                                                        &'
                                          *                             hfill
                                          )-(                     kgsout(1)
                                          *                      hempty(0)
                                          )))/                  (m_tank(1))'
!!WRITE(51,*)'        1',u_tank(1),'= ((',m_tank(0),'*',u_tank(0),') + (',kgsin(1),'*',hfill,') - (',&
!!                                          &kgsout(1),'*',hempty(0),'))/',m_tank(1)
!!WRITE(51,*)''
!!WRITE(51,*)'                                                       m_tank(1)
                                          Trec(1)
                                          Prec(1)
                                          u_tank(1)                   hempty(0)
                                       u_tank(1)*m_tank(1)'
!!WRITE(51,*)'        1',m_tank(1),Trec(1),Prec(1),u_tank(1),hempty(0),u_tank(1)*m_tank(1)      !,kJSUM(i)
!!WRITE(51,*)''

!WRITE(51,*)''
```

```
!WRITE(51,*)'                                                                                  u_tank(i)
                                          =((                                             m_tank(i-1)
                                           *                                              u_tank(i-1)
                                          )+(                                             kgsin(i)',&
!                                          &'                                                        *
                                         hfill                                                      )-(
                                       kgsout(i)                                                     *
                                      hempty(i-1)                                                  )))/
                                      (m_tank(i))'
!WRITE(51,*)''
WRITE(51,*)'                                                                                   kgsin(i)
                                       kgsout(i)                                              m_tank(i)
                                        Trec(i)
                                        Prec(i)
                                       u_tank(i)                                            hempty(i-1)
                                   u_tank(i)*m_tank(i)'

DO i=1,s                         !!PUT IT THE INDEX BACK TO 2

!                                     WRITE(51,*)''
!                                     WRITE(51,*)i,'          ',u_tank(i),'= ((',m_tank(i-1),'*',u_tank(i-1),') +
(',kgsin(i),'*',hfill,') - (',&
!
                                &kgsout(i),'*',hempty(i-1),'))/',m_tank(i)
!!                                    WRITE(51,*)''
!!                                    WRITE(51,*)'
                                        kgsin(i)                                              kgsout(i)
                                       m_tank(i)                                                Trec(i)
                                                                                                Prec(i)
                                                                                              u_tank(i)
                                      hempty(i-1)                                   u_tank(i)*m_tank(i)'

WRITE(51,*)i,kgsin(i),kgsout(i),m_tank(i),Trec(i),Prec(i),u_tank(i),hempty(i-1),u_tank(i)*m_tank(i)

!!PUT THE TAB BACK BETWEEN i AND kgsin

!Calculate tank pressure and temperature              NEED TO IMPORT CODE FROM TANK_TEMP SUBROUTINE
!Calculate (SUM) power out to find energy produced [hourly]

!!NEED TO CALCULATE TANK TEMP, PRESSURE AND MASS, THEN PUT INTO VARIABLE VECTORS Trec(:), Prec(:), Mrec(:)
for i+1 value

END DO

OPEN(UNIT=62,FILE='POWER_OUT_300K.DAT',STATUS='REPLACE',ACTION='WRITE',IOSTAT=ioerr)!Open/Create OUTPUT
FILE

IF(ioerr /= 0)STOP'The file operation 4 did not complete successfully. Please check file(s) and start again.'

WRITE(62,*)''
WRITE(62,*)'                                                                                     kJin(i)
                                                                                             kJSUM(i)
                                      kJREHEAT(i)                                         kWREHEAT(i)'

DO i=1,s

WRITE(62,*)i,kJin(i),kJSUM(i),kJREHEAT(i),kWREHEAT(i)

END DO

WRITE(62,*)''
WRITE(62,*)'The YEARLY EXCESS ENERGY available for storage is:',kJinPOT,'kJ'
WRITE(62,*)'The TOTAL YEARLY sum of the energy actually stored is:',SUM(kJin),'kJ'
WRITE(62,*)'This means that',(SUM(kJin)*100)/kJinPOT,'% of the energy was stored'
WRITE(62,*)'The TOTAL YEARLY sum of reheat energy is:',SUM(kJREHEAT),'kJ'
```

```
WRITE(62,*)'The TOTAL YEARLY sum of the energy out is:',SUM(kJSUM),'kJ'
WRITE(62,*)'The payback for the year is:',((SUM(kJSUM)-SUM(kJREHEAT))*100)/SUM(kJin),'%'
WRITE(62,*)'The MAX energy in for any given minute is:',MAXVAL(kJin)
WRITE(62,*)'The ENERGY OUT per minute for the turbine array is:',kWout*60
WRITE(62,*)'The MAX kJ/m needed for REHEAT is:',MAXVAL(kJREHEAT)
WRITE(62,*)'The MAX kW needed for REHEAT is:',MAXVAL(kWREHEAT)
WRITE(62,*)'The Turbine Inlet Temperature is:',T_hxpd,'K'

WRITE(*,*)''
WRITE(*,*)'The YEARLY EXCESS ENERGY available for storage is:',kJinPOT,'kJ'
WRITE(*,*)'The TOTAL YEARLY sum of the energy actually stored is:',SUM(kJin),'kJ'
WRITE(*,*)'This means that',(SUM(kJin)*100)/kJinPOT,'% of the energy was stored'
WRITE(*,*)'The TOTAL YEARLY sum of reheat energy is:',SUM(kJREHEAT),'kJ'
WRITE(*,*)'The TOTAL YEARLY sum of the energy out is:',SUM(kJSUM),'kJ'
WRITE(*,*)'The payback for the year is:',((SUM(kJSUM)-SUM(kJREHEAT))*100)/SUM(kJin),'%'
WRITE(*,*)'The MAX energy in for any given minute is:',MAXVAL(kJin)
WRITE(*,*)'The ENERGY OUT per minute for the turbine array is:',kWout*60
WRITE(*,*)'The MAX kJ/m needed for REHEAT is:',MAXVAL(kJREHEAT)
WRITE(*,*)'The MAX kW needed for REHEAT is:',MAXVAL(kWREHEAT)
WRITE(*,*)'The Turbine Inlet Temperature is:',T_hxpd,'K'

!Check power data for positivity (input)

!If positive, go to fill cycle (COMPRESSION)

!If zero or negative, go to empty cycle (EXPANSION)

!COMPRESSION (FILL) Cycle

!Use power needed per kg to find MFR [kg/s], VFR [m3/s], heat output [kJ/kg], and tank temp/pressure
!Add the outputs to the "overall numbers" (compressed mass, tank temp/pressure, heat removed)

!EXPANSION (EMPTY) Cycle

!Use design flow rate and available reheat energy to calculate power output [kJ/s], and tank temp/pressure
!Subtract the outputs from the "overall numbers" (compressed mass, tank temp/pressure, heat removed)

!After each second iteration, the "overall numbers" must be updated
!Should be 365 days/year x 24 hours/day x 3600 seconds/hour = 31,536,000 iterations (WHOLLY CREPE!)

CLOSE(51)
CLOSE(62)

DEALLOCATE(Ttab,htab,Prtab,utab,vrtab,sotab)
DEALLOCATE(f,g)
DEALLOCATE(kJin,kgsin,kgsout,Vs,Hout)
DEALLOCATE(Trec,Prrec,Prec,m_tank,Trec_quot,kJSUM)
DEALLOCATE(u_tank,hempty,kJREHEAT,kWREHEAT)

END SUBROUTINE real_run

!-------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
!-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

SUBROUTINE turbine (P_low, P_in, T_in, del_m, W_dot_sum, T_tank, k, R_gas, &
& m_dot_xpd, eta_t_percent, tstages, kWout, T_hxpd, del_h_hot, h_turb_in)

!This subroutine calculates the reheat necessary for expansion, given the number of
!turbine/heat exchanger couples


IMPLICIT NONE

!This program is a component of a Compressed Air Energy Storage Model (For Sizing and Simulation)
!Fortran code written in 2014 by John Nickels, for a joint energy storage project between
!the Mechanical and Aerospace Engineering Department and the Nuclear Engineering Department
!of North Carolina State University in Raleigh North Carolina
!Primary funding provided by the US Department of Energy and the Idaho National Laboratory
!Direct faculty contacts are Dr. Stephen Terry, PE, and Dr. Joseph Michael Doster
!All Rights Reserved

!--------------------------INPUT/OUTPUT VARIABLES-------------------------------

REAL, INTENT(IN) :: P_low                    !Working (low) pressure [kPa]
REAL, INTENT(IN) :: P_in                     !Ambient (atmospheric) pressure [kPa]
REAL, INTENT(IN) :: T_in                     !Ambient (atmospheric) temperature [K]
REAL, INTENT(IN) :: del_m                    !Mass differential between max/high and base/low pressure states [kg]
REAL, INTENT(IN) :: W_dot_sum                !Sum of all excess power values
REAL, INTENT(IN) :: T_tank                   !Steady-state receiver temperature [K]
REAL, INTENT(IN) :: k
                                             !Specific heat ratio Cp/Cv
!REAL, INTENT(IN) :: a,b,c,d,e               !Coefficients for enthalpy equation
REAL, INTENT(IN) :: R_gas                    !Gas constant [kJ/kg-K]
REAL, INTENT(IN) :: m_dot_xpd                !Expansion mass flow rate [kg/s]
REAL, INTENT(IN) :: eta_t_percent            !Isentropic turbine efficiency [Percent]
!REAL, INTENT(OUT) :: kJpsreq                !Power output at given mass flow required to recup energy expended
!REAL, INTENT(IN) :: T_min                   !Minimum temp of turbine exit stream before condensation (and
freezing) occur(s)
REAL, INTENT(IN) :: tstages                  !Turbine/Heat Exchanger couple (stage) number
REAL, INTENT(OUT) :: kWout                   !Power output of the compressor train at the given number of stages
[kW]
REAL, INTENT(OUT) :: T_hxpd                  !Hot expansion Temperature (turbine inlet) [K]
REAL, INTENT(OUT) :: del_h_hot               !Specific enthalpy change under heated conditions [kJ/kg]
REAL, INTENT(OUT) :: h_turb_in               !Turbine inlet enthalpy

!-----------------------------LOCAL VARIABLES-----------------------------------

INTEGER :: ioerr
                                             !Error variable for file I/O operations
REAL :: T_h_out
                                             !Hot expansion Temperature (turbine outlet) [K]
REAL :: TIME
                                             !Time period for expansion [h]
REAL :: t_xpd
                                             !Time period for expansion [s]
!REAL :: del_h_hot
                                             !Specific enthalpy change under heated conditions [kJ/kg]
REAL :: W_hot
                                             !Energy out hot [kJ]
REAL :: eta_t
                                             !Isentropic turbine efficiency [Decimal]
REAL :: W_out_tstage
                                             !Work output required per turbine/heat exchanger couple set
REAL, ALLOCATABLE :: P_int(:)                !Intermediate turbine output pressure [kPa]
REAL, ALLOCATABLE :: P_max(:)                !Dummy variable for iterations
REAL, ALLOCATABLE :: Ttab(:),htab(:),Prtab(:),utab(:),vrtab(:),sotab(:)          !Table variables
```

```
REAL :: trp
                                                    !Turbine stage ratio for expansion
REAL :: turb_int_quot_in                            !Interpolation quotient to find inlet enthalpy
REAL :: turb_int_quot_out                           !Interpolation quotient to find outlet enthalpy
!REAL :: h_turb_in
                              !Turbine inlet enthalpy
REAL :: h_turb_out
                              !Turbine outlet enthalpy
INTEGER :: i,g,m
                              !Indices for iterations, allocation
REAL :: cp,cv
                                                    !Specific heats (isobaric and isochoric)
REAL :: kin
                                                    !Calculated ratio of specific heats for inlet temp
REAL :: kout
                                                    !Calculated ratio of specific heats for outlet temp
REAL :: kavg, kavgold
                              !Average ratio values
REAL :: kJpsreq
                                   !Energy output per stage at given mass flow required to recup energy
expended
REAL :: Pr_quot_out, Pr_out
REAL :: Pr_quot_in, Pr_in
REAL :: T_sin
                                                    !Inlet temp given eta_t=1
INTEGER :: sta
                                                    !Integer value of tstages

!-------------------------------------------------------------------------------

!!VERSION REWRITTEN FOR STATIC TURBINE FLOW AND DYNAMIC EXPANSION PERIOD

!MAY NEED TO BE OPTIMIZED FOR REHEAT AVAILABLITY >>>> FIND STAGES NEEDED (REHEAT TEMPS ARE VERY
HIGH!!!)

!THIS VERSION USES THE ISENTROPIC COMPRESSION/EXPANSION EQUATIONS TO FIND THE INLET AND OUTLET
TEMPERATURES,
!THEN MATCHES THESE TEMPERATURES WITH TABLULAR ENTHALPY DATA

!THE k VALUES USED ARE AVERAGED BETWEEN THE INLET AND OUTLET CONDITIONS (ITERATIVELY)

!--------------------------------FILE OPERATIONS--------------------------------

!OPEN(UNIT=12,FILE='results300K.dat',STATUS='OLD',ACTION='WRITE',POSITION='APPEND',IOSTAT=ioerr)!Open & Append
OUTPUT FILE
!IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

103 FORMAT(A,2X,F10.3,1X,A)
!104 FORMAT(A,/A,2X,F10.3,1X,A)


OPEN(UNIT=9,FILE='air_data_table_T_h_Pr_u_vr_so.dat',STATUS='OLD',ACTION='READ',IOSTAT=ioerr)!Open & Read INPUT
FILE
IF(ioerr /= 0)STOP'The file operation did not complete successfully. Please check file(s) and start again.'

105 FORMAT(F8.0,4X,F9.2,4X,F10.4,4X,F9.3,4X,F10.3,4X,F12.5)

m=121

ALLOCATE(Ttab(m+1),htab(m+1),Prtab(m+1),utab(m+1),vrtab(m+1),sotab(m+1))

DO i=1,m

READ(9,*)Ttab(i),htab(i),Prtab(i),utab(i),vrtab(i),sotab(i)
```

```
END DO

CLOSE(9)

!-------------------------------------------------------------------------------

!WRITE(*,'(A,2X,F5.2,1X,A)')'                    You entered',eta_t_percent,'%'
!WRITE(*,*)''

eta_t=eta_t_percent/100

sta=INT(tstages)

WRITE(*,*)sta,'is the sta value and should be',tstages

!WRITE(*,'(A,2X,I3,1X,A)')'          You entered',sta,'Turbine/Heat Exchanger Couple Sets'
!WRITE(*,*)''

trp=(P_in/P_low)**(1/tstages)                          !Calculate Stage Ratio -- Assumes exit pressure is ambient pressure

!WRITE(*,*)'                              The turbine stage ratio is',trp
!WRITE(*,*)''

ALLOCATE(P_int(sta))
                                              !Allocate intermediate pressures vector

!WRITE(*,'(A,2X,F7.3,1X,A)')'                    You entered',m_dot_xpd,'kg/s'
!WRITE(*,*)''

t_xpd=del_m/m_dot_xpd
                    !Calculate maximum time for expansion [seconds] @ constant tank temp

TIME=t_xpd/3600
                                                                         !Convert time for expansion to hours


ALLOCATE(P_max(sta+1))
                                                !Allocate dummy variable vector

P_max(1)=P_low
                                                                                   !Initialize P_max(1)

DO i=1,sta

P_int(i)=P_max(i)*trp
!WRITE(*,'(A,2X,I3,2X,A,2X,F7.3)')'       The outlet pressure of turbine',i,'is',P_int(i)
!WRITE(*,*)''
!WRITE(12,'(A,2X,I3,2X,A,2X,F15.3)')'The outlet pressure of turbine',i,'is',P_int(i)
P_max(i+1)=P_int(i)

END DO

T_h_out=300

Ttab(0)=0
htab(0)=0
Prtab(0)=0
utab(0)=0
vrtab(0)=0
sotab(0)=0

DO i=1,m

Pr_quot_out=(T_h_out-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
Pr_out=Prtab(i-1)+(Pr_quot_out*(Prtab(i)-Prtab(i-1)))
```

```
IF (Ttab(i) > T_h_out)EXIT
END DO

!WRITE(*,*)''
!WRITE(*,*)'                                    Pr_out is',Pr_out,'at',T_h_out,'K'
!WRITE(*,*)''

Pr_in=Pr_out/trp

DO i=1,m

Pr_quot_in=(Pr_in-Prtab(i-1))/(Prtab(i)-Prtab(i-1))
T_sin=Ttab(i-1)+(Pr_quot_in*(Ttab(i)-Ttab(i-1)))

IF(Prtab(i) > Pr_in)EXIT

END DO

!WRITE(*,*)''
!WRITE(*,*)'                                      Pr_in is',Pr_in,'at',T_sin,'K'
!WRITE(*,*)''


CALL specific_heat_calc (T_h_out, R_gas, cp, cv, kout)
                                !Call the specific heat SUB again for the outlet temp

CALL specific_heat_calc (T_sin, R_gas, cp, cv, kin)
                                  !Call the specific heat SUB for the inlet temp

kavg=(kin+kout)/2                                                    !Average the values

kJpsreq=(((kavg*R_gas)/(kavg-1))*((trp**((kavg-1)/kavg))-1))*T_sin

T_hxpd=T_h_out+((T_sin-T_h_out)/eta_t)

!WRITE(*,*)''

!WRITE(*,*)'      DEBUG: The power out per stage should be approximately',ABS(kJpsreq*m_dot_xpd),'kW'

!WRITE(*,*)'DEBUG: The total power out for the system should be approximately',ABS(kJpsreq*m_dot_xpd*tstages),'kW'

!WRITE(*,*)'DEBUG: The total energy out for the system should be approximately',ABS(kJpsreq*m_dot_xpd*tstages*TIME),'kWh'

!WRITE(*,*)''

!------------------------LOOK UP ENTHALPY PER STAGE-----------------------------

110 FORMAT(A,2X,F5.0,1X,F6.2,1X,/A,2X,F5.0,1X,F6.2)

!!WRITE(*,*)'Here are the inlet temps for the enthalpy calcs!'

DO i=1,m

turb_int_quot_in=(T_hxpd-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
h_turb_in=htab(i-1)+(turb_int_quot_in*(htab(i)-htab(i-1)))

!                             WRITE(*,110)'The low temp/enthalpy for the outlet is'&
!                   &,Ttab(i-1),htab(i-1),'and the high temp/enthalpy for the outlet is',Ttab(i),htab(i)

IF (Ttab(i) > T_hxpd)EXIT
END DO

!WRITE(*,*)'                                     turb_int_quot_in     h_turb_in'
!WRITE(*,*)''
```

```
!WRITE(*,*)'                                        ',turb_int_quot_in,h_turb_in

!!WRITE(*,*)'Here are the outlet temps for the enthalpy calcs!'

DO i=1,m

turb_int_quot_out=(T_h_out-Ttab(i-1))/(Ttab(i)-Ttab(i-1))
h_turb_out=htab(i-1)+(turb_int_quot_out*(htab(i)-htab(i-1)))

!                                       WRITE(*,110)'The low temp/enthalpy for the inlet is'&
!                               &,Ttab(i-1),htab(i-1),'and the high temp/enthalpy for the inlet is',Ttab(i),htab(i)

IF (Ttab(i) > T_h_out)EXIT

END DO

!WRITE(*,*)''
!WRITE(*,*)'                                       turb_int_quot_out    h_turb_out'
!WRITE(*,*)''
!WRITE(*,*)'                                       ',turb_int_quot_out,h_turb_out
!WRITE(*,*)''

!-------------------------------------------------------------------------------

!del_h_hot = eta_t*R_gas*(((a)*(T_h_out-T_hxpd))+((b/2)*((T_h_out**2)-(T_hxpd**2)))+((c/3)*((T_h_out**3)-&
!                       &(T_hxpd**3)))+((d/4)*((T_h_out**4)-(T_hxpd**4)))+((e/5)*((T_h_out**5)-(T_hxpd**5))))

del_h_hot=h_turb_in-h_turb_out                                          !Calculate the enthalpy change for each stage

W_hot=m_dot_xpd*del_h_hot*TIME*eta_t                    !Expansion energy out per stage [kg/s * kJ/kg * h * eff = kWh]

!I WOULD LIKE TO CHANGE THE BELOW EQUATIONS TO BE BASED ON TABLE DATA  >>> DID IT, FOUND LESS PRECISE
BUT SAME APPROXIMATE ANSWER

!FIND CHANGE IN ENTHALPY NEEDED PER STAGE, THEN MAX ENTHALPY NEEDED, THEN

!CORRESPOND THE TEMPERATURE TO THE CHANGE IN ENTHALPY FROM THE STEADY (COLD) STATE

!DO WHILE(ABS(W_hot) < ABS(W_out_tstage))                                            !Change in enthalpy equation

!                                       T_h_out=T_hxpd*((P_int(1)/P_low)**((k-1)/k))   !Calculate hot isentropic expansion outlet
temperature

!       del_h_hot = eta_t*R_gas*(((a)*(T_h_out-T_hxpd))+((b/2)*((T_h_out**2)-(T_hxpd**2)))+((c/3)*((T_h_out**3)-&
!                       &(T_hxpd**3)))+((d/4)*((T_h_out**4)-(T_hxpd**4)))+((e/5)*((T_h_out**5)-(T_hxpd**5))))
!Calculate change in enthalpy for hot expansion (IG) FOR EACH TURBINE

!                                       W_hot=m_dot_xpd*del_h_hot*TIME
                                       !Calculate energy out from hot expansion

!                                       IF(ABS(W_hot) >= ABS(W_out_tstage))EXIT!Exit when goal is met, otherwise inlet temp
increases

!                                               T_hxpd=T_hxpd+1

                                        !Switch with the dummy variable

!!WRITE(*,'(A,2X,F10.3)')'The hot inlet temperature is now',T_hxpd
                                       !TO CHECK REHEAT ITERATION
!!WRITE(*,'(A,2X,F10.3,1X,A,2X,F10.3,1X,A)')'The total energy out is',W_hot,'kJ'&
!

                                &'and the total enthalpy change is',del_h_hot,'kJ/kg'
!END DO
```

```
!del_h_hot=del_h_hot*tstages
                                        !Adjust change in enthalpy for number of stages <FOR ALL TURBINES>

W_hot=W_hot*tstages
                                        !Adjust work (energy) output for number of stages <FOR ALL
TURBINES>

kWout=W_hot/TIME
                                        !Average power out

!WRITE(*,*)'',sta,'is the sta value'
WRITE(*,*)'',tstages,'is the t_stages value'
WRITE(*,*)'',m_dot_xpd,'is the m_dot_xpd value'
WRITE(*,*)'',del_h_hot,'is the del_h_hot value'
WRITE(*,*)'',eta_t,'is the eta_t value'
WRITE(*,*)'',TIME,'is the TIME value'
WRITE(*,*)'',W_hot,'is the W_hot value'
WRITE(*,*)'',kWout,'kW is the turbine powertrain power out'
WRITE(*,*)'',T_hxpd,'K is the required turbine inlet temp and'
WRITE(*,*)'',T_h_out,'K is the turbine outlet temp for',sta,'turbine stages'
WRITE(*,*)'The air exits the turbine array with a specific enthalpy of',h_turb_out,'kJ/kg'

!WRITE(*,103)'                    The power output of the compressor train is:',kWout,'[kW]'
!WRITE(*,*)''

!-------------------------------------------------------------------------------

!WRITE(12,103)'The Isentropic Turbine Efficiency is:',eta_t_percent,'%'
!WRITE(12,*)''

!WRITE(*,103)'                 The expansion process may occur for a period up to:',TIME,'hours'
!WRITE(*,*)''
!WRITE(12,103)'The expansion process may occur for a period up to:',TIME,'hours'
!WRITE(12,*)''

!WRITE(*,103)'          at the designated mass flow rate for the expansion process of:',m_dot_xpd,'[kg/s]'
!WRITE(*,*)''
!WRITE(12,103)'at the designated mass flow rate for the expansion process of:',m_dot_xpd,'[kg/s]'
!WRITE(12,*)''

!WRITE(*,103)'               The change in enthalpy per stage for hot expansion is:',del_h_hot,'[kJ/kg]'
!WRITE(*,*)''
!WRITE(12,103)'The change in enthalpy per stage for hot expansion is:',del_h_hot,'[kJ/kg]'
!WRITE(12,*)''

!WRITE(*,103)'                 The energy produced from hot expansion is',ABS(W_hot),'[kWh]'
!WRITE(*,*)''
!WRITE(12,103)'The energy produced from hot expansion is',ABS(W_hot),'[kWh]'
!WRITE(12,*)''

!WRITE(*,103)'                 This will be',(ABS(W_hot)*100)/W_dot_sum,'% of the input energy'
!WRITE(*,*)''
!WRITE(12,103)'This will be',(ABS(W_hot)*100)/W_dot_sum,'% of the input energy'
!WRITE(12,*)''

!WRITE(*,103)'  The hot expansion air will have to be reheated to',T_hxpd,'[K] before each stage in order for the',&
!
                                        &'    outlet temperature and pressure to return to ambient conditions'
!WRITE(*,*)''
!WRITE(12,103)'The hot expansion air will have to be reheated to',T_hxpd,'[K] before each stage in order for the',&
!
                                        &'outlet temperature and pressure to return to ambient conditions'
!WRITE(12,*)''

!WRITE(*,103)'       The (outlet) temperature of the expanded air for hot isentropic expansion is:',T_h_out,'[K]'
```

```
!WRITE(*,*)''
!WRITE(12,103)'The (outlet) temperature of the expanded air for hot isentropic expansion is:',T_h_out,'[K]'
!WRITE(12,*)''

!WRITE(*,103)'                  In units that real people use, this is a reheat temp of',(T_hxpd*1.8)-459.67,'F'
!WRITE(*,*)''
!WRITE(12,103)'In units that real people use, this is a reheat temp of',(T_hxpd*1.8)-459.67,'F'
!WRITE(12,*)''

!WRITE(*,103)'                                  and an a outlet temp of',(T_h_out*1.8)-459.67,'F'
!WRITE(*,*)''
!WRITE(12,103)'and an a outlet temp of',(T_h_out*1.8)-459.67,'F'
!WRITE(12,*)''

!WRITE(*,103)'                               This is a temperature gradient of',(T_hxpd-T_h_out),'K'
!WRITE(*,*)''
!WRITE(12,103)'This is a temperature gradient of',(T_hxpd-T_h_out),'K'
!WRITE(12,*)''

!WRITE(*,103)'                                              or',((T_hxpd-T_h_out)*1.8),'F'
!WRITE(*,*)''
!WRITE(12,103)'or',((T_hxpd-T_h_out)*1.8),'F'
!WRITE(12,*)''

!CLOSE(12)

DEALLOCATE(P_int,P_max)

DEALLOCATE(Ttab,htab,Prtab,utab,vrtab,sotab)

CLOSE(12)

END SUBROUTINE turbine

!----------------------------------------------------------------------------------------------------------------------------------------------------------------
```