NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGN OF A REGENERATIVELY COOLED BI-PROPELLANT ROCKET ENGINE USING ADDITIVE MANUFACTURING

by

Morgan C. Trent

June 2020

Thesis Advisor: Christopher M. Brophy
Second Reader: James H. Newman

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
DESIGN OF A REGENERATIVELY COOLED BI-PROPELLANT ROCKET ENGINE USING ADDITIVE MANUFACTURING

5. FUNDING NUMBERS

6. AUTHOR(S) Morgan C. Trent

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Empirical heat transfer values and thermodynamic models were verified and expanded for a uni-element film-cooled liquid rocket engine operating on a gaseous oxygen and RP-1 mixture. This effort was motivated by the likely reduction of the overall engine mass by integrating regenerative cooling channels directly into the combustion chamber and nozzle walls through the use of additive manufacturing. The data was collected for a range of operating conditions from 1.77 to 2.29 oxidizer-to-fuel mass mixture ratio and 9.65% to 19.69% film cooling. The combustion chamber of the engine experienced damage at heat flux values of 4.54 MW/m^2 that occurred at a chamber pressure of 6.96 MPa, a mixture ratio of 2.0, and 9.59% film cooling. The data collected was used with computational tools to develop a novel integrated chamber-nozzle engine design for both regenerative cooling and film cooling conditions. The final design possessed less than 6% variation of flow though the 18 regenerative liner passages and was predicted to be able to handle the current and expected heat transfer values. The unit was printed using Stainless Steel 17-4PH with additive manufacturing techniques but will need to be qualified with future open and closed-loop testing to evaluate the delivered regenerative cooling effectiveness.

14. SUBJECT TERMS rocket, engine, liquid cooling, propulsion, additive manufacturing	**15. NUMBER OF PAGES** 133
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

DESIGN OF A REGENERATIVELY COOLED BI-PROPELLANT ROCKET ENGINE USING ADDITIVE MANUFACTURING

Morgan C. Trent
Captain, United States Marine Corps
BSE, Case Western Reserve University, 2014

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
June 2020

Approved by: Christopher M. Brophy
Advisor

James H. Newman
Second Reader

James H. Newman
Chair, Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Empirical heat transfer values and thermodynamic models were verified and expanded for a uni-element film-cooled liquid rocket engine operating on a gaseous oxygen and RP-1 mixture. This effort was motivated by the likely reduction of the overall engine mass by integrating regenerative cooling channels directly into the combustion chamber and nozzle walls through the use of additive manufacturing. The data was collected for a range of operating conditions from 1.77 to 2.29 oxidizer-to-fuel mass mixture ratio and 9.65% to 19.69% film cooling. The combustion chamber of the engine experienced damage at heat flux values of 4.54 MW/m^2 that occurred at a chamber pressure of 6.96 MPa, a mixture ratio of 2.0, and 9.59% film cooling. The data collected was used with computational tools to develop a novel integrated chamber-nozzle engine design for both regenerative cooling and film cooling conditions. The final design possessed less than 6% variation of flow though the 18 regenerative liner passages and was predicted to be able to handle the current and expected heat transfer values. The unit was printed using Stainless Steel 17-4PH with additive manufacturing techniques but will need to be qualified with future open and closed-loop testing to evaluate the delivered regenerative cooling effectiveness.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AM	Additive Manufacturing
BMD	Bound Metal Deposition
CFD	Computational Fluid Dynamics
Cu-101	Copper alloy C101
GOX	Gaseous Oxygen
GUI	Graphical User Interface
LPRE	Liquid Propulsion Rocket Engine
RP-1	Rocket Propellant-1
SOP	Standard Operating Procedure
SS 17–4	Stainless Steel 17–4 Variant
SS304	Stainless Steel–Grade 304
SSAG	Space Systems Academic Group
TC	Thrust Chamber

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I need to give an immense thank you to my wife, Katherine, the most driven, dedicated, hard-working person I know. Every day she teaches me that grit and creativity can go a long way in achieving goals. She also exhibited endless amounts of patience while I was up late running simulations and writing, and offered gentle reminders to shower when I came home smelling of RP-1. I would not have been nearly as successful in this endeavor without her.

I'd also like to thank Alexis Theony, Dr. Josh Codoni, and Birger Neick. Alexis is a walking, breathing (and deadlifting) database of all things practical engineering and was a big contributor in the development of the new engine design. Dr. Codoni served as a great sounding board for all ideas good and bad (and wasn't afraid to let me know when they were bad) and constantly encouraged me to push limits. Mr. Neick, the NPS Physics Department's resident metal 3-D printing "sachverständige," offered a wealth of knowledge on the 3D printer suite and was pivotal in turning the engine vision into reality.

Another huge thank you goes to Dr. Brophy as my thesis advisor. From day one, Dr. Brophy gave me ownership of the rocket hardware, tools, and workspaces. He offered patience and suggestions when things went wrong, as well as a steady guiding hand throughout this entire project. Dr. Brophy has vastly increased my knowledge of all things aeronautical both in the lab and during our desert rocket launches.

Finally, I would like to thank the Marine Corps Training and Education Command for giving Marines like myself the opportunity to learn at an institution such as NPS. I look forward to bringing this knowledge back to the fleet and making the Marine Corps a more lethal fighting force.

There are so many more people not listed here that were a huge influence. To the students I struggled alongside, thank you for offering constant entertainment as we embraced the suck. And to the rest of my family, thank you for your unwavering enthusiasm and support.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The Space Systems Academic Group (SSAG) and Mechanical and Aerospace Engineering Department share a common goal of building and launching payloads to altitudes exceeding 300,000 ft under both ballistic flight conditions as well as those including guidance and control. This will allow users to meet specific trajectory requirements and tailor flight profiles for more sensitive devices to include: communication payloads such as software defined radio CubeSats, the SSAG Terrahertz Imaging Camera, and other earth imaging hardware.

This thesis is a continuation of research conducted by Ensign Zachary Lewis, who characterized the film cooling heat transfer properties of thrust chamber liners fabricated using different metal alloys and wall thicknesses in a uni-element rocket engine with a water-cooled jacket design [1].

A. BACKGROUND

Robert H. Goddard of the United States was the first scientist to successfully design, build, and eventually fly a liquid propellant rocket engine (LPRE) in 1926 [2], [3]. Since then, engineers such as Hermann Oberth and Wernher von Braun continued to push the limits and develop and scale LPRE systems to launch larger and larger payloads [3]. There is a myriad of reasons why LPREs are better platforms over solid fueled rocket motors, including increased performance, throttling, and restart capabilities [3]. These performance factors are very important when designing a launch system for sensitive payloads that require a tailored thrust profile, but perhaps the most important factor is the ability of liquid systems to be repeatedly test fired, characterized, and certified on the ground before flight [3]. This is standard practice with all LPREs in production today.

These advantages result in additional challenges. Robust cooling systems are required for liquid rocket engines because the combustion chamber is always exposed to the combustion products and they do not have the inherent insulation properties found in solid rocket motors. Liquid rockets designed to run for any significant or practical duration (more than 3 seconds) need to possess a sufficient cooling method above what can be

offered by heat sinks [2], [3]. The first approach is to simply reduce the amount of heat being transferred to the walls within the combustor. Thus, a film cooling approach may be used. Film cooling is a cooling method in which some fuel is injected along the chamber walls at a low velocity, causing a boundary layer of cooler fluid to flow and protect the inner thrust chamber (TC) wall from direct exposure to high combustion gas temperatures [3]. A film cooling configuration most similar to the one on the left side in Figure 1, with a manifold for film cooling injection partway down the chamber, was used in this thesis' engine setup.

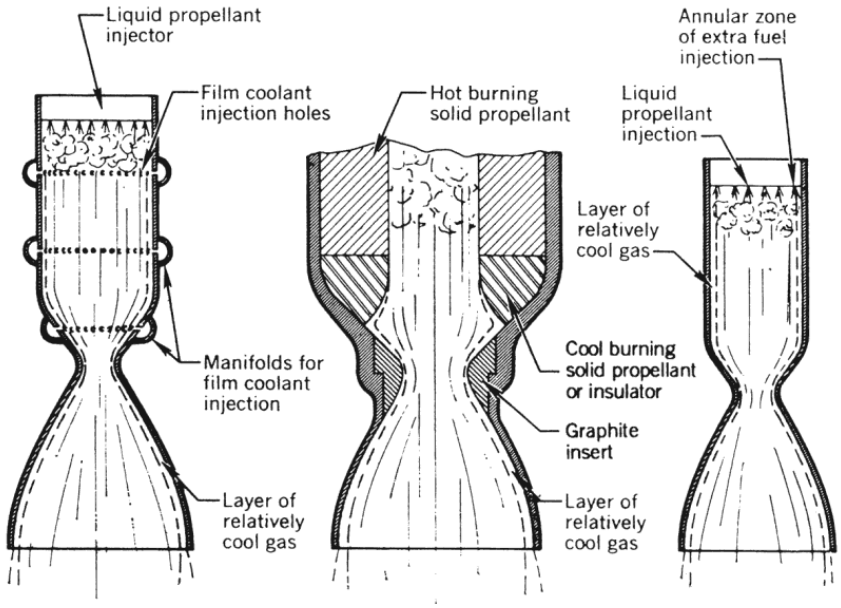


Figure 1. Three Boundary Layer Forming Methods. Source: [2].

While film cooling offers a first approach to dealing with high temperature flows, the film cooling layer itself is carried downstream with the combustion products and thus that fuel does not contribute to the combustion process. Additionally, heat transfer properties for unique designs are still being characterized, defined, and modeled today. This process is complicated because the boundary layer develops with a mixed combustion products-liquid fuel fluid. Because of this, rocket engines that utilize film cooling are difficult to model theoretically.

Due to their high TC temperatures, many liquid rockets utilize film cooling as one of multiple cooling methods. The other cooling method commonly used is regenerative cooling, which takes fuel prior to combustion and circulates it on the outside of the nozzle and thrust chamber to cool the engine [3]. An example of this can be seen in the RD-191 TC schematic in Figure 2.

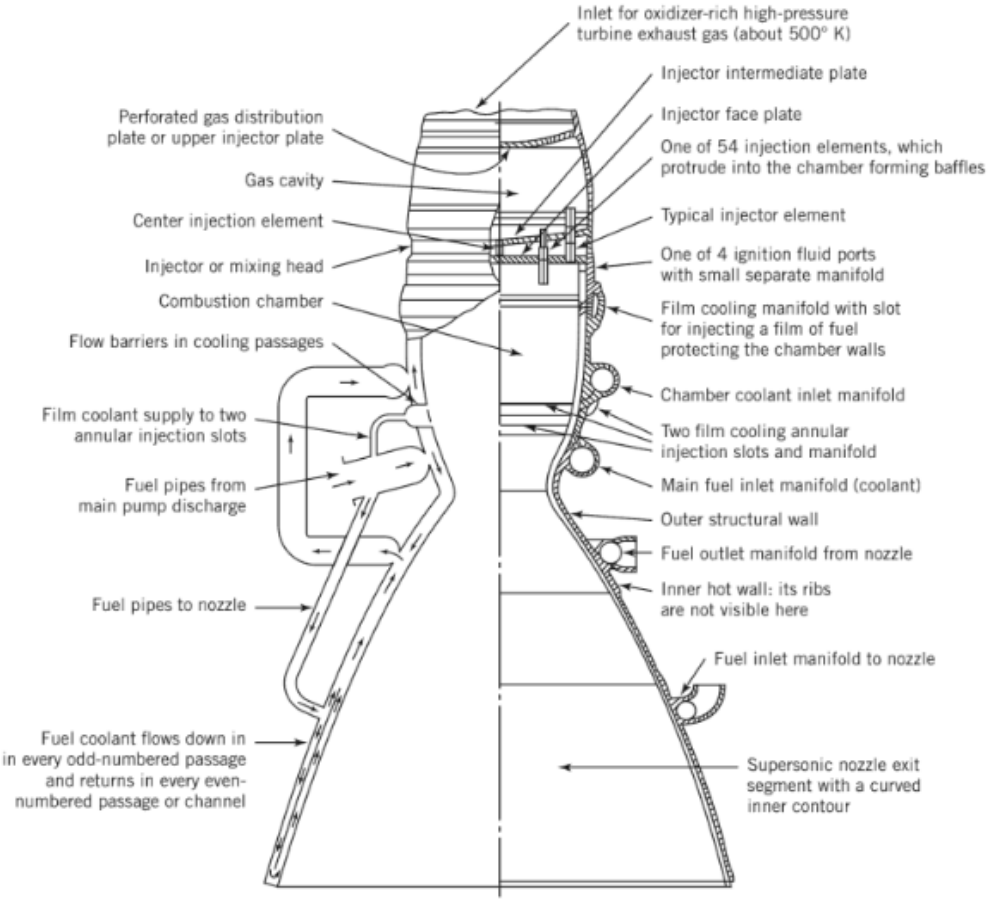


Figure 2. RD-191 Thrust Chamber Showing Regenerative Cooling Paths. Source: [3].

The benefits to regenerative cooling are two-fold. First, film cooling is often inadequate to keep TC walls and nozzles within material operating temperatures. Secondly, it has been shown that there is a performance benefit to using the heat transferred through the wall to “pre-heat” the fuel prior to injection and combustion, thereby creating a nearly

adiabatic “system.” This is discussed in Chapter II.D. Typically, the coolest fuel enters the cooling channels near the base of the nozzle or near the nozzle throat because heat flux is greatest at the nozzle throat as shown in Figure 3. Both strategies are depicted on the RD-191 in Figure 2.

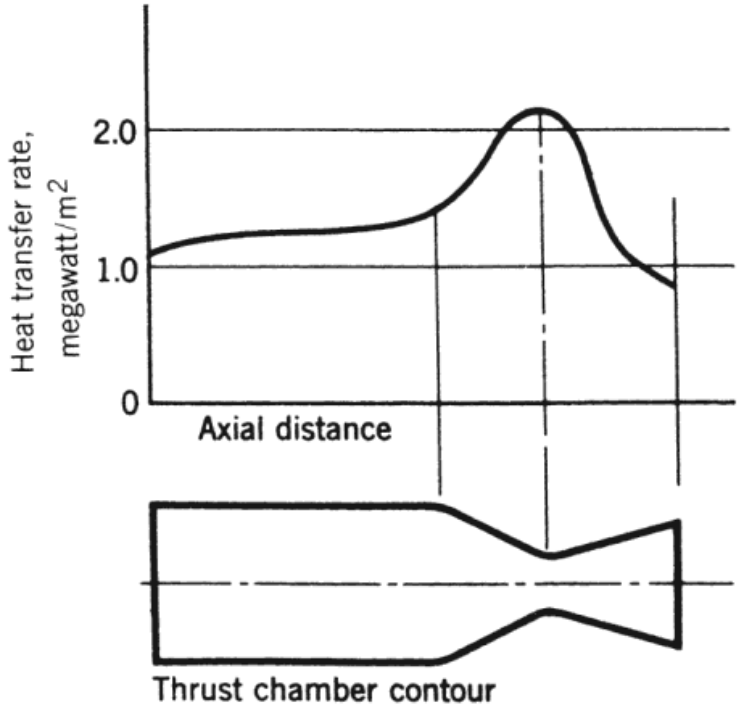


Figure 3. Axial Distribution of Typical Heat Transfer Rates. Source: [2].

Some regenerative cooling passage manufacturing styles are shown in Figure 4. While the primary consideration of existing designs is their ability to be manufactured to maintain structural integrity and overall heat transfer paths, the ability to increase their efficiency, reliability, and reproducibility by utilizing metal additive manufacturing (AM) processes allows complicated internal cooling channel configurations to be printed directly into the thrust chamber and nozzle. This thesis involved designing and printing a Stainless Steel 17–4PH (SS 17–4PH) nozzle with internal cooling channels. Refer to Chapter IV for more detail regarding cooling channels and additive manufacturing, and for further discussion about the design and analysis of the printed nozzle.

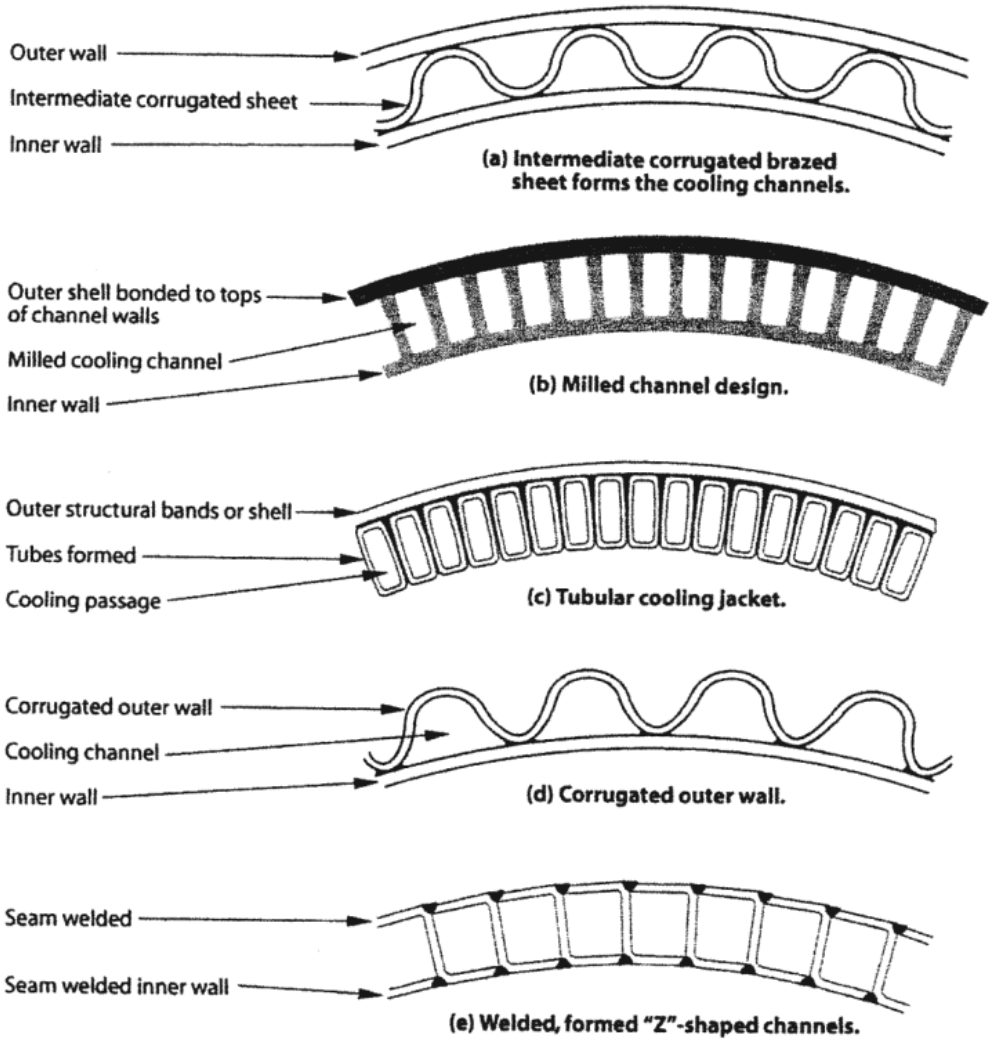


Figure 4. Regenerative Cooling Channel Schemes. Source: [2].

LPRE performance and chamber pressure are directly proportional to engine cooling system complexity and the need for regenerative cooling, as shown in Figure 5. The engine studied in this thesis operated with TC pressures on the order of 3.5 to 7 MPa (500–1000 psia). Figure 5 shows that regenerative cooling alone is an insufficient cooling method for this pressure range [4].

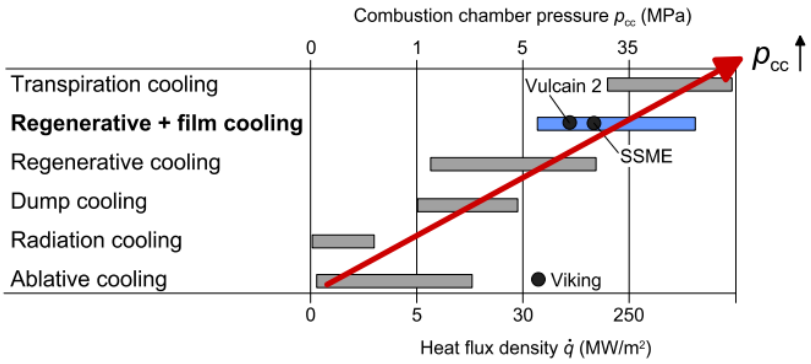


Figure 5. LPRE Cooling Methods and Operating Envelopes. Source: [4].

This thesis studied a high performance uni-element liquid rocket engine with circumferential continuous injection film-cooling. The reactants were Rocket Propellant-1 (RP-1) and Gaseous Oxygen (GOX) and the combustion chamber liners were cooled with a variable flow water bath for maximum flexibility. Little experimental data exists from engines with this specific configuration, size, and chamber pressures, but the primary goal was to evaluate material properties (both AM and conventionally manufactured) and determine the heat transfer model validity. There is also little publicly available data that characterizes heat transfer of 3-D printed liquid rocket thrust chambers of any size, as most testing is on proprietary equipment. The final component of this thesis used additive manufacturing to develop a chamber-nozzle system with built-in regenerative cooling channels for integration into the existing engine test stand.

B. RELATED RESEARCH EFFORTS

Papers by P. R. Grandl of NASA Marshall Space Flight Center and A. Kwas et. al. provide in-depth studies on rapid fabrication techniques and their applications to spacecraft propulsion components such as nozzles with internal cooling channels. Additive manufacturing is discussed and concluded to be a valid technique for building increasingly complicated components with internal channels [5], [6]. However, the studies did not include any test firing of rocket components, nor was data collected on these materials' thermal properties. AM techniques for metals such as steel alloys have improved since these studies were performed and will likely further advance the applications of this technology. The AM

process used in this thesis, Bound Metal Deposition (BMD), is a relatively new technique and is currently proprietary [7]. As such, there is only one study, conducted by A. Watson, which covers the process and the resulting materials' mechanical properties in any depth [8].

Dr. Kirchberger has conducted many studies on rocket engines of a similar scale to the one in this thesis [9], [10]. He studied a uni-element film-cooled rocket engine with various film cooling percentages and similar TC pressures. He also correlated these film cooling percentages to heat flux and wall temperature values with a Jet A-1, GOX mixture. Additionally, his goal was not to optimize the engine for flight or thrust deliverance, thus his rocket operated closer to stoichiometric mixture ratios of 2.8–3.4 [9]–[11]. As discussed in Chapter II.A, mixture ratios near 2.1–2.3 are more ideal for thrust optimization [12].

Dr. Buss et al. also published papers utilizing hydrocarbon fueled rocket engines of similar size to the one in this thesis. Particularly useful are his discussions of film cooling characterization parameters. Like Dr. Kirchberger's engine, however, Dr. Arnold used different fuels than the ones in this thesis- liquid oxygen and gaseous hydrogen- so there are few similarities in the data. Dr. Arnold's papers still offer a valuable reference for film cooling [4], [13], [14].

C. OBJECTIVES

Ultimately, this effort is one step in many towards a flight-weight liquid propellant engine for use with NPS sub-orbital rockets. The success of this portion of the project is measured by these main objectives:

- Expand the existing database by continuing to "collect and correlate heat transfer data for a range of chamber pressure, mixture ratio, % film cooling, and chamber liner alloy composition" [1]. This also includes collecting data for additively manufactured components.
- Design a single integrated chamber-nozzle to be printed from Stainless Steel 17–4PH.
- Determine the possibility of thermally balancing a closed-loop system utilizing regenerative cooling with RP-1.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LIQUID PROPELLANT ROCKET ENGINE THEORY

A. HYDROCARBON FUEL COMBUSTION THEORY

In most LPRE configurations the fuel and oxidizer are stored in separate tanks as shown in Figure 6. Typically, an inert gas under high pressure is used to feed the fuel and oxidizer through an injector into a thrust chamber, where they are mixed for combustion. Adequate mixing is required for an efficient combustion process, which is provided by an appropriate atomizing injector as discussed in Chapter III.A. For this thesis the fuel was RP-1 and the inert pressurant was gaseous nitrogen. The oxidizer was GOX which was fed from its own high-pressure system (unlike Figure 6, which shows a liquid oxidizer fed by the same pressurant).

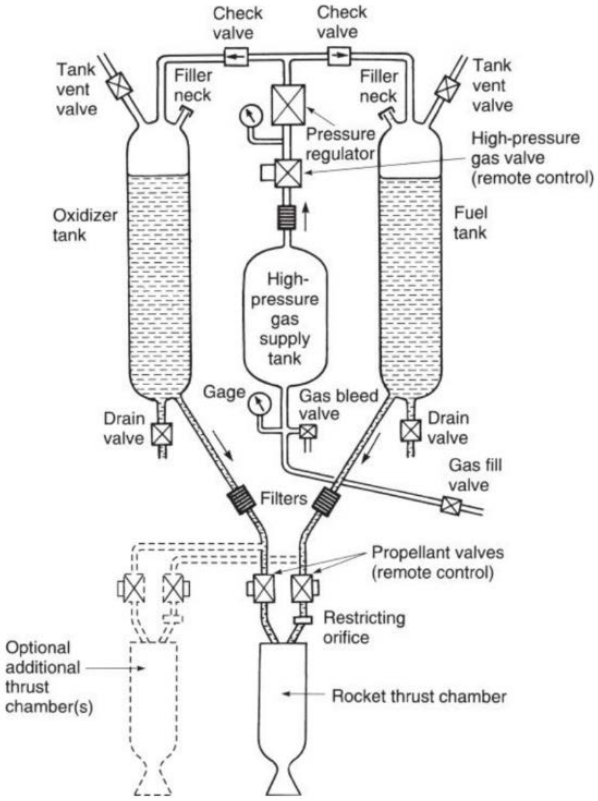


Figure 6. Example of an LPRE with a Gas Pressure-Fed System. Source: [3].

A complete hydrocarbon combustion reaction occurs when the only products are CO_2 and H_2O along with any excess fuel or oxidizer. Similarly, a stoichiometric reaction occurs when there are no excess reactants. RP-1 is a complex hydrocarbon fuel with very low sulfur content designed to be used in liquid form on many types of liquid rockets for multiple reasons: the low sulfur content helps minimize polymerization of fuel within the metallic combustion chamber liners; it is storable and stable at room temperature; and it is a relatively dense hydrocarbon fuel where complete combustion is possible with O_2 [3]. The closest approximation to a chemical formula for RP-1 is $C_{12.6}H_{25.6}$ [15], [16]. Equation 1 is the complete combustion equation for a stoichiometric ratio of RP-1 to GOX. This includes 19 mol of O_2 to one mol of RP-1.

$$C_{12.6}H_{25.6} + 19\ O_2 \rightarrow 12.6\ CO_2 + 12.8\ H_2O \tag{1}$$

In rocketry, this ratio of oxidizer to fuel is referred to as the Mixture Ratio or O/F ratio and is defined as:

$$O/F = \frac{\dot{m}_{ox}}{\dot{m}_f} \tag{2}$$

where $\dot{m}_{ox}$ is the mass flow rate of the oxidizer and $\dot{m}_f$ is the mass flow rate of the fuel, both in kg/s. Therefore, O/F is unitless. When the RP-1/GOX engine for this thesis is operating at its stoichiometric O/F as shown in Equation 1, the O/F is 3.47 when using a molecular weights of 32 g/mol for O_2 and 175 g/mol for RP-1.

While stoichiometric combustion is understood to maximize temperature due to all reactants combining completely with no excess reactants, this does not necessarily mean that stoichiometric combustion maximizes engine performance. Figure 7 shows the positive relationship between RP-1/GOX O/F and combustion temperature.

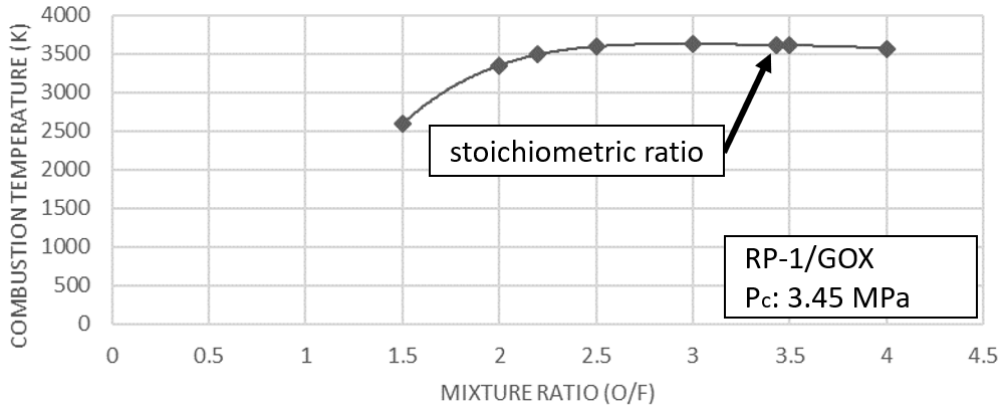


Figure 7. Combustion Temperature vs. O/F for RP-1/GOX. Source: [1].

Figure 7 shows that the maximum temperature is reached closer to an O/F between 2.5 and 3.0 rather than at the expected stoichiometric ratio of 3.43. This is because real combustion occurs more closely to "equilibrium" conditions where secondary combustion species can slightly shift the O/F to a point where the maximum temperature does not occur. This happens because the high combustion temperatures and pressures in the TC cause combustion products to dissociate. Dissociation of the combustion products uses energy and lowers the overall combustion temperature. It is therefore necessary to consider the difference between stoichiometric and equilibrium when designing an operating condition for a rocket engine.

B. ROCKET ENGINE PERFORMANCE PARAMETERS

The goal of most rockets is to get payloads of significant mass moving at a significant velocity; either to deliver a warhead to a target in a timely manner, or to put objects into orbit. This requires positive acceleration; thus, it is important to have the ratio of the rocket engine thrust to fueled (or "wet") vehicle weight to be greater than one. This ratio, known at the thrust-to-weight ratio, F/w_0, "expresses the acceleration (in multiples of the earth's surface acceleration of gravity, g_0) that the engine is capable of giving…" to the loaded launch vehicle [3]. Thrust-to-weight is the reason why rocket engines' weights are minimized.

Thus, while it is important to understand the relationship between O/F and combustion values, the primary metric for optimizing rocket performance and efficiency is thrust produced, not combustion temperature. Sutton and Biblarz define thrust as "the force produced by the rocket propulsion system acting at the vehicle's center of mass. It is the reaction force, experienced by vehicle's [sic] structure from the ejection of propellant at high velocities" [3]. Equation 3 is the total thrust for steady operation in a homogenous atmosphere:

$$F = \dot{m}V_e + (p_e - p_0)A_e \tag{3}$$

The first term on the right-hand side, called "momentum thrust," is given by the product of the propellant (fuel and oxidizer) mass flow rate in kg/s and exhaust exit velocity (relative to the vehicle) in m/s. The second term is referred to as pressure thrust, and is the product of the difference between the exhaust exit pressure and ambient pressure, both in Pa, and the exit nozzle's cross-sectional area in m^2 [3].

Exhaust exit velocity, V_e, in Equation 3 is defined as:

$$V_e = \sqrt{2\left(\frac{\gamma}{\gamma-1}\right)\frac{\overline{R}}{\overline{M}}T_c\left[1-\left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right]} \tag{4}$$

where γ is the ratio of the specific heats of the combustion products (RP-1 and GOX in this case), $\overline{R}$ is the universal gas constant in J/mol-K, $\overline{M}$ is the molecular weight of the combustion products, T_c is the combustion temperature in K, p_e is the nozzle exit pressure in Pa, and p_c is the chamber pressure in Pa [3]. If the engine, reactant flow rates, and rocket flight profile are designed such that chamber and exit pressures and other combustion products' properties are held constant, then exit velocity becomes proportional to the root of the ratio of combustion temperature to molecular mass as shown in Equation 5 [3].

$$V_e \propto \sqrt{\frac{T_c}{\overline{M}}} \tag{5}$$

Examining Equation 5 and plotting $\sqrt{\frac{T_c}{M}}$ as a function of O/F for RP-1/GOX shows that maximizing combustion temperature while minimizing the molecular mass of the products will maximize V_e [1]. This relationship can be seen in Figure 8.

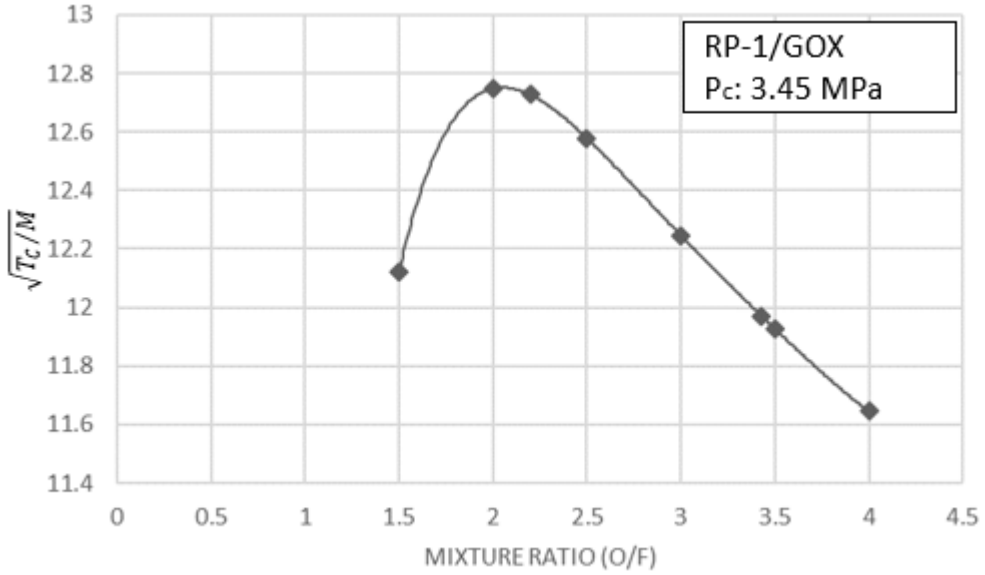


Figure 8. Exit Velocity vs O/F for RP-1/GOX. Source: [1].

A useful value for determining overall system performance, sizing, and for informing fuel selection is ρI_{sp}. Here, ρ is the average density of the reactants (in kg/m^3) and I_{sp}, specific impulse (in seconds), is defined as:

$$I_{sp} = \frac{F}{\dot{m} g_0} \tag{6}$$

ρI_{sp} is typically one of the first metrics calculated when selecting fuels and designing a rocket, and is known as the density-impulse. However, the reactants evaluated in this work were a common mixture with a respectable density-impulse and were selected previously. One of the main objectives of this thesis was to thermally balance the engine, not necessarily optimize the system. Thus, ρI_{sp} is not a required metric for this study.

Another value used to measure rocket engine chamber performance is the characteristic velocity, C^*, defined as:

$$C^* = \sqrt{\frac{1}{\gamma}\left(\frac{\gamma+1}{2}\right)^{\frac{\gamma+1}{\gamma-1}} \frac{\overline{R}}{\overline{M}} T_c} \tag{7}$$

Reference [3] provides more background on the utility of characteristic velocity, but, in general, C^* indicates how a given propellant will generate low molecular weight combustion products with a high combustion temperature (shown in Equation 7 to correspond positively with engine thrust). Equation 7 indicates that, holding all other combustion products' gas properties constant, higher combustion temperatures and lower molecular weights will maximize C^*. Figure 9 shows how C^* changes with increasing O/F for an RP-1/GOX engine.

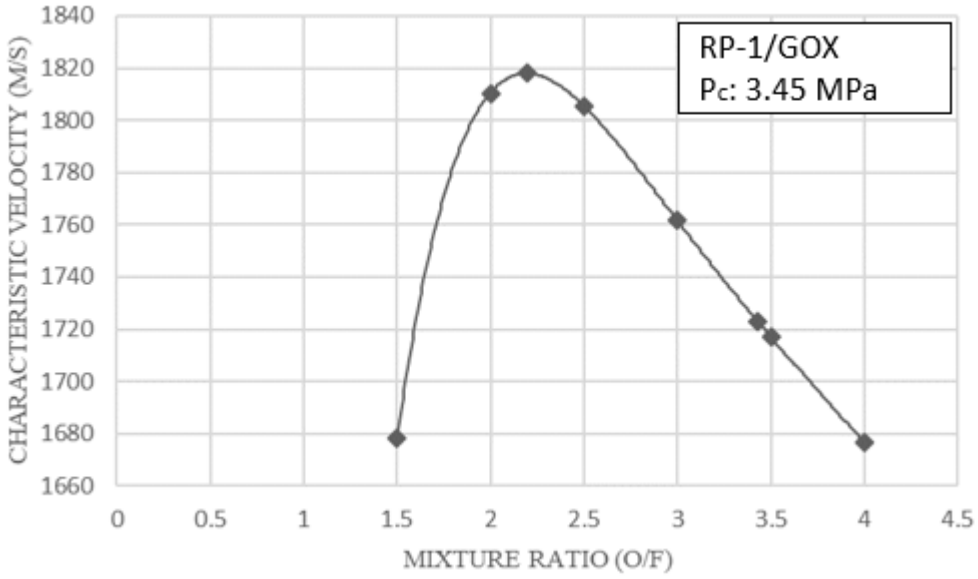


Figure 9. Characteristic Velocity vs. O/F for RP-1/GOX. Source: [1].

Figure 9 indicates that the characteristic velocity, and thus engine performance, is maximized near $C^* = 1819\, m/s$ and an O/F around 2.2. However, this O/F yields combustion temperatures near 3500K as shown in Figure 7. These high temperatures,

coupled with high chamber pressures, are close to the operating limits of the hardware that was used in this thesis and led to failure modes discussed in Chapter V.B when coupled with lower film cooling percentages.

C. HEAT TRANSFER THEORY

Steady-state heat transfer principles and equations can often be applied to liquid rocket engines throughout most of their operation, excluding startup and shutdown. As explained in Sutton's Rocket Propulsion Elements, "steady-state heat transfer through a chamber wall of a liquid-cooled rocket chamber can be treated as a series-resistance-type, steady-state heat transfer problem with a large temperature drop across the wall, and, in cases of cooled chambers, a third temperature drop across the film of the moving cooling fluid" [3]. Figure 10 is the guiding diagram for temperature gradients and heat transfer in a liquid-cooled steady-state rocket engine.

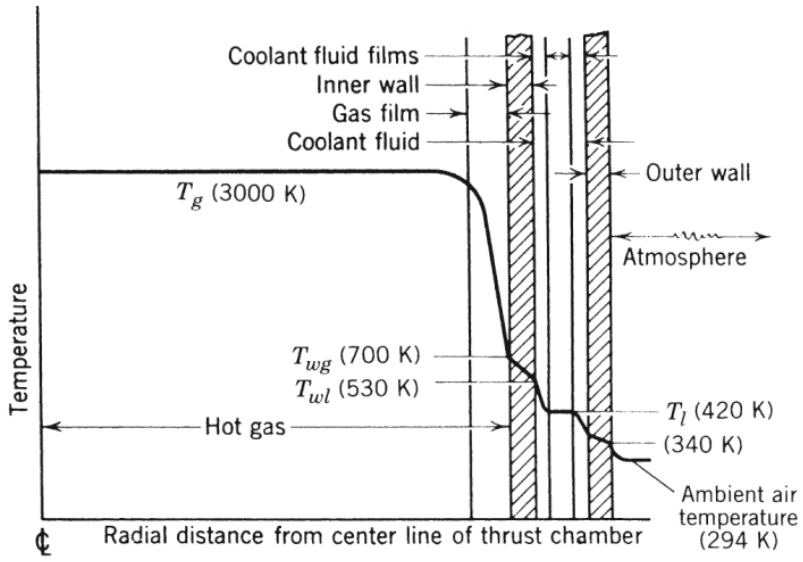


Figure 10. Temperature Gradients in a Cooled Thrust Chamber. Source: [3].

From left to right, the heat from combustion is transferred to the outer wall by way of distinct convection and conduction regions as follows:

1. T_g, the hot gas combustion temperature, is assumed steady and is determined by the chosen fuel and oxidizer.

2. Between T_g and T_{wg} is the gas film layer as discussed in Chapter I.A, representing the largest temperature gradient. The heat flux for this phase is defined in Equation 8. Here, q is the heat flux in W/m^2, h_g is the gas film coefficient in W/m^2-K, and T_g and T_{wg} are the hot gas and inside chamber wall temperatures in Kelvin (3000 K and 700 K in Figure 10) [3].

$$q = h_g(T_g - T_{wg}) \tag{8}$$

3. The next section is the chamber wall. Since the ratio of the rocket TC diameter to the chamber's inner wall thickness is small, this heat flux is typically modeled as 1-dimensional conduction. This heat flux is defined in Equation 9. Here, κ is the thermal conductivity of the chamber material in W/m-K, t_w is the thickness of the chamber wall in meters, and T_{wl} is the temperature of the outer part of the chamber wall in Kelvin (530 K in Figure 10) [3].

$$q = (\frac{\kappa}{t_w})(T_{wg} - T_{wl}) \tag{9}$$

4. Between the inner and outer wall is the cooling fluid, which has three regions: the inner boundary layer, the bulk fluid itself, and the outer boundary layer. The cooling fluid boundary layers are similar to the gas film layer and are also governed in a similar manner to Equation 8. The heat flux for the cooling channel section is defined in Equation 10. T_l is the bulk temperature of the moving cooling fluid in Kelvin (420 K in Figure 10), and h_l is the cooling fluid's film coefficient [3].

$$q = h_l(T_{wl} - T_l) \tag{10}$$

5. Containing the cooling fluid on the outside is the outer wall, whose heat flux is also modeled as 1-dimensional conduction and is defined by Equation 9.

6. Finally, heat is radiated from the outer wall to the atmosphere. This section does not significantly contribute to the chamber's cooling scheme.

Attempts were made to model and correlate much of this cooling scheme to empirical data in Computational Fluid Dynamics (CFD) models as discussed in Chapter V.A.

The cooling fluid mass flux, and thus cooling channel geometry, contributes to the effectiveness of the cooling fluid in transferring heat from the chamber. Various channel configurations were considered for the chamber-nozzle design. The relationship between the fluid's mass flow rate and the overall heat transfer ability is found using enthalpy balance per Equation 11 [3].

$$Q = \dot{m} C_p \Delta T \tag{11}$$

where Q is the heat transfer in W, $\dot{m}$ is the cooling fluid's mass flow rate through the channels in kg/s, C_p is the cooling fluid's heat capacity at a given temperature (420 K in Figure 10) in J/kg-K, and ΔT is the difference in cooling fluid temperature from the beginning to the end of the channels in K.

The channel wall will have an uneven heat distribution with hotter areas directly under the channels that cause thermal stresses in the chamber wall. These temperature gradients are seen in Figure 11.

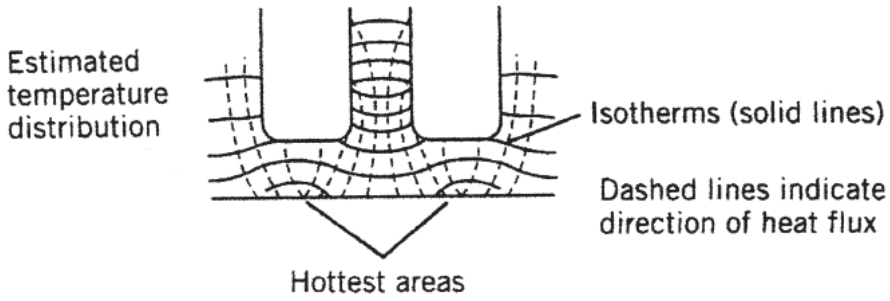


Figure 11. Heat Transfer in Cooling Channels. Source: [3].

Thus , the inner wall should be thin for three reasons: to maximize heat transfer across the wall per Equation 9; to minimize the thermal stresses between channels as illustrated in Figure 11; and to minimize overall engine weight [3].

D. REGENERATIVE COOLING

Regenerative cooling uses fuel to cool the outside of the chamber before being injected into the chamber. Using Equations 2 and 12, the coolant mass flow (or fuel mass flow for a regeneratively cooled system) can be determined if chamber pressure, O/F, nozzle throat area, and characteristic velocity are held constant, as they were during most long burns of this engine. Related to Equation 7, the characteristic velocity can alternatively be determined experimentally by Equation 12:

$$C^* = \frac{P_c A_{th}}{\dot{m}} \tag{12}$$

where P_c is chamber pressure in Pa, A_{th} is the nozzle throat area in m^2, and $\dot{m}$ is the combustion product mass flow rate in kg/s [3]. The fuel's mass flow rate through the cooling channels is matched to its combustion product mass flow rate. It is also factored into Equation 11 to determine its heat transfer and temperature rise. Equation 11 is then divided by the internal surface area of the cooling zone (or cumulative surface area if multiple channels are used) to determine heat flux.

This temperature rise must be limited in regenerative systems because hydrocarbon fuels will decompose when heated beyond certain temperatures- for RP-1 this is around 615K [17], [18]. "RP-1 is low in olefins and aromatics, which can cause carbonaceous

deposits inside fuel cooling passages" [3], but when RP-1 does decompose it exhibits a depositing behavior known as coking. These deposits have very poor thermal conductivity properties and thus raise channel wall temperatures, impede heat transfer, and impede the flow of the cooling fluid [3], [17], [18]. This may lead to the chamber wall experiencing untenable temperatures after any appreciable engine run time. Therefore, if RP-1 is to be used as an effective cooling fluid, its temperature exiting the channels (typically the highest in the system) must remain well below 615K.

A goal of this thesis was to determine the feasibility of a closed-loop, flight weight system utilizing regenerative cooling with RP-1. In order to do this, the heat flux (and thus cooling fluid temperature rise) and chamber wall thickness were minimized.

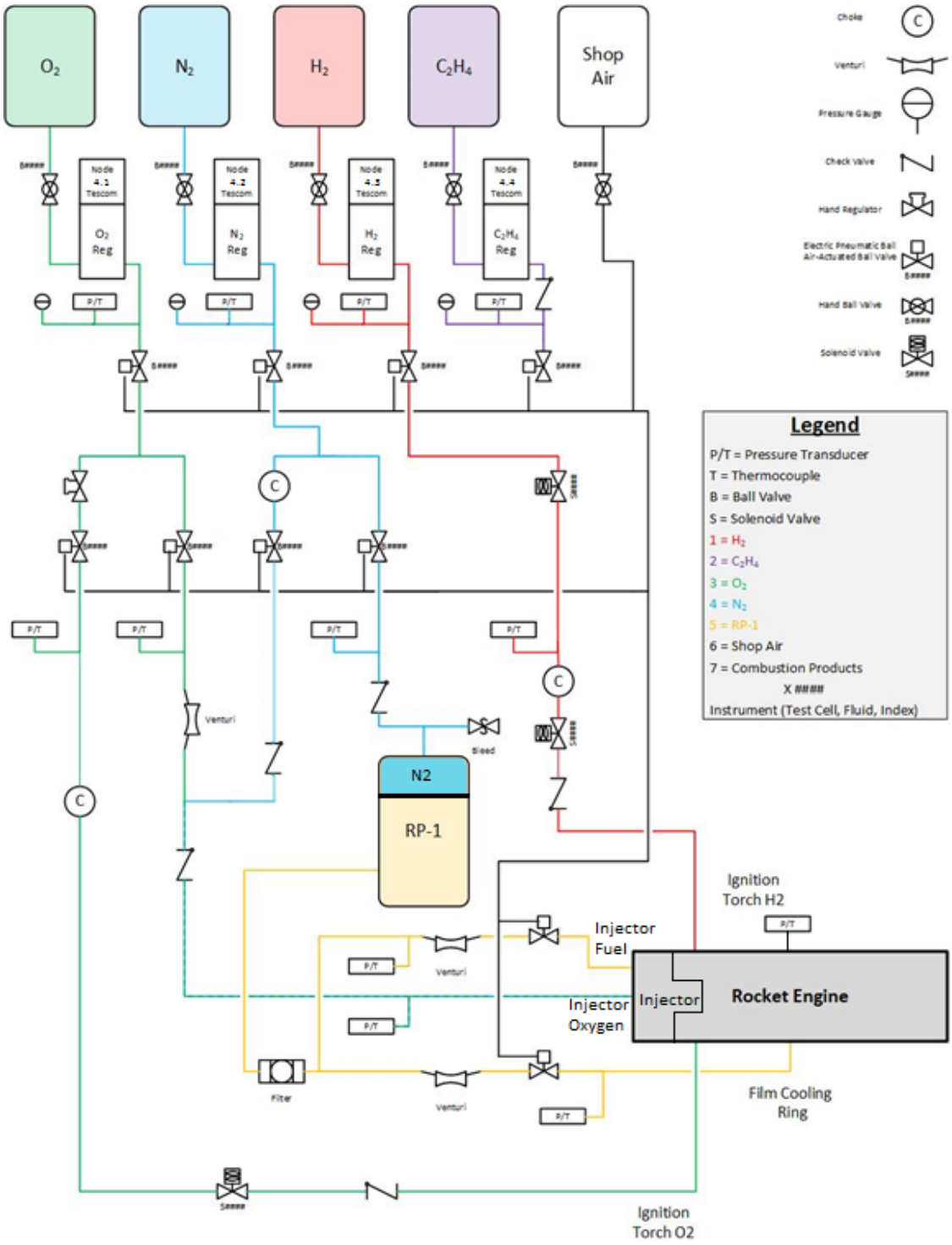
THIS PAGE INTENTIONALLY LEFT BLANK

III. ROCKET ENGINE TESTING

A. EXISTING TEST APPARATUS

The NPS liquid rocket engine is a modular, pressure-fed, water cooled uni-element RP-1/GOX engine with one injector and one film cooling ring. The full test stand assembly is shown in Figure 12. A schematic of the pressure feed system used to supply the engine with gasses and fuel is shown in Figure 13. In this schematic, most components outside of the grey box labeled “Rocket Engine,” excluding some venturis, remained fixed throughout the test campaign, while the engine itself consisted of modular parts as shown in Figure 14.

Figure 12. Liquid Rocket Engine Mounted on Test Stand



The water-cooling system is not included in the schematic. Additionally, C_2H_4 (Ethylene) was not a part of this thesis' testing scheme.

Figure 13. Liquid Rocket Engine Feed System Schematic. Source: [18].

O_2, N_2, and H_2 were stored in tanks a safe distance from the test stand and were supplied to the engine through high pressure lines with electro-pneumatic ball valves powered by 0.689 MPa (100 psia) shop air stored in a separate tank. The red accumulator tank seen on the right in Figure 12 (and the yellow box in the middle of Figure 13) stored the fuel in liquid phase. The fuel was pressurized by N_2 and subsequently driven through filters and venturis to the engine. Venturis were often switched out to meet desired flow rates, and the data analysis code in Appendix A was updated accordingly. O_2 and H_2 were supplied to a standard spark plug to provide ignition. The outside of the engine was cooled by constantly flowing water fed from a tank through a pump operating between 0.552-1.03 MPa (80-150 psia). Due to the amount of film cooling fuel that needed to be drained, the engine was mounted on the stand at approximately 45°. Reference [3] discusses why most liquid rocket engine test stands are mounted upright, but this one was mounted at 45° to balance the fuel draining requirement with ease of repeated assembly and disassembly. Some effects of this angled mounting are discussed in Chapter V.B.

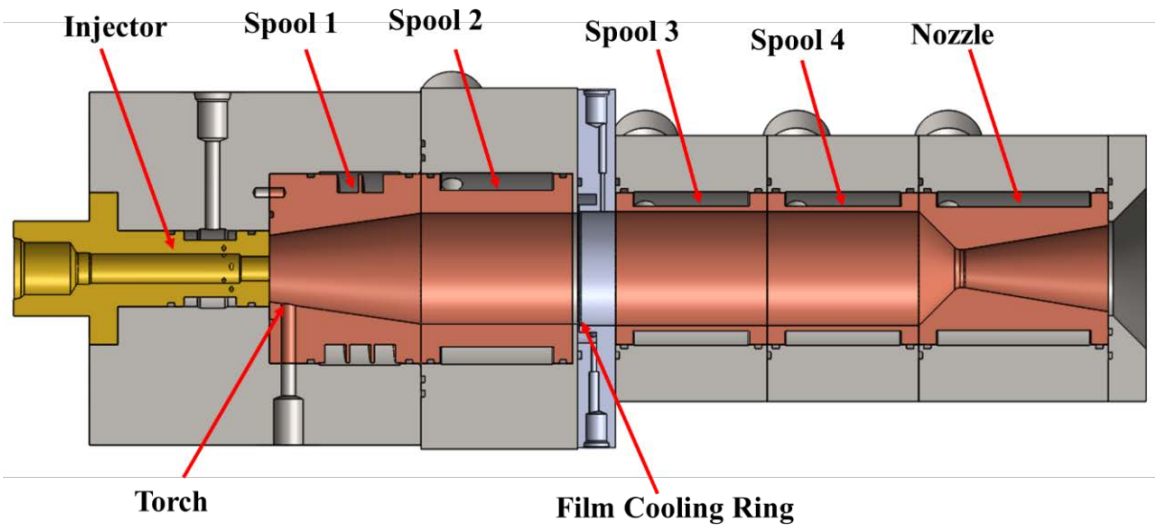


Figure 14. Liquid Rocket Engine Cross-Sectional View. Source: [19].

As seen in Figure 14, there are six modular segments that make up the test engine. Everything between the injector and film cooling ring remained unchanged for all tests.

1. The injector/torch section was where combustion began by introducing a spark to the H_2, O_2 gas mixture in the torch body. A hot flame then initiated combustion of the RP-1/GOX mixture in the main chamber.

2. Spool 1 was a 15° half-angle expansion section which allowed the mixture to expand from the injector head diameter to the full spool diameter.

3. Spool 2 was the first constant-diameter spool, and allowed for a more uniform flow to develop before film cooling was introduced. In a flight-configured system, it would be more useful for the constant-diameter section of the chamber to be downstream of where film cooling is introduced. However, this setup facilitated ease of assembly and disassembly of Spools 3 and 4.

4. The film cooling ring was a continuous injection circumferential design with an injection gap of 0.508 mm (0.020 in) where fuel used for film cooling swirled and accumulated before being injected into the combustion chamber. “The film cooling fuel swirls in the reservoir indicated by the [blue] ring [in Figure 15]. When the reservoir fills with fuel, film cooling fluid beings to enter the engine through the circumferential continuous injection gap” [1]. Figure 16 shows a side view cross section of the film cooling reservoir and 0.508 mm injection gap. It was also important to fully understand the geometry of the cooling ring because this was the part of the engine to which the printed chamber-nozzle system was designed to attach. See Chapter IV for details on this integration process.

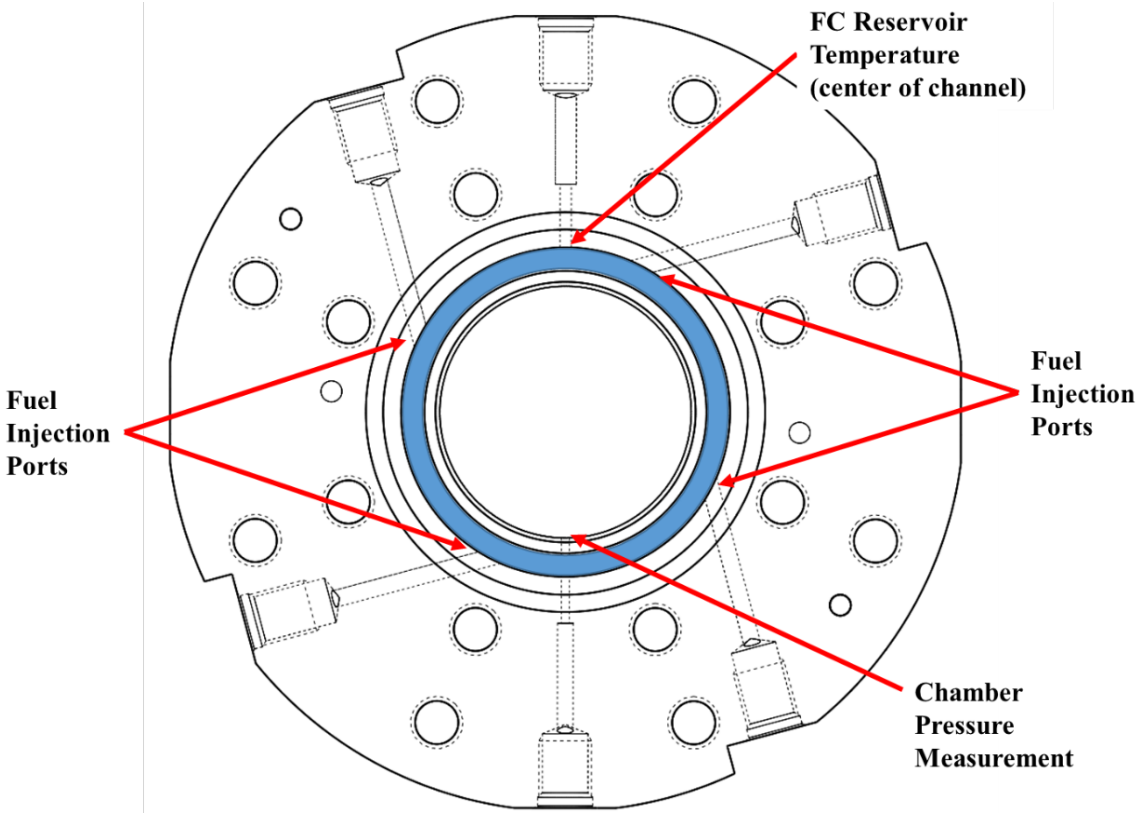


Figure 15. Film Cooling Ring. Adapted from [19].

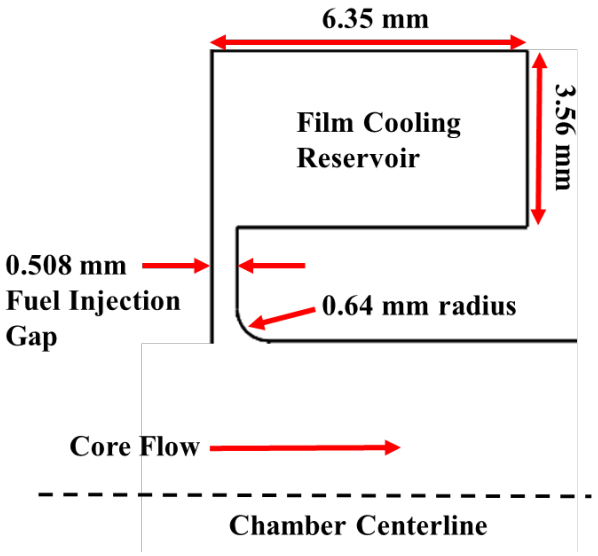


Figure 16. Fuel Film Cooling Injection Gap. Adapted from [19].

5. The inner diameter of the TC (Spools 2, 3, and 4 in Figure 14) was 3.81 cm (1.5 in). Spools 3 and 4 were swapped to collect data on different materials and wall thicknesses throughout this test campaign to gather various test points that were correlated to the sizing tool developed in [1]. See Chapter V.B and Appendix C for more detail on the data collected with these test spools. The materials tested were Copper Alloy C101 (Cu-101), standalone Stainless Steel–Grade 304 (SS304), SS304 with alloy coatings, and additively manufactured SS 17–4PH spools, of varying wall thicknesses from 1.0 mm to 2.54 mm (0.040 in to 0.100 in).

6. The nozzle throat diameters were 1.27 cm and 1.58 cm (0.500 in and 0.625 in) as noted in Chapter V.B and Appendix C. A standard 15° half-angle conical nozzle was used because its behavior is well understood and does not greatly affect or degrade combustion chamber performance. Thrust optimization is out of the scope of this thesis, but for more information see [2] and [3].

See [1] and Appendix B for more detailed information on how pressure, temperature, and flow rate data were collected. Data lines from all thermocouples and pressure transducers were run from the engine test stand to the data acquisition unit, colloquially named R2DQ, depicted in Figure 17.

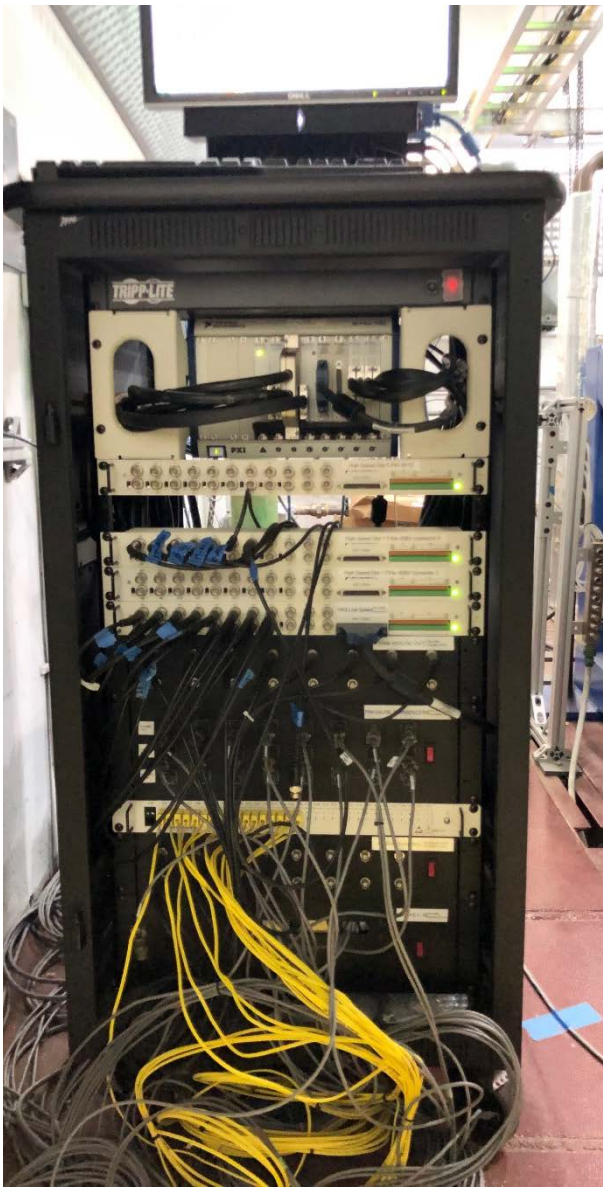


Figure 17. R2DQ Mobile Data Acquisition Unit.

The data signals were fed from the data collection devices on the engine through the yellow (thermocouple) and grey (pressure transducer) wires to R2DQ. LabView code on R2DQ converted the voltages read from the thermocouples and transducers into readable data as shown in Figure 20. This test data was analyzed using the MATLAB code in Appendix A after the run was completed.

B. ENGINE OPERATION

Appendix A lists all standard operating procedures (SOP) for engine assembly, setup, running, shut-down, and disassembly. Once the engine was assembled and attached to the test stand, the water pump was run and the engine was visually checked for any obvious water leaks. A major water leak could severely reduce the engine's cooling ability and lead to overheating and failure of the chamber liners. Every test cycle typically consisted of two or three runs, all following the same sequence in Table 1 with only step 8 ("the burn") varying depending on the run length. The first would be a one-second to two-second burn to check all seals. The last would be the actual test run with a duration between 12 and 30 seconds. Depending on the test run length, another run of medium length was added before the test run to ensure the temperatures leveled out and there were no other failures or anomalies. While all high-speed and low-speed data was collected for the shorter runs that did not reach steady state, post-processing was usually not conducted for these runs.

Table 1. Engine Test Fire Run Sequence. Adapted from [18].

Step	Time Elapsed (sec)	Event
1	0	Start data collection and N_2 purge of O_2 passage though injector
2	3	Open film-cooling ball value and initiate flow through film cooling ring
3	4.5	Open O_2 ball valve and supply O_2 to engine
4	5	Open main fuel ball valve to initiate flow though injector
5	5	Open solenoid valves to ignition torch
6	5.025	Initiate torch spark: engine ignition
7	5.075	Close ignition solenoid valves, close N_2 purge, turn off ignition signal
8	**35.075**	**Run for specified time (30 seconds depicted)**
9	35.075	Close O_2 supply to engine, turn on N_2 purge
10	35.408	Close main fuel and film cooling ball valves
11	38.408	Secure test stand, set supply pressures to zero

This run sequence incorporated multiple N_2 purges designed to ensure safe conditions during startup and shut-down. The first purge occurred during the introduction of O_2 into the system, and ensured a softer ignition and prevented hard starts from occurring. At the end of the programmed run duration (step 9) another N_2 purge extinguished any remaining flames.

Cameras mounted on the lab building and placed near the test stand offered lab personnel many angles of the test runs, and the test engineer could visually monitor the engine on a TV inside the control room. Figure 18 is the view of the camera feed the engineers monitored from inside the control room. Typically, abnormal engine operation was indicated by either excessive smoke or changing exhaust colors (from burning metal engine components). If this happened, the run could be immediately aborted by activating the Emergency-Stop (E-Stop) button at the Engine Control Station. Failure modes that occurred during the longest runs (typically with the lowest film cooling percentages) are discussed in Chapter V.B.

Figure 18. Main Camera View of a Nominal Engine Run

Figure 19. Engine Control Station

Although the computer monitored all temperatures and pressures during a test, the operator kept a hand over the E-Stop button (yellow box with red button to the left in Figure 19) for the duration of the engine run, and was prepared to press it immediately if necessary. This button served as a manual backup to the primary method for engine abort by closing all gas supply valves and stopping all power to cut ignition. The operator also monitored the overall health of the engine, including individual spool temperatures, by way of the LabView Graphical User Interface (GUI) shown in Figure 20. The interface also showed the open/closed status of all gas supply ball valves, gas and water pressures, and chamber pressure. A spike in any of these numbers could also indicate off-nominal operation.

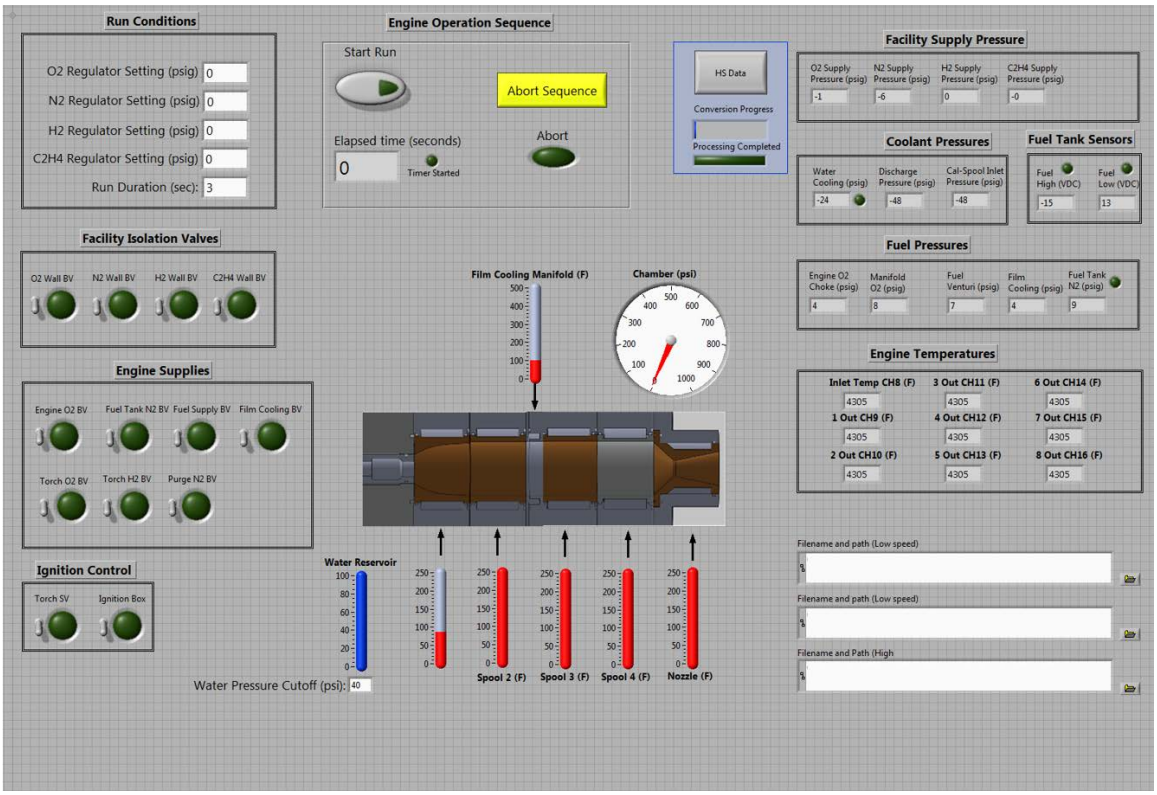


Figure 20. LabView Engine Control Graphical User Interface

The LabView GUI was also where the operator set the desired O_2 and H_2 regulator pressures and run duration as required per the SOP in Appendix A. O_2, H_2, and N_2 pressures were adjusted before each run to reach the target O/F.

Information for all the test runs was stored at the lab in both an electronic spreadsheet and a paper test log. In addition to acting as a paper copy backup, the engine test log served a crucial role during testing. During the testing scheme, theoretical values for run conditions and associated gas driving pressures were calculated using the equations in Chapter II. However, it was impossible to account for all real-world line losses in the pressure-feed system. Thus, the O_2, N_2, and H_2 feed pressures had to be empirically adjusted over time in order to reach desired run conditions. The test log was referred to whenever these pressure settings needed to be adjusted.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ENGINE DESIGN AND BUILDING

An iterative approach was taken to design and build an integrated chamber-nozzle system with internal cooling channels. This was designed to fit onto the existing test stand setup, discussed in Chapter III.A, with the goal of being test fired and analyzed for regenerative heat transfer effectiveness.

The reasons to utilize AM for this build were threefold. First, the available literature suggested that a printed thrust chamber and nozzle utilizing film cooling and regenerative cooling at this scale has never been test fired, and the availability of an NPS-owned Studio-System™ printer suite and trained personnel provided a unique opportunity to do so. Secondly, the quantity of available SS 17–4PH allowed for an iterative design approach, and multiple versions of the nozzle were created. Third, and most importantly, the very nature of AM lends itself to unique designs with geometries such as internal cooling channels that are either difficult or impossible to create by the traditional processes of casting, forging, and machining.

A. SIZING

1. Thermal Management and Material Selection

The hydrocarbon fuel coking limits and chamber material melting points are two thermal management numbers that determine how much heat transfer the engine can tolerate during operation. As discussed in Chapter II.D, RP-1 begins to experience decomposition and coking near 615 K (647 °F) [17]. A factor of safety is applied to account for any combustion spikes, local hot spots, and other anomalies, so the maximum allowable hulk fuel temperature is 544 K (520 °F) [1]. SS 17–4PH has a melting point of 1678K (2561 °F), and with a 20% factor of a safety the hot gas film temperature is limited to 1342 K (1956 °F) [20]. Other materials in this study and their associated melting points and thermal conductivity values are seen in Table 2.

Table 2. Material Properties

Material	Melting Point (K)	Thermal Conductivity (W/m-K)	Source
Cu-101	1311	391	[21]
SS304	1673	16	[22]
SS 17–4PH	1678	18.3	[20]
ZrO_2	2988	2.5	[23]

Three values were used to determine that stainless steel is a better option over Cu-101 for an actively cooled engine: thermal conductivity, density, and yield strength. The cooling demands for a stainless steel chamber are less than that of Cu-101 because stainless steel's thermal conductivity is lower. Thus, "[f]or any location on an engine that does not have sufficient cooling, stainless steel chamber liners will become a heat sink [and likely fail]. Copper on the other hand, has the ability to quickly transport the heat to an area that is properly cooled" [1]. This means that the amount of heat being absorbed by the cooling fluid in the channels will be lower, and the cooling fluid flow rate can be lower as well, which may allow for smaller scale engine designs to be operated in a regenerative mode. In addition to the better thermal properties, stainless steel was also chosen for its density. SS 17–4PH has a density of 7800 kg/m^3 (487 lb/ft^3) [20]. Bound Metal Deposition (BMD) forms a 96–99% dense part [8]. Therefore, the density of the new chamber-nozzle manufactured with BMD is approximately 7605 kg/m^3 (475 lb/ft^3), which is below copper's density of 8920 kg/m^3 (557 lb/ft^3) [21]. Finally, SS 17–4PH has a yield strength of 604 MPa (87.6 ksi) as compared to Cu-101's yield strength of 80 MPa (11.6 ksi) [8], [21]. This means that a chamber manufactured from SS 17–4PH can safely operate with a thinner wall thickness.

2. Previous Engine Sizing Results

Reference [1] developed an engine scaling tool to predict that a thermally balanced system could be achieved with a chamber length of 10 cm (3.9 in) and chamber diameter of 5.227 cm (2.058 in). Table 3 lists the recommended minimum engine dimensions and corresponding parameters.

Table 3. Engine Recommendations for Minimum Size. Source: [1].

Parameter	Value	Comments
Chamber Area Ratio	3.5	
Mach(3.5:1)	0.173	
Spool Length	5	cm
Nozzle Length	6.35	cm
Number of Spools	2	
Cooling Fuel Cp	2020	J/kg-K
Fuel Incoming Temp	311	K
Nozzle Exit Pressure	10	design alt = 3km
Throat Diameter	2.794	cm
Chamber Diameter	5.227	cm
mdotFuel	0.6096	kg/s
mdotTotal	1.9509	kg/s
Thrust	5.778	kN
Heat Flux	10.5	MW/m^2
Hot Gas Wall Temp	499	K
Cooling Fuel Temp Rise	229	K
Final Bulk Fuel Temp	540	K

In [1], the ideal chamber length was determined by measuring hot gas wall temperatures along the length of the chamber, and using those values to compute heat flux. Heat flux values in MW/m^2 for a range of operating conditions can be seen in Figure 21.

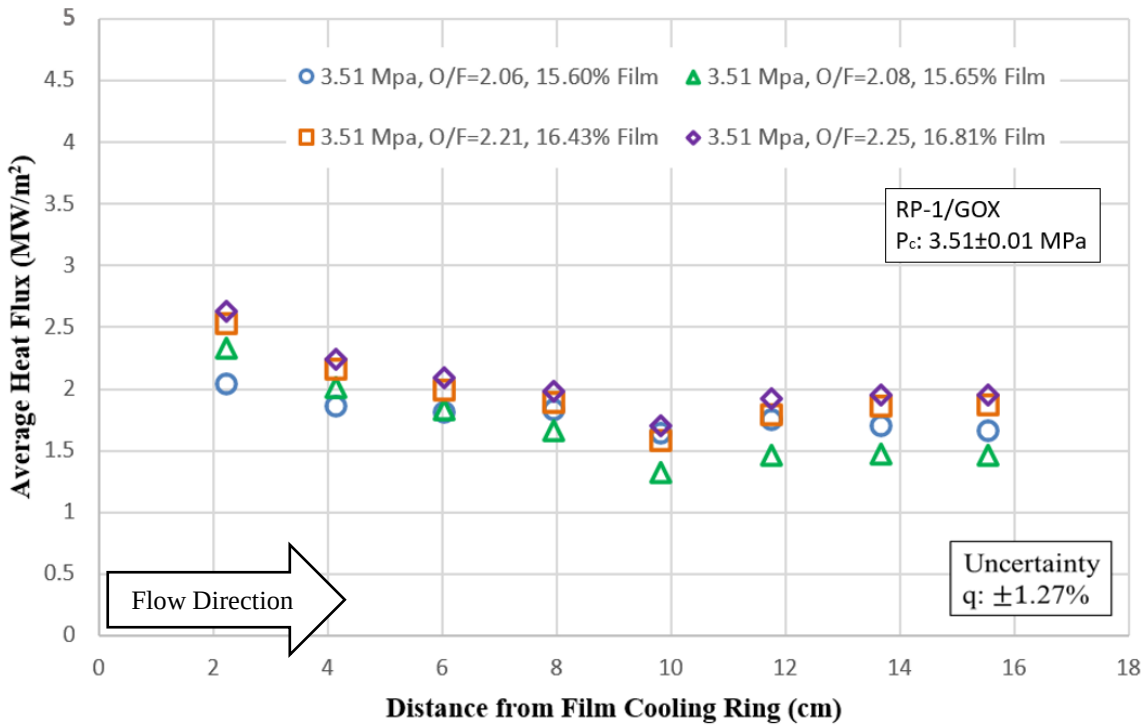


Figure 21. Average Heat Flux vs Axial Distance for RP-1/GOX at 3.51 MPa. Source: [1].

These test conditions were within the range of conditions studied for this thesis. Figure 21 indicated that the portion of the chamber 10 cm (3.94 in) from where film cooling was introduced experienced the lowest average heat flux. Reference [1] concluded that a minimum heat flux occurring at 10 cm (3.94 in) from the film cooling ring indicated a length at which film cooling is most beneficial. This data is also valuable because it shows that a chamber length approaching 8 cm (3.15 in) would experience 13.78% more heat flux at the beginning of its nozzle than one of 10 cm (3.94 in). Because of the factors listed in Chapter IV.B and [24], this was taken as the new chamber length for this study.

B. CHAMBER-NOZZLE DESIGN

1. Overview

The integrated chamber-nozzle component with internal cooling channels was designed to be additively manufactured out of SS 17–4PH on a Studio System™ printer suite from Desktop Metal. Reference [24] has more information on the Desktop Metal

printer suite and associated design considerations. The nozzle heat transfer rate distribution illustrated in Figure 3 necessitated a design in which cooling fluid swirls in a header near the nozzle bell, passes through channels flowing “forward” up the engine to the nozzle throat and chamber, and exits through outlets near the film cooling ring. For integration onto the existing film cooling ring, and for consistency of data collected, the initial design had the same internal chamber diameter as the test spools, 3.81 cm (1.5 in), and a 15° half-angle nozzle with a 1.588 cm (0.625 in) throat diameter, yielding a chamber-to-throat area ratio of 5.76. The chamber length was shortened to 7.62 cm (3.00 in) due to printer geometry limitations (see [24]).

Working from forward to aft, the main sections are: cooling fluid outlets and base (section 1 in Figure 22), the thrust chamber and nozzle with data collection ports and internal cooling channels (section 2), and the cooling fluid header and inlets (section 3). Overall dimensions are also shown. Figure 23 shows the chamber-nozzle mounted to the rest of the engine.

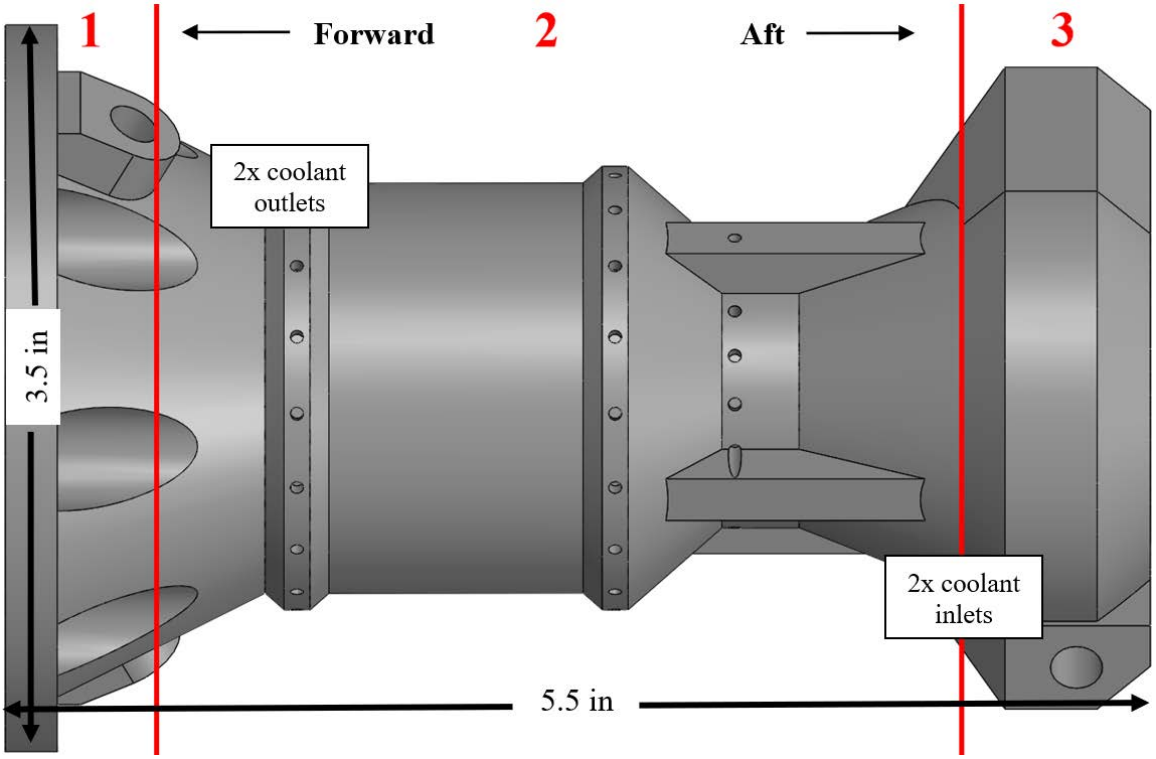


Figure 22. Integrated Chamber-Nozzle

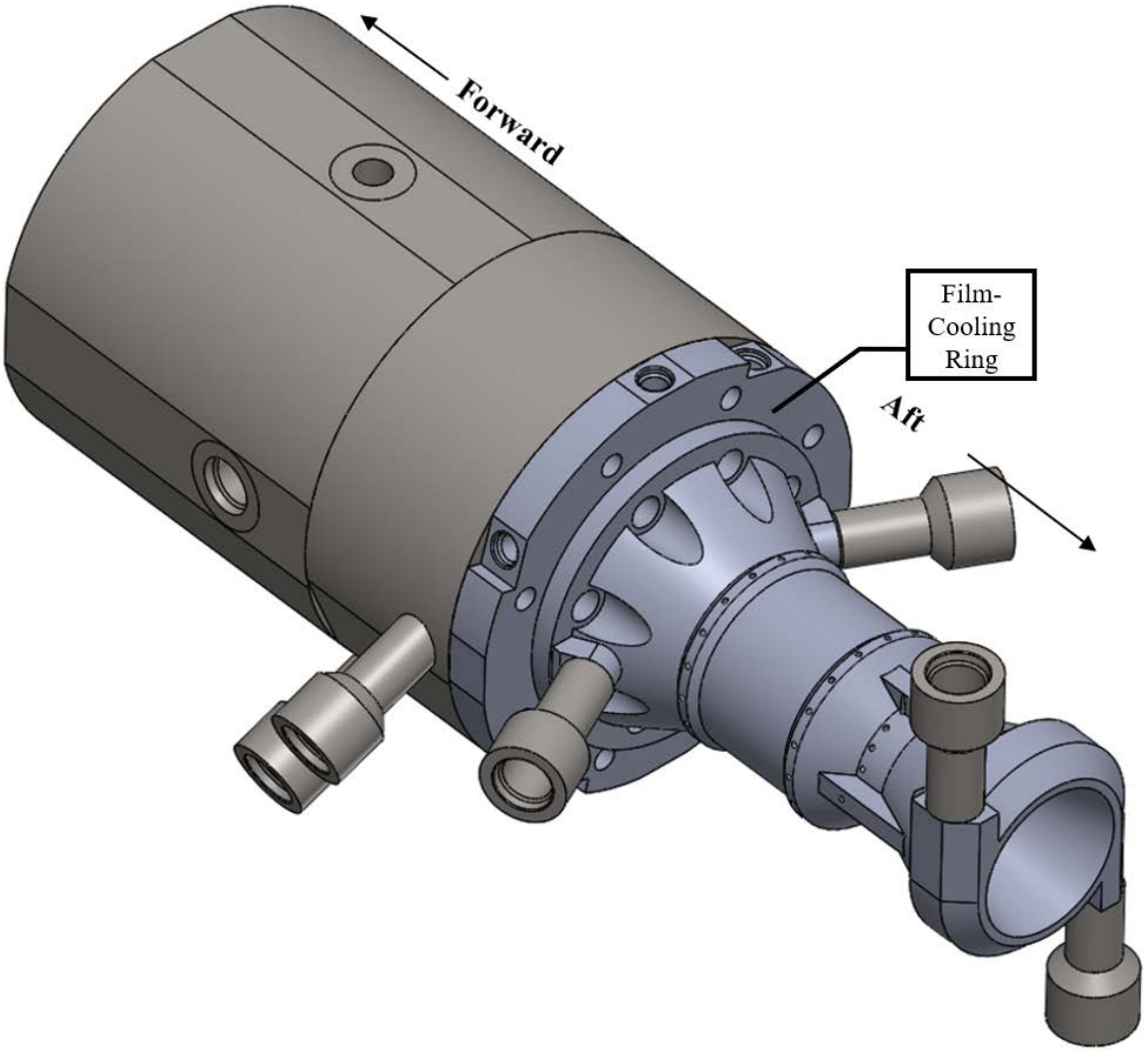


Figure 23. Mounted Chamber-Nozzle Assembly

The engine was designed to ensure minimal post-machining was required. In addition to the four cooling fluid fixtures that require welding to the inlets and outlets (seen attached in Figure 23), an O-ring groove also needs to be machined into the forward end to ensure hot gasses do not escape between the cooling ring and the base.

2. Base and Outlets

The cooling fluid outlet and base were designed to integrate directly with the film cooling ring. Eight counter-sunk bolt holes in the same pattern as the cooling ring allow

for direct attachment with 1.5-in 5/16-18 socket cap screws. One of these holes can be seen in Figure 24.

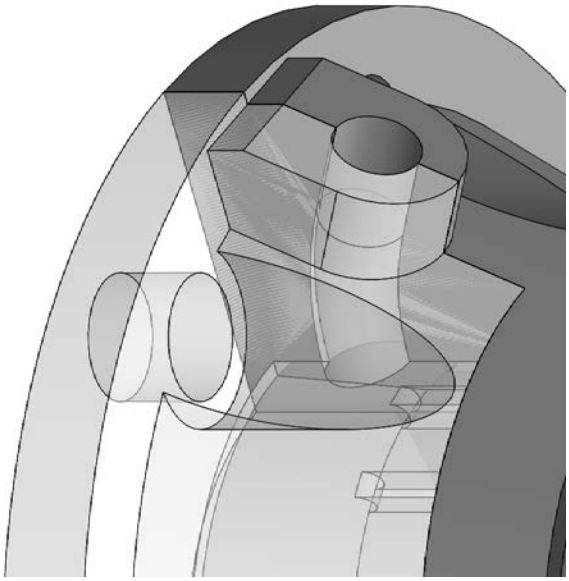


Figure 24. Section View of Chamber-Nozzle Base and Outlet

One of two 6.35 mm (0.25 in) -diameter curved outlet holes is also shown. These are offset in the aft direction from the base to allow for a post-machined O-ring groove and attachment of cooling fluid bosses. The outlets lead to an open internal section into which all 18 cooling channels empty. The forward end of the chamber can also be seen.

3. Channels

Aft of the base and outlets are the chamber and nozzle. The full 18-channel layout can be seen in the section view in Figure 25, and six of them can be seen in Figure 26.

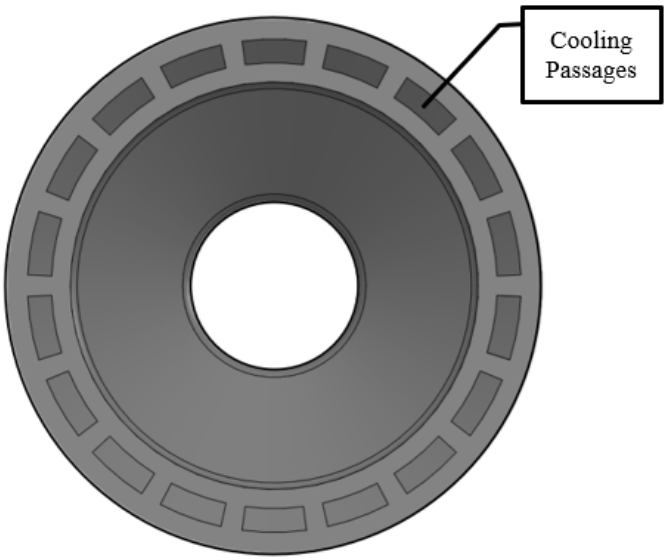


Figure 25. Chamber-Nozzle Cooling Channel Layout

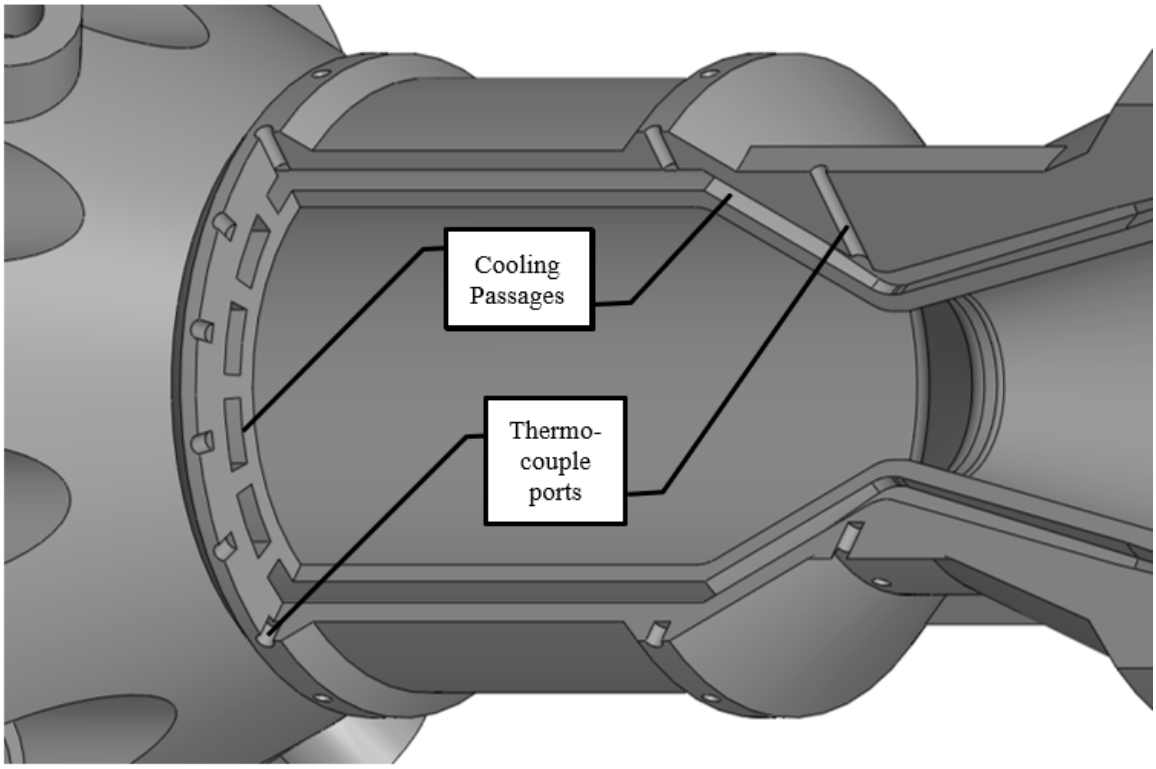


Figure 26. Section View of Chamber and Nozzle

Included in the design are 54 thermocouple ports, each with a diameter of 1.5875 mm (0.0625 in). They are evenly spaced circumferentially in three rings of 18 holes each.

Two of these holes in each ring continue through to the cooling channel for temperature measurements. 16 other holes, each centered on a cooling channel, act as guides for future drilling if more channel temperature measurements are required.

4. Inlets and Cooling Fluid Header

Surrounding the nozzle bell is the cooling fluid header. This acts as a large area in which the cooling fluid accumulates before entering the cooling channels. Figure 27 is a section view of the header.

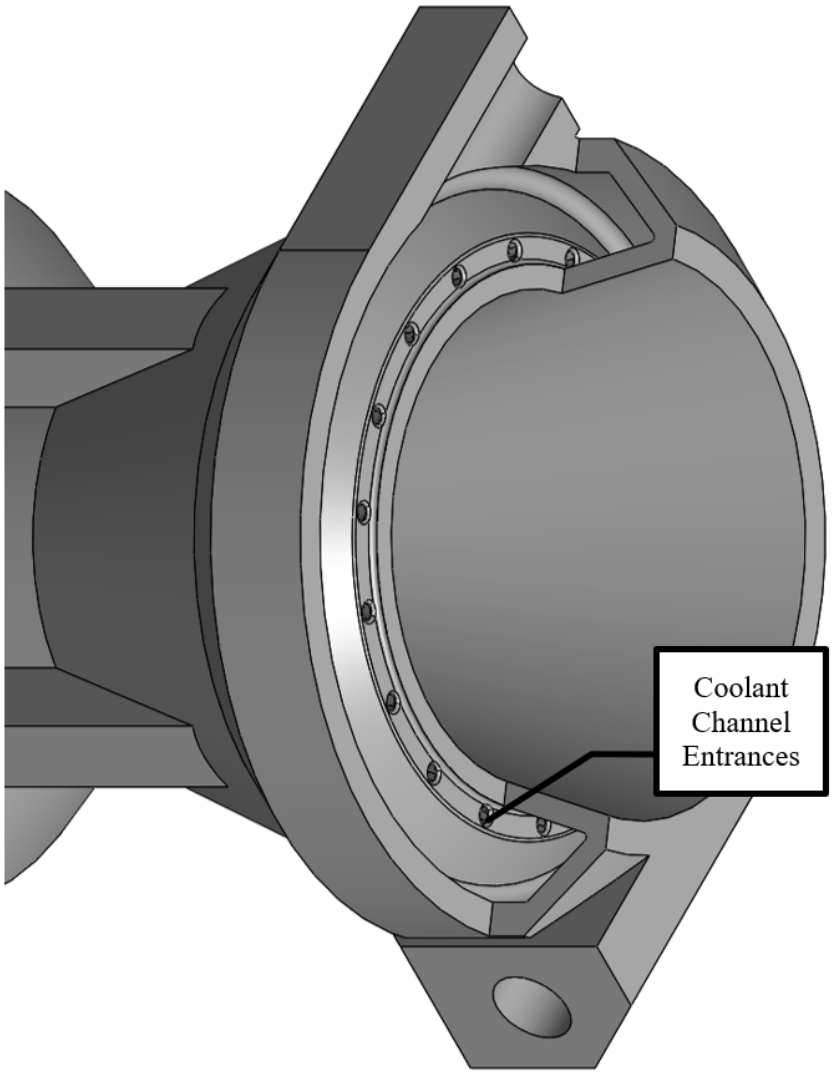


Figure 27. Section View of Chamber-Nozzle Header, Channel Entrances, and Inlets

Here, 11 of the 18 cooling channel entrances can be seen. The entrances transition from a circular to the near-rectangular profile seen in Figure 25 and were designed to ensure the fluid enters the channels with adequate velocity, determined iteratively as described in Chapter V.A.

Figure 27 also shows two 0.635 cm (0.250 in) diameter inlets. They are designed for two coolant attachments to which the existing test stand water hoses attach.

C. ADDITIVE MANUFACTURING FAILURE MODES

While speculating on the causes of these failure modes can be done, a rigorous failure analysis is outside the scope of this thesis. However, it is useful to note what occurred for future consideration. The two main failure modes observed were filament bulging and axial layer separation. Figure 28 is an example of filament bulging from an initial print attempt.

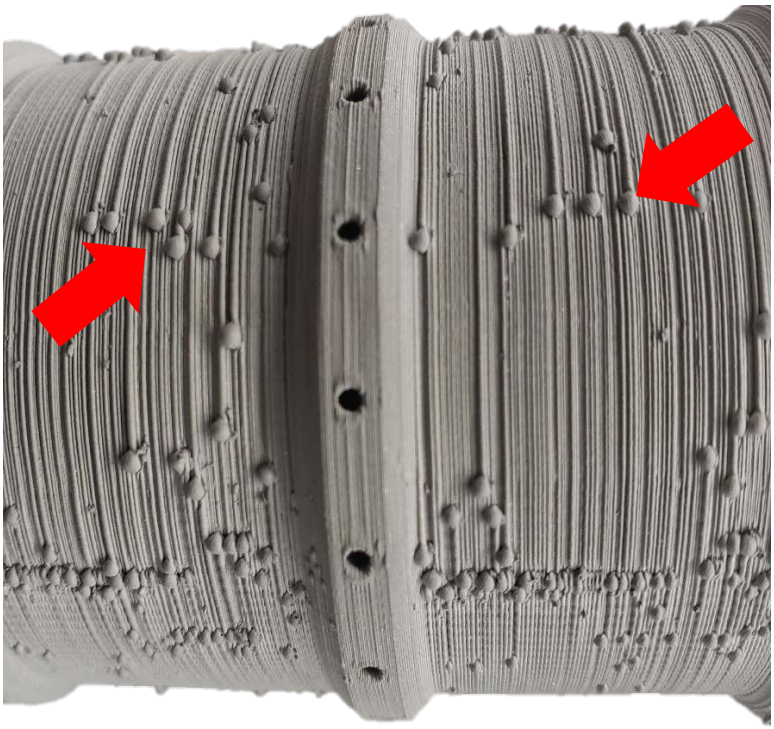


Figure 28. Additive Manufacturing Filament Bulging Failure Mode

While external bulges (the small droplets indicated by the red arrows in Figure 28) will not affect engine performance and can be machined off of the engine if necessary, any bulges that formed inside the cooling channels could severely impede cooling fluid flow and potentially lead to local overheating and engine failure. While internal bulging has the potential to be problematic, the layer separation depicted in Figure 29 is what ultimately made this print unusable.

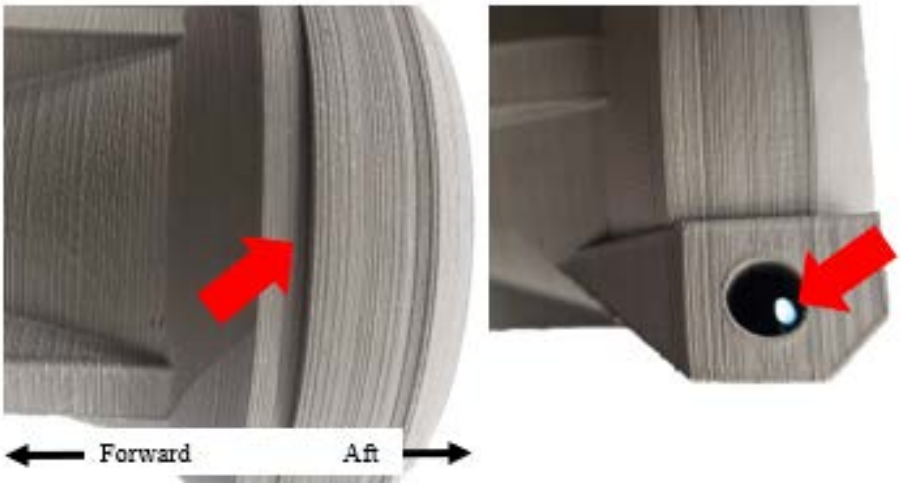


Figure 29. Additive Manufacturing Axial Layer Separation Failure Mode

On the left of Figure 29 is what the separation (the protrusion indicated by the left red arrow) looks like near the outside of the nozzle bell. It is not immediately clear from external observation that separation has occurred because, while the layers are clearly offset radially, there is no visible gap between them axially. The layers did not maintain a consistent external radius as expected. Looking into one of the inlets, light seen through the wall (red arrow on the right of Figure 29) comes from the separation failure shown on the left and indicates appreciable axial separation. As seen in Figure 27, this is a deviation from what is expected because, since the two inlets are offset 180° from each other, the view into the inlet should be an internal wall with no holes. Ignoring the structural integrity implications, this also renders the header section non-watertight. Cooling fluid leakage during operation will lead to a decreased mass flow through the cooling channels, a lower heat transfer rate per Equation 11, and chamber heating beyond anticipated values.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS AND DISCUSSION

A. MODELING RESULTS

Several CFD simulations were run on various test spool configurations, as well as on the integrated nozzle, in an attempt to better understand internal coolant fluid flow. The results helped understand and modify appropriate internal geometry details for the chamber-nozzle and also indicated a flow bias behavior in the test spool configuration that had not been previously observed.

CFD software is useful when the goal is to visualize and quantify either external or, in this case, internal fluid flow. The software does this by breaking up the fluid and solid volumes into very small sections called "cells." The summation of all cells in the simulation is known as the "mesh." Once these are defined by the user, CFD software will systematically and iteratively solve the fluid governing equations with various numerical methods. The intricacies of CFD techniques is beyond the scope of this thesis, but [25] and [26] offer very detailed discussions on the governing equations and how they are solved by CFD software.

This chapter shows two example simulations. One was run to visualize internal flow of the cooling fluid through the chamber-nozzle channels and quantify the difference in mass flux through each channel with the goal of determining if there is a channel bias. Chamber heat flux values were then fixed to represent steady state combustion, and the temperature difference from inlet to outlet was measured as well. The second simulation offers a visualization of the flow of the cooling fluid through the existing test spool configuration.

1. Computational Fluid Dynamics Software

The CFD software utilized for this project was SolidWorks Flow Simulation. This software has proven to be reliable for small- to medium-scale fluid flow calculations [27], [28]. SolidWorks Flow Simulation was chosen for two reasons: it was seamless to integrate with the existing part models, which were already created using SolidWorks; and the solver

was less computer-resource intensive than other available CFD software. Figure 30 through Figure 33 depict the setup steps for testing the integrated nozzle.

2. Setup and Boundary Conditions

Specific steps were needed to prepare for the CFD analysis once the model was loaded in Solidworks. Figure 30 shows the model in the Solidworks window during setup.

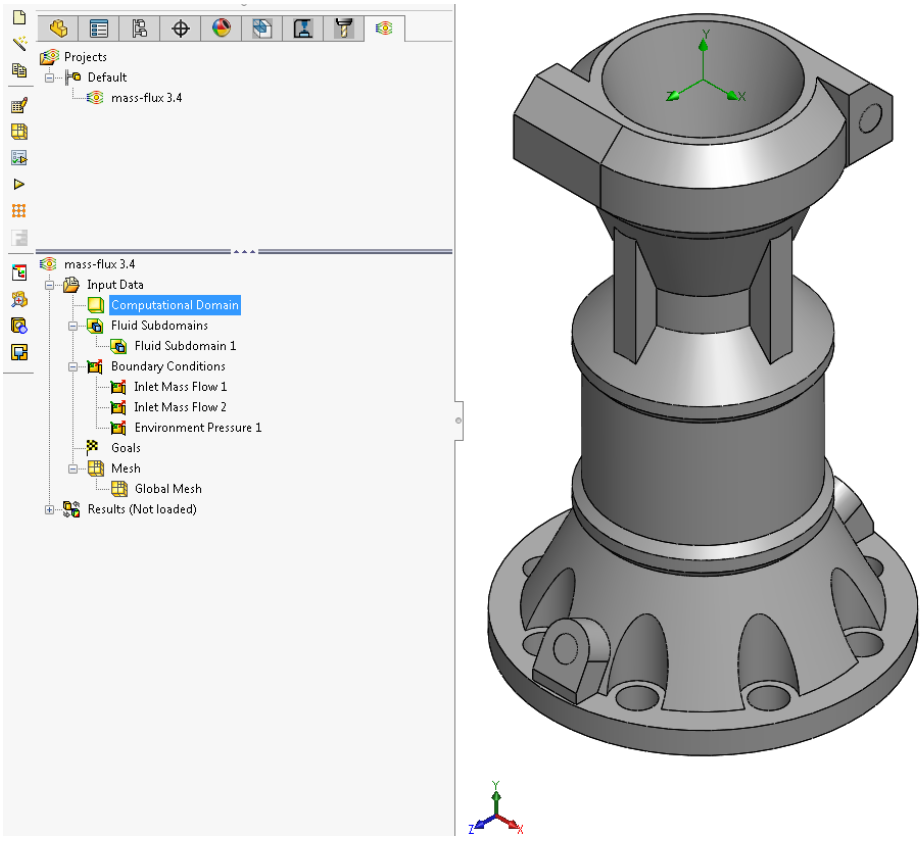


Figure 30. SolidWorks Computational Fluid Dynamics Software Initial Setup

Next, the fluid subdomain was defined such that it would encompass the entire interior volume of the headers and channels. The fluid subdomain is shown in blue in Figure 31.

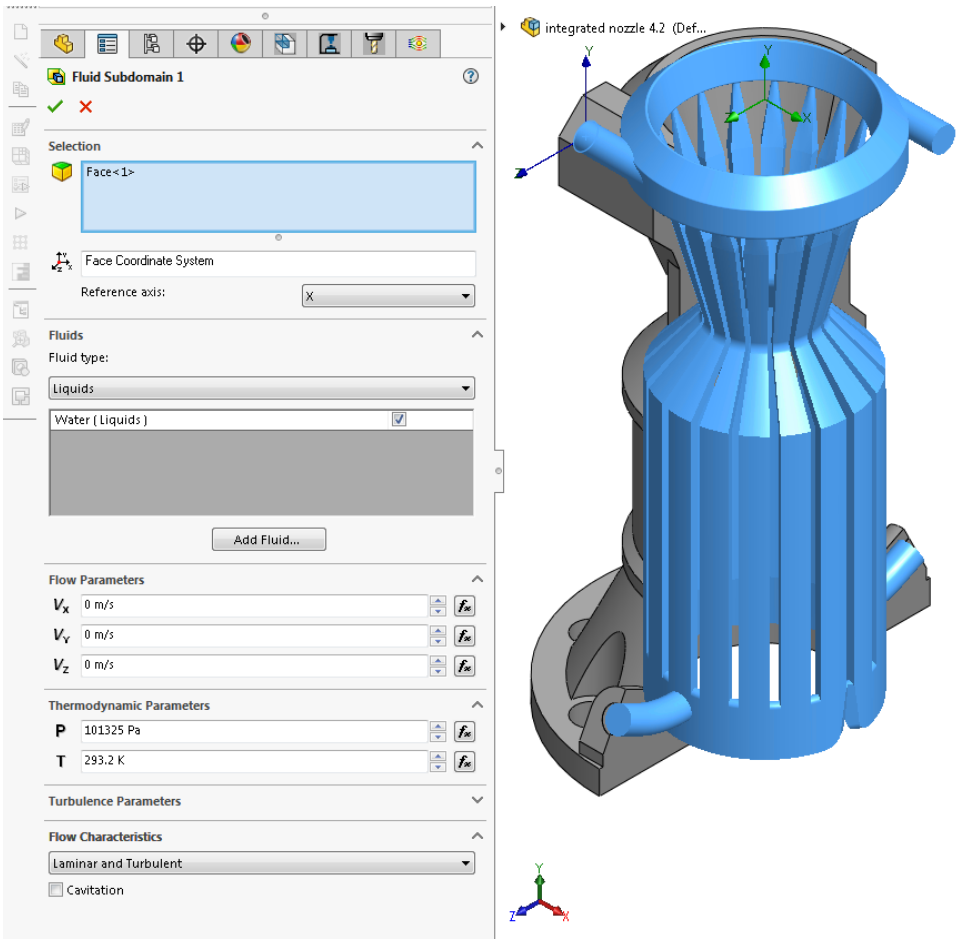


Figure 31. Defining the Fluid Domain

The flow characteristic of “Laminar and Turbulent” was selected because there was a reasonable assumption that the complicated geometry would yield a transition between these types of flow. This allowed for more accurate results.

The next step was to define appropriate boundary conditions. Since this simulation is meant to represent what the behavior of the cooling fluid would be when attached to the test stand, the Inlet Mass Flow boundary condition parameters were set to represent 1.0 kg/s (2.205 lb/s) of water flow total. Thus, each inlet needed to be set to 0.5 kg/s (1.102 lb/s), as shown in Figure 32.

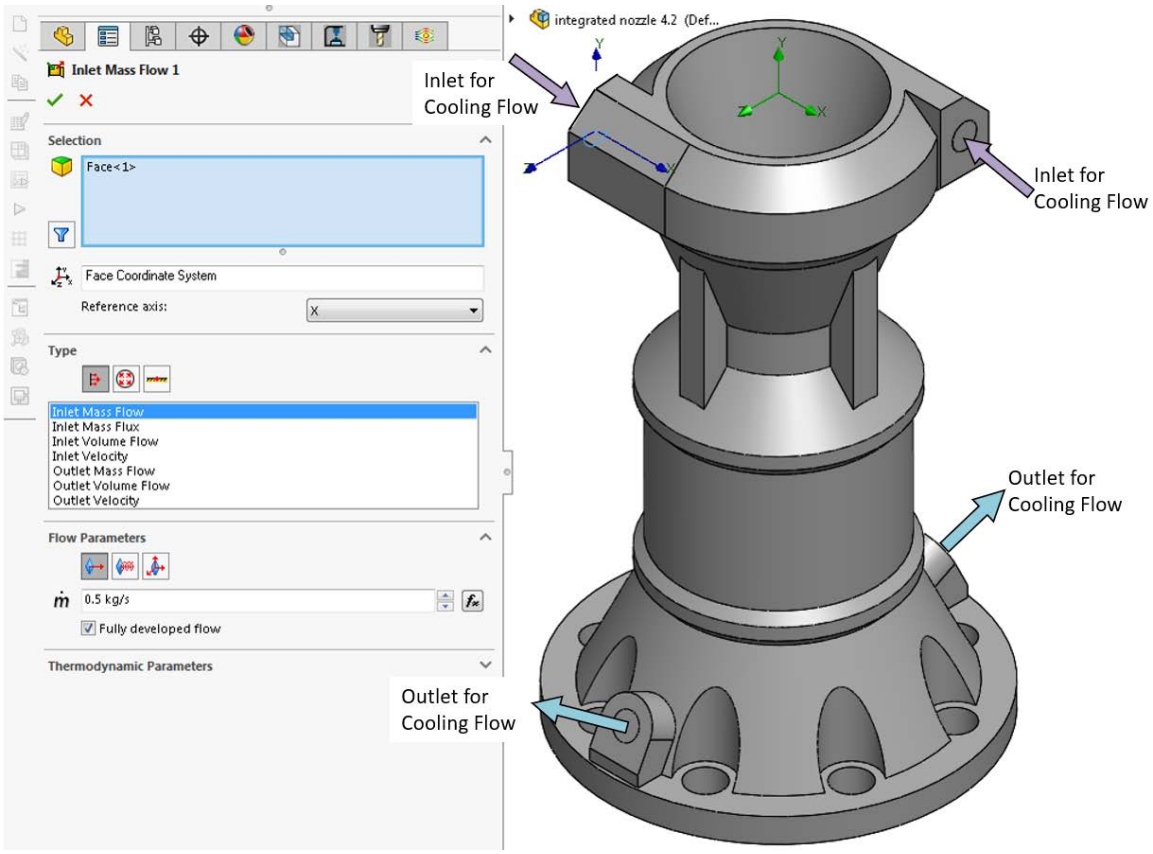


Figure 32. Inlet Boundary Condition Setup

Here, one of the two required inlet boundary conditions is shown. The outlet boundary conditions were defined to discharge to ambient pressure, which simulates the open-loop system the chamber-nozzle will experience during its first test.

The final boundary condition that was set was a constant representative surface heat generation rate of 3 MW/m^2 (264 $BTU/s\text{-}ft^2$) on the inner chamber wall and nozzle. This heat flux approximated an experimental run condition at 1.8 O/F, 3.5 MPa (500 psia), and assumed steady state combustion conditions.

Finally, the mesh had to be defined such that the cells were of an appropriate size to capture the behavior of the cooling fluid within the channels. The basic mesh grid is shown in Figure 33.

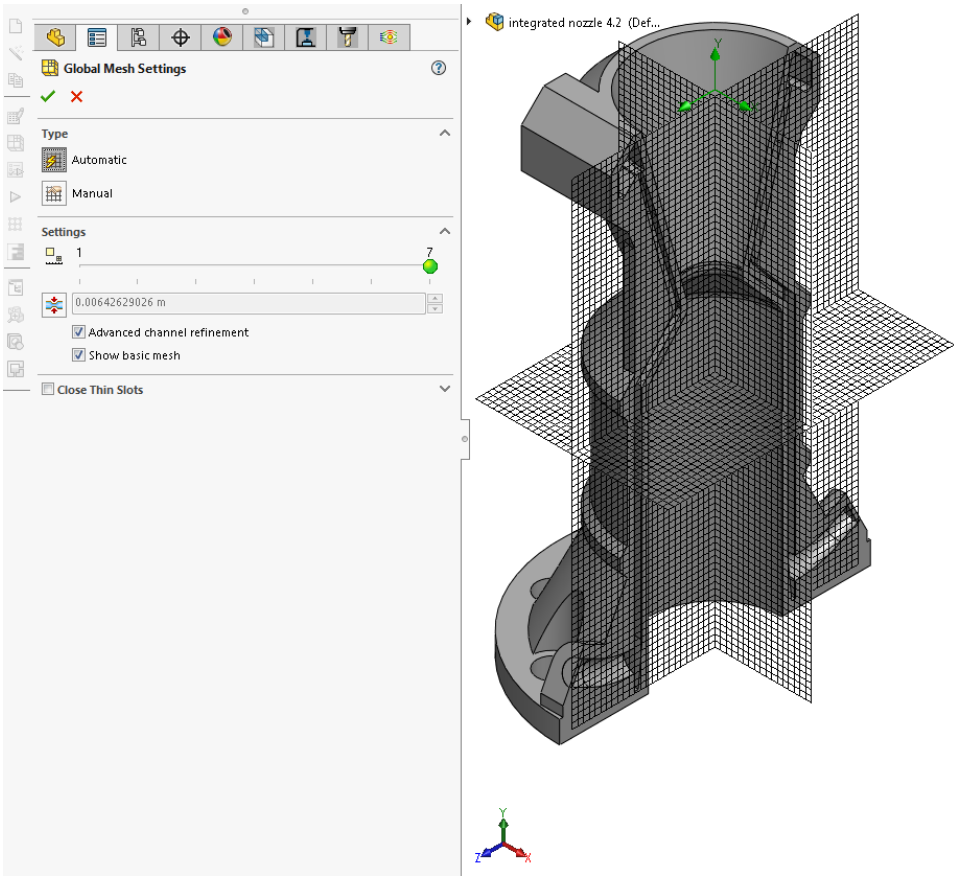


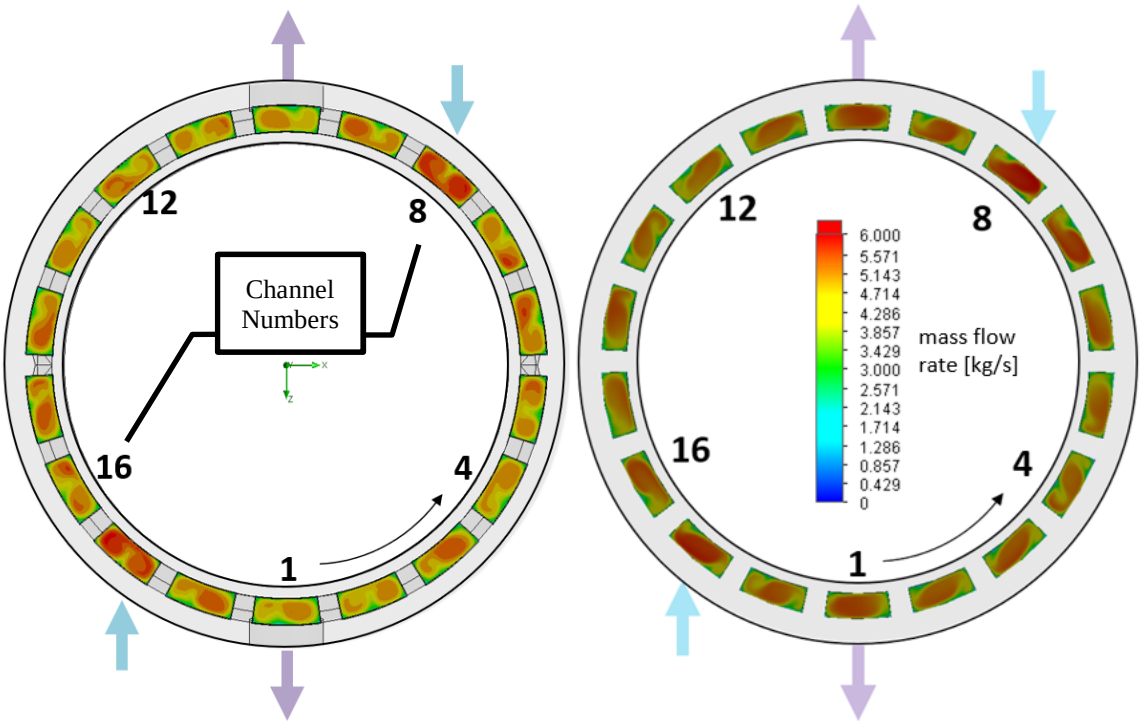
Figure 33. Defining the Global Mesh

SolidWorks gives the option to further define cell geometry based on channel dimensions, if desired. Typically, it is recommended to define a mess geometry that will balance simulation accuracy with computation time. It was determined that the highest mesh setting of "7" resulted in cell numbers on the order of 7 million and offered adequate refinement and accurate results that were commensurate with the empirically measured data presented in Chapter V.B ad Appendix C.

3. CFD Results

Two types of results were used to inform the nozzle's design: a mass flux analysis and a flow trajectory study. Results from both were incorporated into subsequent iterations of the design, and more analysis is encouraged during future engine integration.

A mass flux analysis was conducted on the channels to quantify fluid mass flow rates within each of the 18 channels. These changes resulted in a less than 6% variation of flow through the channels as seen in Figure 34 and Figure 35.



Cross section of mid-chamber channels from design iterations 2.0 (left) and 4.2 (right). Channels numbered 1 through 18 counter-clockwise. Legend applies to both flow rate analyses. Blue arrows denote inlet positions. Purple arrows denote outlets.

Figure 34. CFD Mass Flow Rate Analysis Results

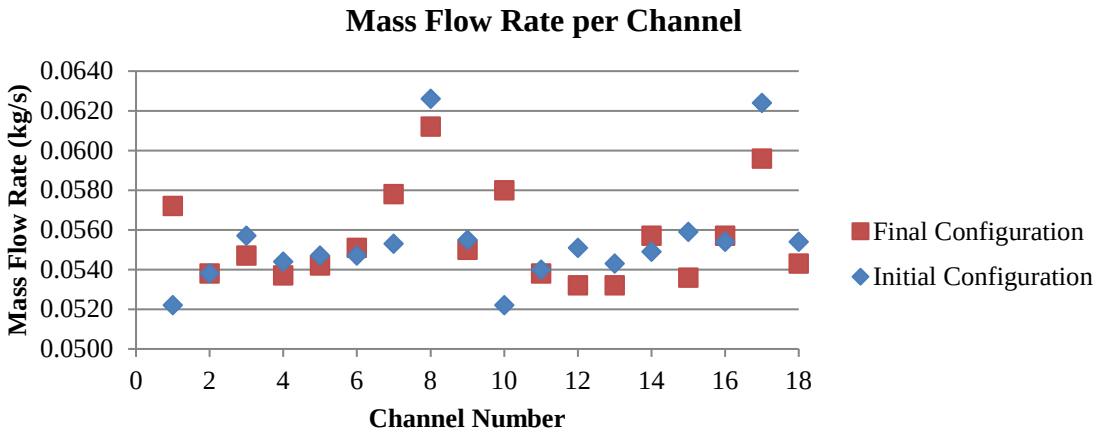


Figure 35. Mass Flow Rate Per Channel

The CFD results were integrated over the exit area for each cooling channel and the initial results revealed a flow bias towards the channels that were immediately adjacent to the inlets (the darker red portions in channels 8 and 17 on the left of Figure 34). Because of this, subsequent iterations included two changes: large flanges in the downstream header to more uniformly distribute and direct the flow towards the outlets, and inlet ports that were more tangential to the inlet header.

Flow trajectories were useful in visualizing recirculation zones within the inlet header and channels and were used to inform channel inlet geometry. An example of a velocity flow trajectory visualization from an intermediate channel-nozzle version is shown in Figure 36. Slight channel bias for the rightmost channel is indicated by the red lines.

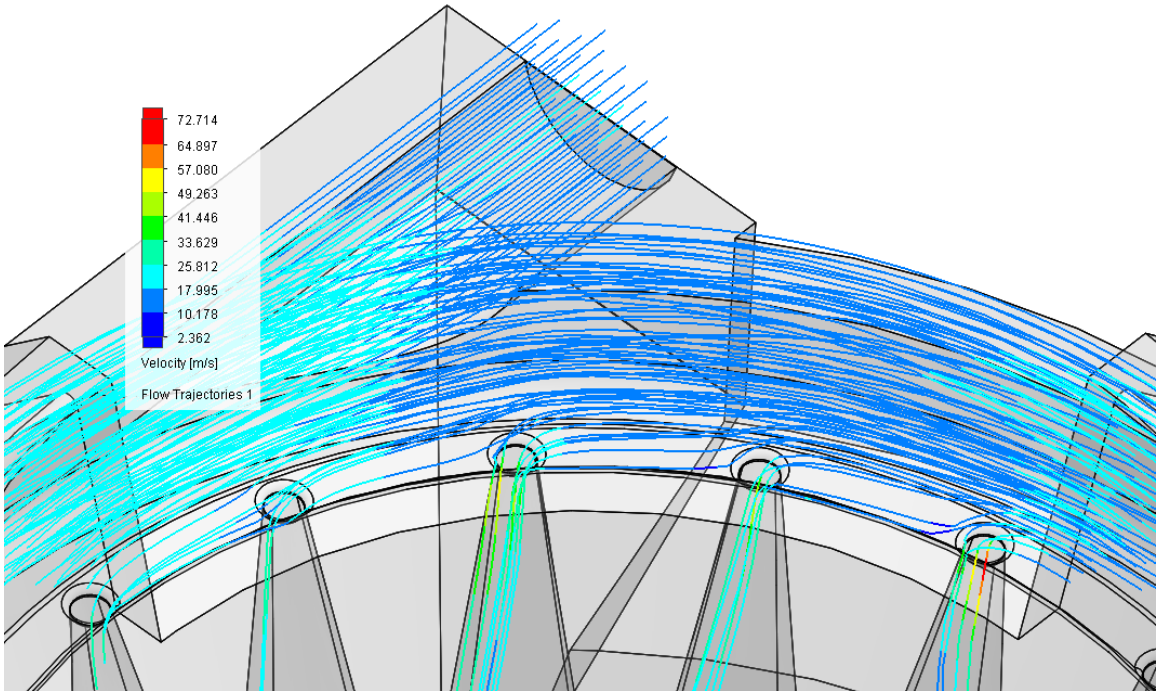


Figure 36. CFD Inlet Header Velocity Visualization

The coolant fluid temperature rise was predicted when the heat flux was applied to the chamber. Figure 37 shows a temperature difference of less than 20 K (36 °F) over a combined chamber and nozzle length of 14 cm (5.5 in).

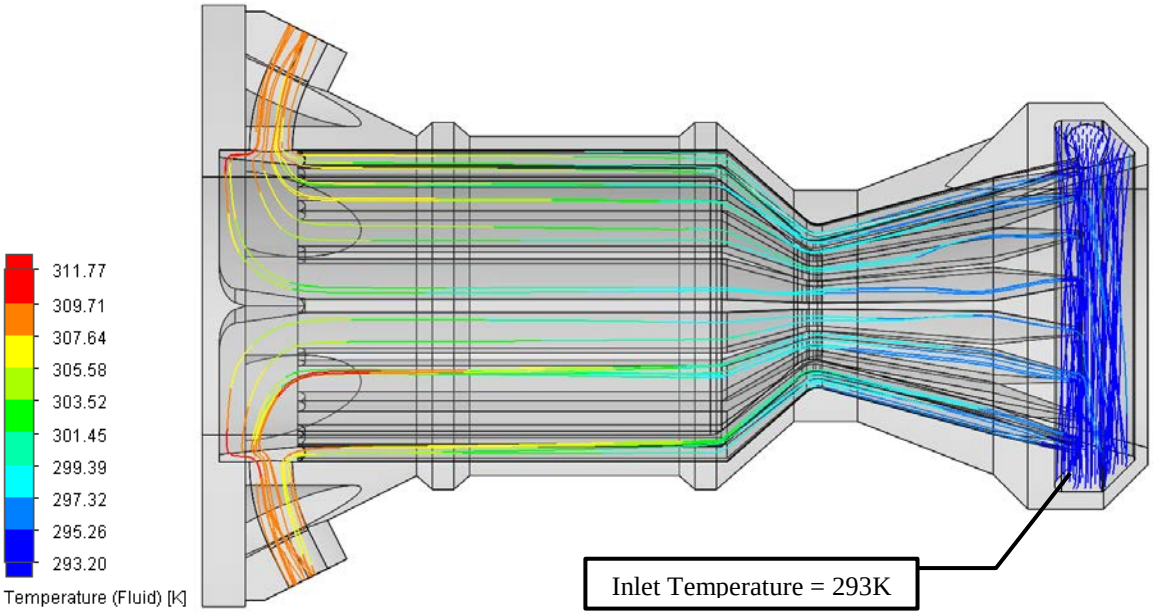


Figure 37. CFD Temperature Rise Results

The temperature rise seen in Figure 37 is commensurate with temperature increases during the hot fire results discussed in Chapter V.B.

The existing single test spool and water cooling configuration was also simulated by the same methods, but without the heat flux parameter. The results offered a visualization of the fluid flow within the cooling jacket, and revealed decreased flow in the dark blue area shown in Figure 38 that had not been witnessed before.

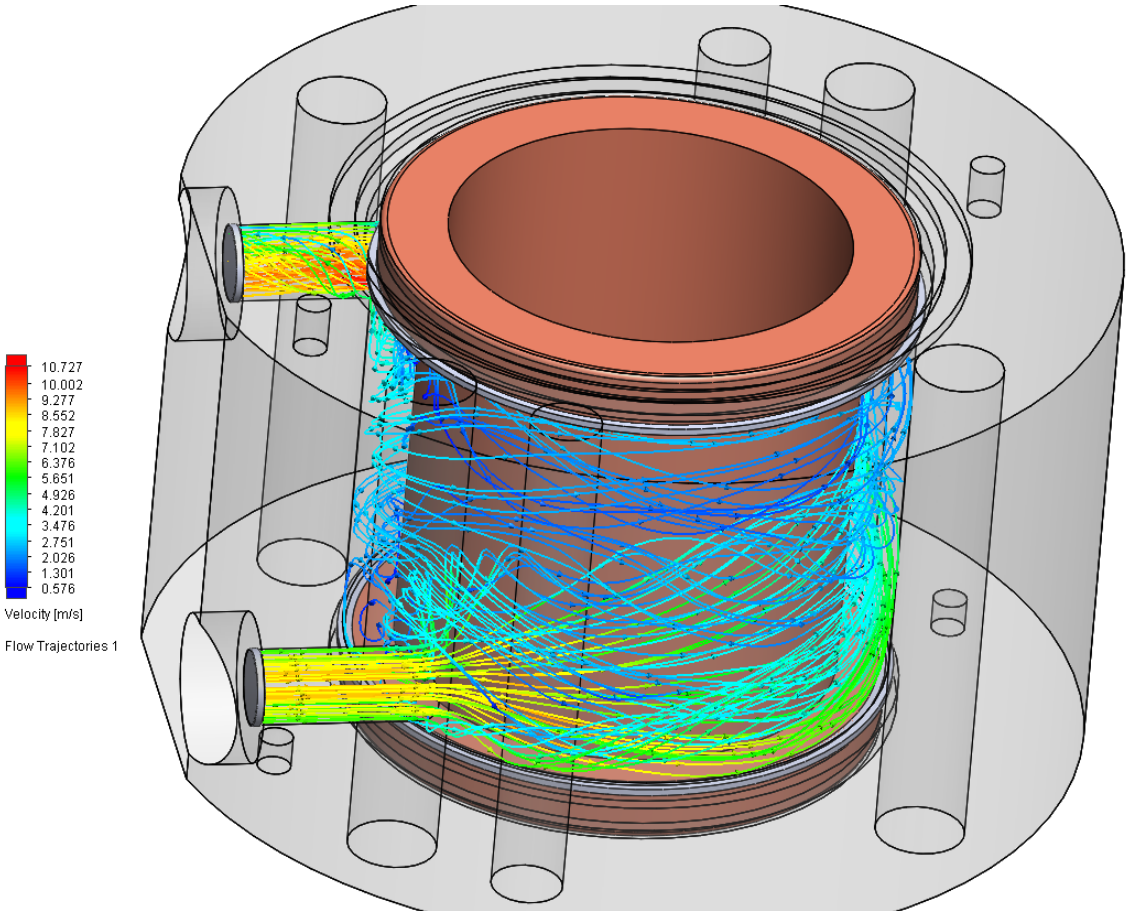


Figure 38. Spool Flow Simulation

This decreased flow led to decreased heat transfer into the cooling water in that area and resulted in higher local material temperatures and discoloration of the outside surface of the spool. This discoloration can be seen in Figure 39.

Figure 39. Spool Discoloration After Apr09Run03

This phenomenon did not negatively affect the outcome of this heat transfer study, nor did it contribute to any component deformation or failure. However, it should be noted that circumferentially biased coolant flow in an open cooling system (as opposed to utilizing discrete cooling channels) could lead to uneven chamber cooling and ultimately an engine failure. A chamber cooling scheme such as axial cooling channels should be used to ensure a more even coolant distribution.

B. TEST FIRE CONDITIONS AND RESULTS

A total of 33 test runs were conducted over the course of the test campaign spanning from October 2019 to April 2020, 15 of which yielded valuable data. The other runs were either too short to reach steady state, resulted in hardware failure, or exhibited some other data anomaly. Table 4 shows logs of the tests with representative data.

Table 4. Liquid Rocket Engine Test Log

Run	Run Time (sec)	Chamber Pressure (MPa)	O/F Core	% Film Cooling
Oct25Run03	10	4.0342	1.78	19.01
Oct25Run04	10	3.8852	1.84	19.38
Oct25Run05	10	4.0622	1.91	19.34
Nov01Run02	12	4.1529	2.07	19.69
Nov01Run03	12	4.2471	1.91	19.16
Nov01Run04	12	4.2506	1.76	18.79
Nov12Run01	12	4.1721	1.96	13.36
Nov12Run02	12	4.2357	1.78	13.28
Nov20Run03	30	6.8709	1.92	13.33
Nov21Run01*	21	6.9627	2.28	9.650
Dec04Run01	15	4.1137	2.23	13.46
Dec10Run02	15	4.1530	2.16	13.41
Dec10Run03	15	4.1602	2.27	13.46
Apr08Run03	12	4.2802	2.00	13.33
Apr09Run03	12	4.2908	2.00	13.33

* Nov21Run01 resulted in hardware failure.

Table 4 shows the mixture ratio (O/F) quoted in core flow. This refers only to the propellant that was injected and combusted, and neglects the mass flow rate of the film cooling fluid. The film cooling percentage was calculated using Equation 13.

$$\%Film\ Cooling = 100 * \frac{\dot{m}_{fuel_{film}}}{\dot{m}_{fuel_{film}} + \dot{m}_{fuel_{core}}} \tag{13}$$

Additional data was available for the test fire results from [1], and all test runs from 25 October to 10 December 2019 for this project were added to that database. The tests on 8 and 9 April 2020 were conducted to verify that the additively manufactured SS 17–4PH spools would exhibit adequate cooling characteristics near the other materials studied, and maintain their structural integrity after an appreciable run duration. Two SS 17–4PH test spools with 0.254 cm (0.100 in) and 0.178 cm (0.070 in) chamber wall thicknesses were designed, printed, and tested. A detailed log of the spools tested is shown in Appendix C.

First, a preliminary burst pressure test was conducted to ensure the printed spools would maintain their structural integrity under test fire chamber pressures on the order of 4 MPa (580 psia). Figure 40 shows the burst test setup.

The pressure was read remotely off of a standard pressure gauge. An aluminum cylinder was placed around the device to help contain debris in the case of test failure.

Figure 40. SS 17–PH Spool Preliminary Burst Test Setup

The spools were placed in a water-cooling jacket from the test stand apparatus and aluminum caps were affixed to each end. The chamber area was filled with water and a high-pressure N_2 tank was used to pressurize the chamber spool to 8 MPa (1160 psia) for 10 seconds. This was approximately twice the expected chamber pressure. Upon completion, the spools were visually inspected and there was no bowing, layer splitting, or other deformation. The 0.254 cm (0.100 in) spool can be seen in Figure 41 after its burst test.

Figure 41. 0.254 cm Spool After Pressure Test

The results from these tests indicated that the SS 17–4 spools would not deform under test fire conditions.

1. Mass Flow Rate, Temperature, and Pressure Data

The time histories in Figure 42–Figure 47 are for a 4.28 MPa (621 psi) chamber pressure, 2.0 mixture ratio, and 13.33% film cooling over a 12-second test fire of the 0.254 cm (0.100 in) wall thickness SS 17–4PH chamber spool. Mass flow for this run can be seen in Figure 42. If a test condition reached steady state, this time history data was used to determine the proper time interval for averaging.

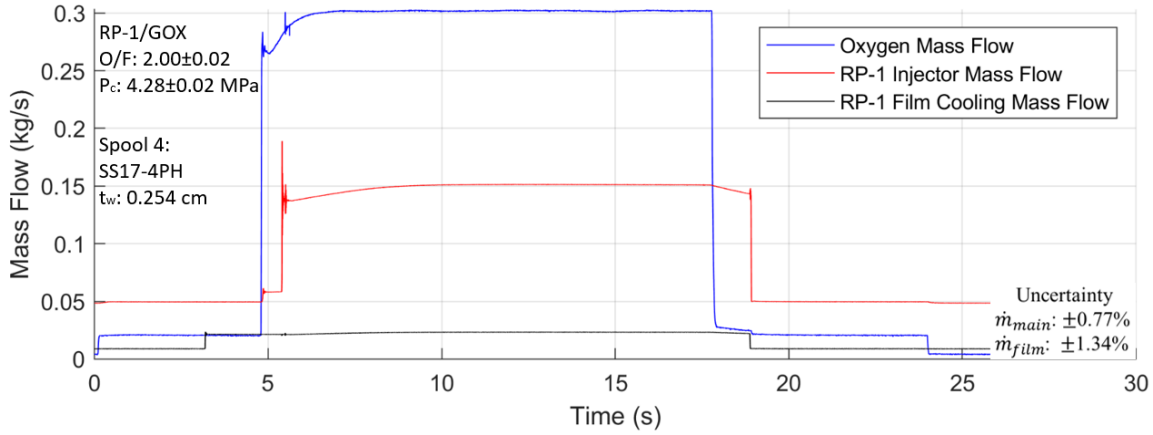
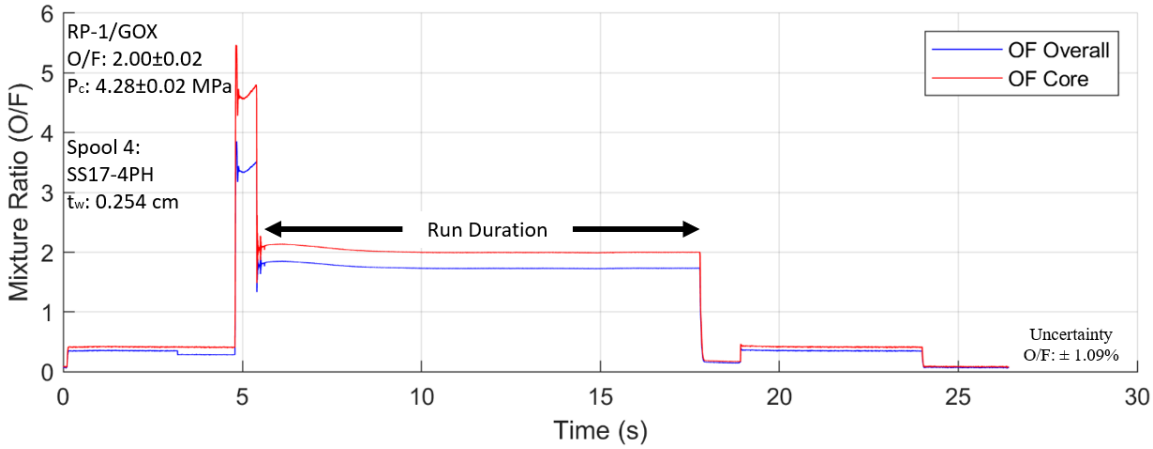


Figure 42. Film Cooled Rocket Mass Flow for Apr08Run03

Here, the small spikes in mass flow at five seconds indicate startup surges. Using Equation 2, the O/F ratio was determined and plotted in Figure 43.



Higher regions from 5–6 seconds and 19–24 seconds represent artificial O/F from start-up and shut-down sequence.

Figure 43. Film Cooled Rocket Mixture Ratio for Apr08Run03

Figure 43 shows constant mixture ratios for the duration of the run. The O/F Core values are the difference between the O/F Overall and the O/F from film cooling. The core flow is what drove the operating condition.

Chamber pressure similarly was held steady for the duration of the run, as can be seen in Figure 44.

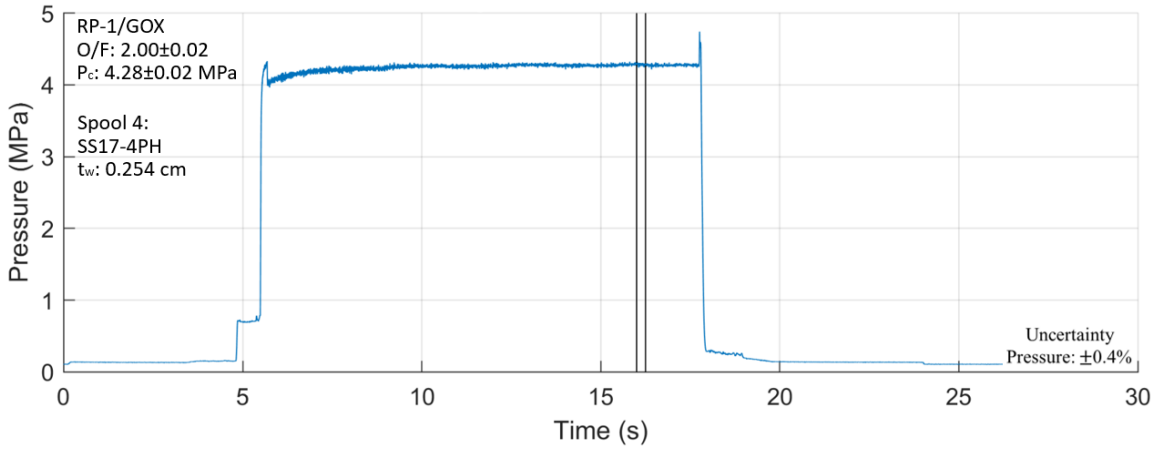


Figure 44. Engine Chamber Pressure for Apr08Run03

Again, the small spikes at 6 seconds 17 seconds were due to starting and shutdown sequence.

Temperatures were taken of each individual water outlet for each spool and the nozzle, as seen in Figure 45.

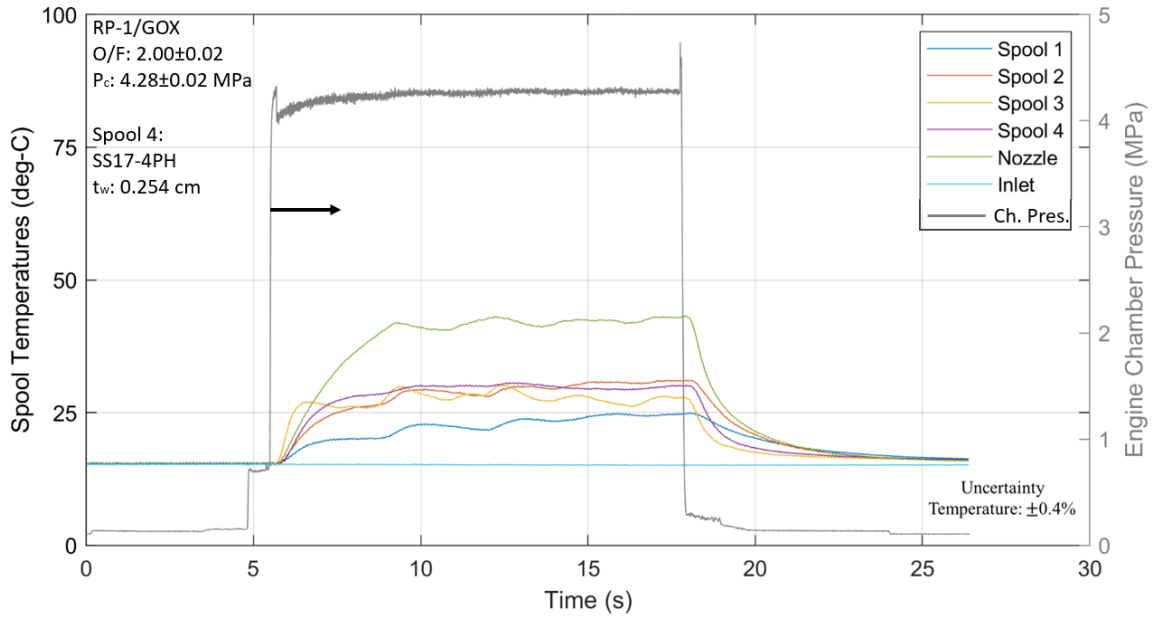


Figure 45. Test Spool and Nozzle Temperatures with Chamber Pressure for Apr08Run03

It is clear that the nozzle has the highest heat flux within the engine. Some spools showed a slight oscillation phenomenon in the temperature and heat flux values that was seen in most test runs. The exact cause of such oscillations is unknown, but it could be a function of engine dynamics, cooling jacket flow field changes, or measurement device peculiarities. In order to account for these oscillations, data was time-averaged over a short interval (grey vertical lines in Figure 44).

Film cooling ring temperature data is shown in Figure 46.

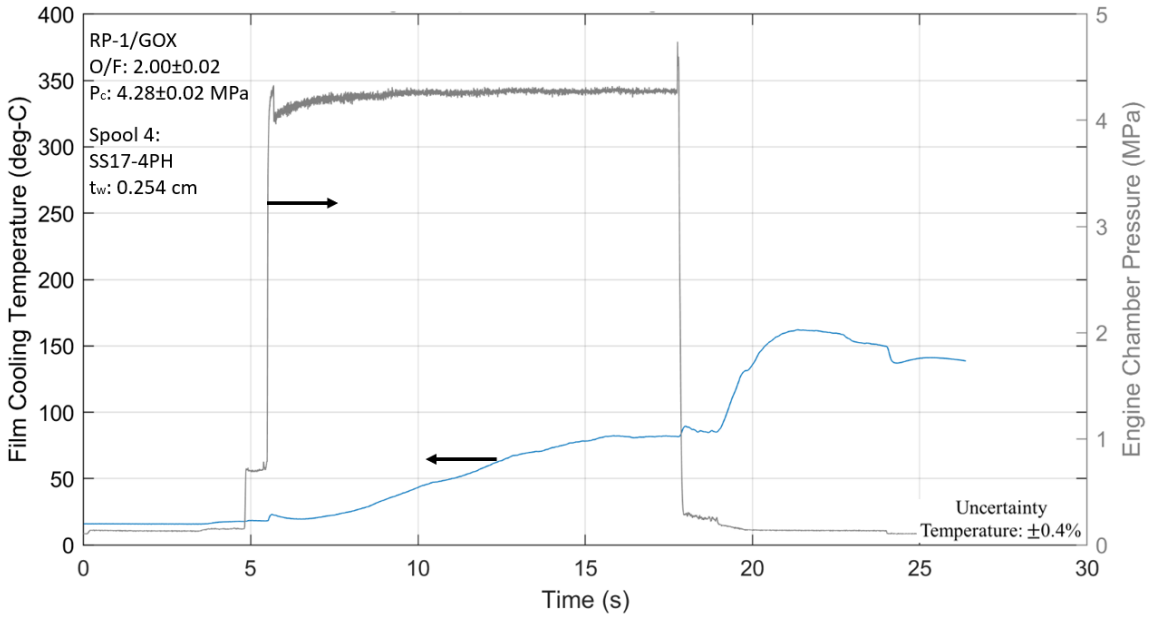


Figure 46. Film Cooling Temperature and Chamber Pressure for Apr08Run03

These temperatures are for a film cooling percentage of 13.33%. The temperature of the film cooling fluid in the reservoir experienced a steady increase over time. At the end of test fire, the temperature spiked, as shown at about 21 seconds, because the film cooling flowrate went to zero. This residual fuel instantly absorbed a significant amount of heat from the film cooling ring in which it was pooled. This behavior was expected and observed in every run. Figure 48 shows what happened to this film cooling ring temperature data during hardware failure.

The heat flux can be seen in Figure 47, and exhibited a similar oscillatory behavior that was also dealt with by time-averaging.

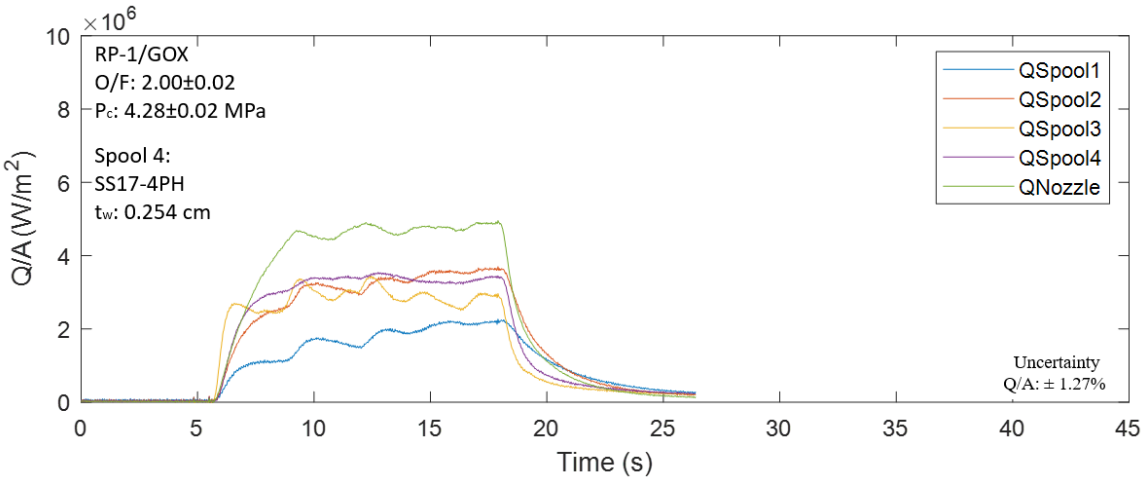


Figure 47. Test Spool and Nozzle Heat Flux for Apr08Run03

2. Failure Modes

There were two failure modes that the engine experienced during this testing program: the film cooling ring and nozzle overheated at a very low film cooling percentage, and the ignition system's spark plug over-pressurized and detached.

The first failure mode occurred during Nov21Run01 at an O/F of 2.0, 9.59% film cooling, the lowest of the test campaign, and a chamber pressure of 6.96 MPa (1009 psi), the highest of the test campaign. While engine chamber pressure appeared nominal throughout the run, the failure was a result of high film cooling ring temperatures due to inefficient film cooling. The engine component temperature and chamber pressure for this run are depicted in Figure 48 and Figure 49.

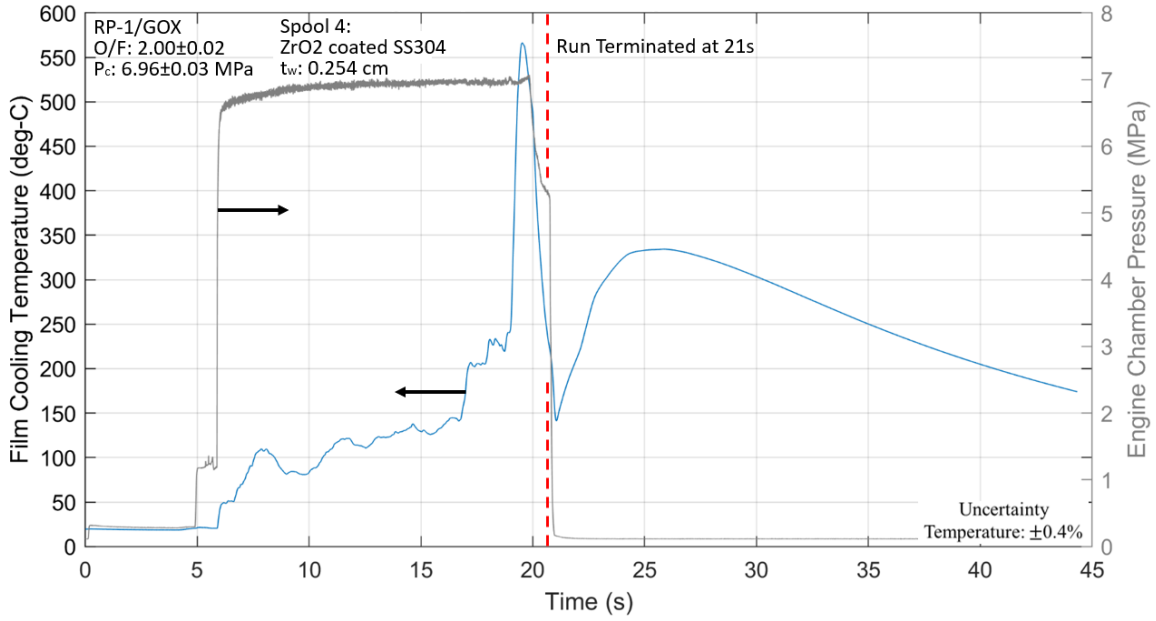


Figure 48. Film Cooling Temperature and Chamber Pressure During Equipment Failure

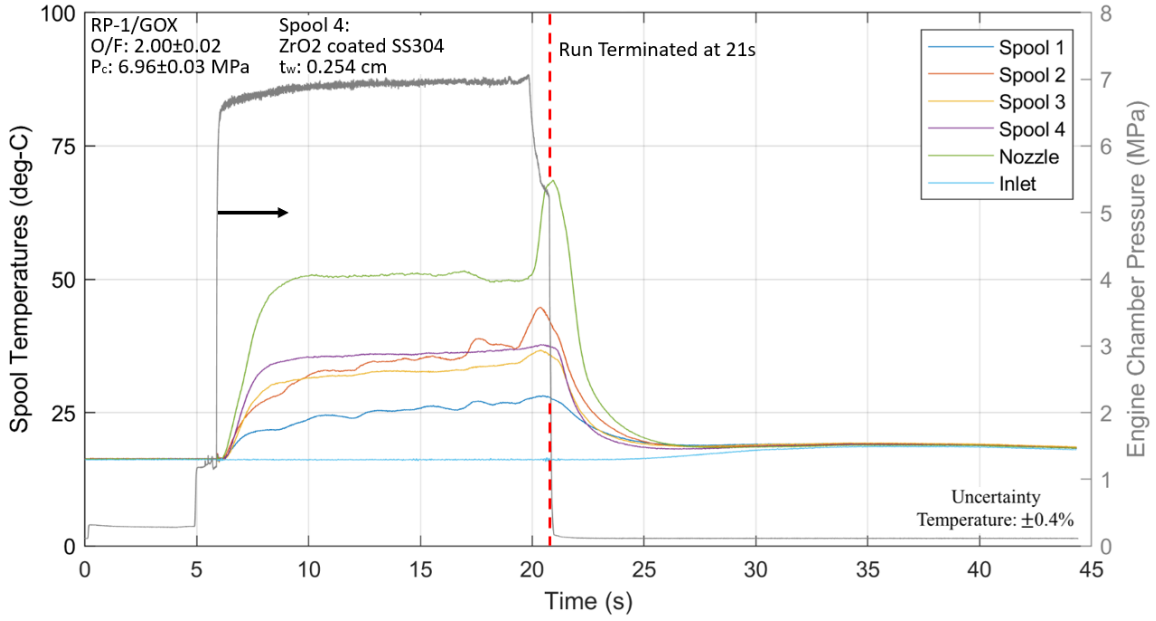


Figure 49. Test Spool and Nozzle Temperatures with Chamber Pressure During Equipment Failure

The temperature history for a typical run is shown in Figure 45 and Figure 46. While the temperatures were clearly rising immediately before termination, the temperature spike

at 19 seconds in Figure 48 may have been caused by damage to the film cooling thermocouple itself. All temperatures are monitored during test runs and were below the abort temperature of 250 °C (440 °F) up until 19 seconds. There was also a sudden pressure drop of 1.75 MPa (254 psi) that occurred at 20 seconds. Equation 12 shows why this occurred; nozzle throat area is inversely proportional to chamber pressure, so this pressure change was likely caused by a sudden increase in nozzle throat area due to throat erosion, as seen in Figure 51. A green plume seen in Figure 50 was the first visual indication that there was hardware damage.

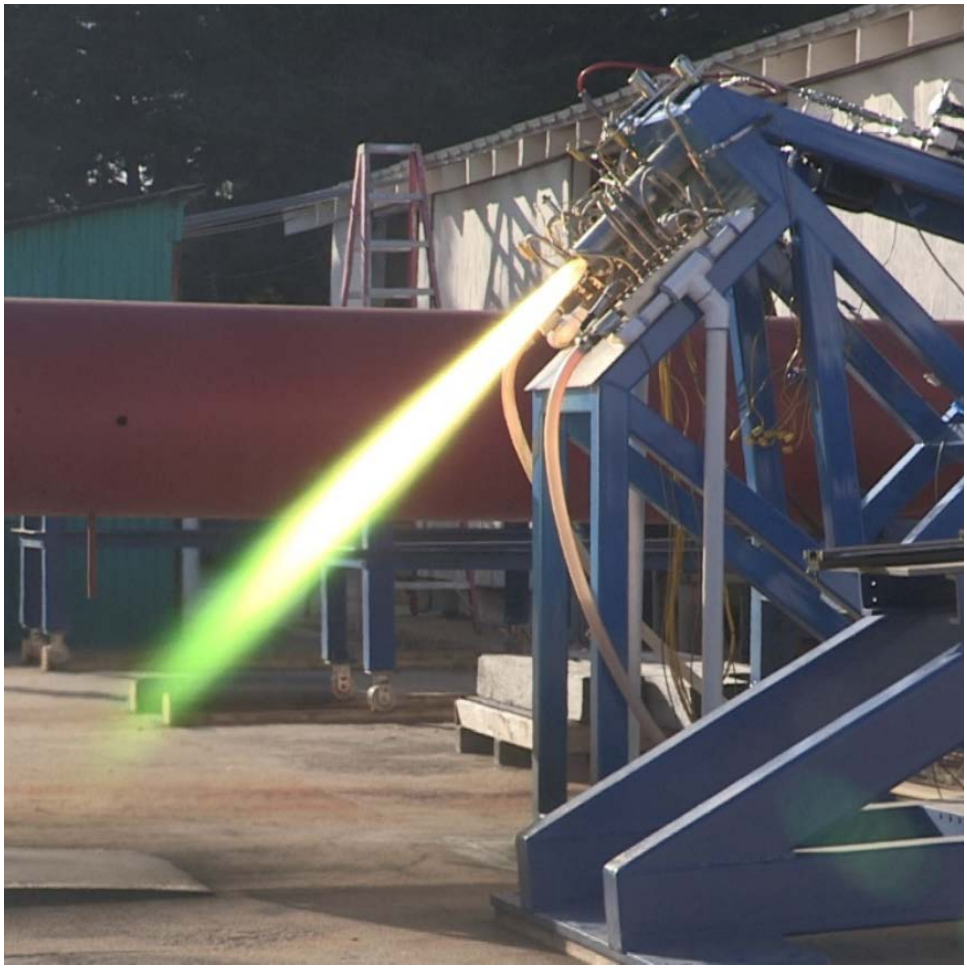


Figure 50. Changing Exhaust Color Indicating Component Failure During Nov21Run01

These conditions caused the nozzle throat and film cooling ring to overheat and melt as shown in Figure 51 and Figure 52, respectively.

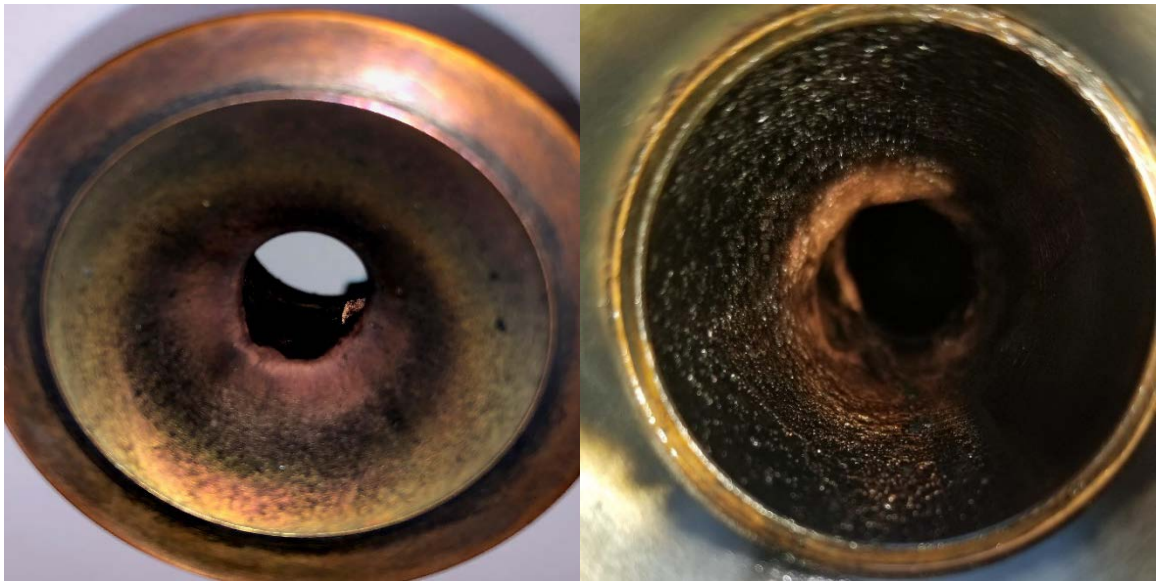


Figure 51. Failed 1.27 cm Nozzle Converging and Diverging Sections

Figure 52. Failed Film Cooling Ring

An examination of the nozzle cross section indicates why the nozzle overheated. In addition to being the component that experiences the highest levels of heat flux (see Figure 3), the nozzle throat is also the furthest from the cooling fluid radially. This is another reason why integrated channels that follow the nozzle contour were incorporated into the new chamber-nozzle system. The material thickness at the original throat was 1.427 cm (0.562 in) versus the new material thickness of 0.103 cm (0.0400 in), as indicated by the blue lines in Figure 53.

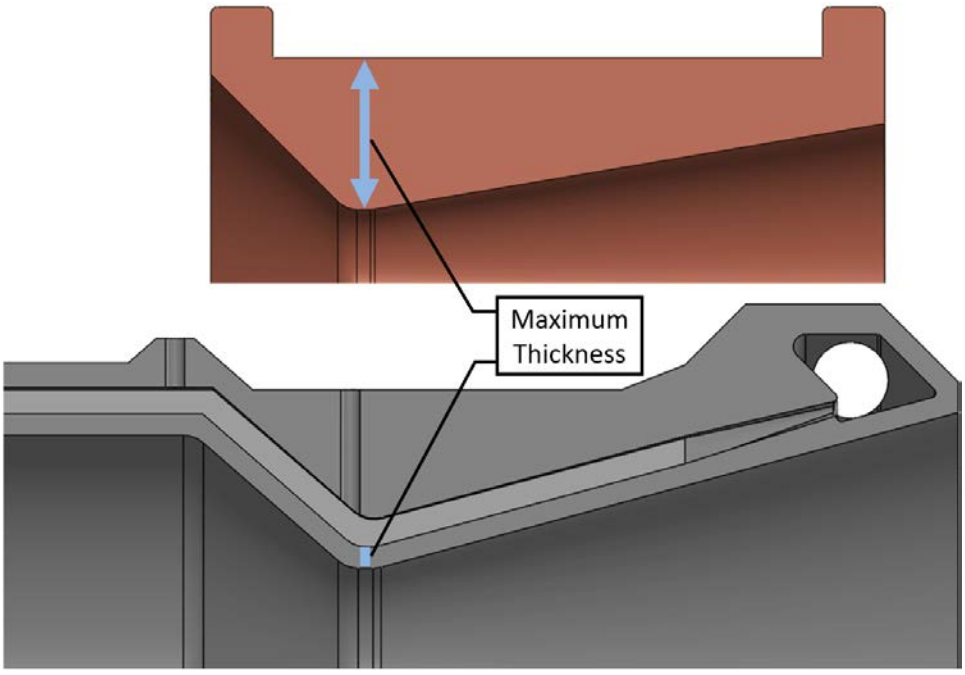


Figure 53. Nozzle Cross Section Comparison

This failure mode clarified three assumptions. First, it indicated that there should be an overall temperature threshold for the film cooling reservoir [1]. Secondly, this showed that the film cooling percentage of 9.59% was evaluated to be too low for the chamber conditions. Lastly, this hardware failure revealed a “pooling” bias of the film cooling reservoir, as the melted portion of the ring was on the top portion of the test stand and experienced less cooling flow. This phenomenon likely would not be observed in a vertically downward test stand configuration.

The second failure mode was the spark plug failure, and was likely not directly related to any one testing parameter being too high or low. During Nov12Run03 the ignitor spark plug cracked and became dislodged from its housing, causing a combustion product leak out of the top of the engine that caused an immediate test abort. The most likely cause of the spark plug failure was fatiguing from repeated load cycles. The solution to this failure mode is to re-design the system to use a more robust and streamlined ignition method. The ignitor failure immediately before and after abort can be seen in Figure 54.

Figure 54. Ignitor Failure During Nov12Run03

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND FUTURE WORK

A test campaign of 33 hot-fire tests provided valuable heat transfer data over operating conditions ranging from 4.03 to 6.96 MPa (585 to 1009 psia) chamber pressure, 1.77 to 2.29 oxidizer-to-fuel mass mixture ratio, and 9.65% to 19.69% film cooling with Cu-101, SS304, SS 17–4PH, and other chamber liner alloy variants. Of those tests, 27 utilized the conventionally manufactured test spools and six utilized the SS 17–4PH additively manufactured test spools.

These tests confirmed the expected heat flux values and the feasibility of utilizing RP-1 for a closed-loop regenerative cooling system. Wall thicknesses from 0.1016–0.2540 cm (0.040–0.100 in) exhibited $\dot{Q}$ values from 2.14–4.29 MW/m^2 (188.4–377.8 Btu/s-ft^2). Specifically, additively manufactured SS 17–4PH spools with wall thicknesses of 0.254 cm (0.100 in) and 0.178 cm (0.070 in) exhibited heat flux values of 3.249 MW/m^2 (286.09 Btu/s-ft^2) and 3.237 MW/m^2 (285.03 Btu/s-ft^2), respectively. Comparing those to conventionally manufactured spools of the same thickness shows that additively manufactured SS 17–4PH is a viable material with which to construct future thrust chamber components if the wall thickness remains between 0.178 cm (0.070 in) and 0.254 cm (0.100 in).

Additionally, an original chamber-nozzle configuration with integrated cooling channels was designed, modeled, and built using additive manufacturing. While it has yet to be hot-fired, the chamber-nozzle was iteratively designed with the aid of CFD, and the increase in fuel temperatures observed in the modeling effort indicate the fuel temperatures should remain within the endothermic limits of RP-1.

Before integration and an eventual flight test, the way ahead for the liquid rocket engine effort is two-fold: closed-loop regenerative cooling must be thoroughly tested; and the engine must be integrated into a single system. Due to restrictions and base closures that were out of the control of lab personnel, fewer tests were conducted with the printed SS 17–4PH test spools and integrated chamber-nozzle than originally planned. Future work should involve more tests with those spools at higher film cooling percentages (>12% film

cooling) with the goal of verifying heat transfer rates and structural integrity. Comparing the thermal conductivity values of SS304 and SS 17–4PH of 16 W/m-K (9.24 BTU/hr-ft-°F) and 18.3 W/m-K (10.57 BTU/hr-ft-°F) respectively, it is evident that printed spools will produce similar data to the machined SS304 spools previously tested, but the internal channel conditions of those printed spools are still poorly characterized.

Unlike the current water-cooled system, RP-1 should be used as a jacket coolant for the spools in an open system, and temperature measured before and after cooling to verify the expected temperature rise. Closed-loop regenerative cooling should only be conducted once that temperature rise and $\dot{Q}$ values are thoroughly understood and predicted. It must be kept in mind that "fresh" (as opposed to pre-heated) RP-1 should be used for film cooling because increased fuel temperature affects film cooling effectiveness. This will hold true for both the spool tests as well as the integrated chamber-nozzle tests. Following these tests, the entire engine's weight should be reduced by integrating the film cooling ring and components forward of it into a single component. Subsequently, this design could be integrated with fuel and oxidizer tanks and the associated plumbing, valves, and turbopumps.

APPENDIX A. STANDARD OPERATING PROCEDURES AND DATA PROCESSING CODE

A. LIQUID ROCKET SOP

Setup Procedures

1. Uncover each outdoor test stand by removing tarp/shed. Place sheds on east side of paved area, well out of the way
2. Verify all valve control lines and pressure transducers/thermocouples desired are connected. Confirm and record venturi tags.
3. Ensure R2DQ Red emergency stop button is pushed IN
4. Yellow warning lights ON and notify lab personnel.
5. Turn on R2DQ in TC#4
    a) PX1e-8381 (top) Verify green power light is ON
    b) Turn on Transducer excitation (red power switch to ON)
    c) Turn on ER5000 power (red power switch to ON)
    d) Turn on Relay power (red power switch to ON)
6. Open and Run LabVIEW control code "TestCell4RocketControl" on R2DQ and enter new test run name.
7. Ensure all pressures are set to zero under "RUN CONDITIONS" (actively zeroes regs)
8. ****** DON SAFETY GLASSES******
9. H2 valves (Located in outside alley—open all valves slowly to prevent hammer shock.)
    a) Open 3 H2 bottles (inside or outside 3) on 1 six-pack
    b) Open H2 isolation valve on six pack manifold
    c) Open Alley BV#3 to supply H2 pressure to header (SLOWLY)
10. O2 valves (Located in outside alley—open all valves slowly to prevent hammer shock.)
    a) OPEN individual O2 bottle valves on both six packs
    b) OPEN O2 six pack isolation valves
    c) OPEN Alley BV#1 (SLOWLY)
    d) OPEN individual O2 bottle valves on HIGH PRESSURE six pack
    e) OPEN O2 HIGH PRESSURE six pack isolation valve
    f) OPEN O2 HIGH PRESSURE BV (SLOWLY)
11. OPEN Shop Air TC#4 wall BV
12. OPEN Shop Air test stand BV
13. N2 valve (2 person job, Located outside TC#3)
    a) TC#4 red emergency button OUT (notify lab personnel and leave walkie-talkie at button in case of emergency)
    b) OPEN N2 bottle valve
    c) In LabView, OPEN N2 Wall Ball Valve then OPEN Fuel Tank N2 Ball Valve
    d) Set hand regulator to desired pressure (wait until pressure drift slows)
    e) In LabView, CLOSE Fuel Tank N2 Ball Valve and N2 Wall Ball Valve
    f) TC#4 red emergency button IN
14. ***CLOSE TC#4 BLAST DOOR (data will be cut off if door is left open)***
15. OPEN O2 TC#4 BV#1 wall BV (SLOWLY)

16. OPEN H2 TC#4 BV#3 wall BV
17. Open S1/F1 cooling tank valves (and S2 for calorimetry)
18. Place 220VAC power knife switch to "ON" (and plug in yellow power cord for calorimetry)
19. Turn on Main Engine Cooling Pump, ensure feed pressure is within 90–170 psi
20. CONNECT engine torch ignition line
21. Set up camera
    a) Turn on TV in control room
    b) Camera on tripod in designated area
    c) Power via extension cord
    d) Flip out screen, set date and time with arrow button, center test stand on screen
    e) Plug in coax cable
    f) HIT RECORD BUTTON on right of camera, near thumb
22. CHECK Shop air tank (closet) (95-120 psi)

Run Sequence

1. **CLOSE TEST CELL#4 Blast Door**
2. LOCK gate
3. Ensure all personnel are in control room (count cars)
4. Golfers CLEAR
5. Water Pump ON
6. Siren ON
7. SET all pressures
    a. O2
    b. H2
    c. N2
8. SET run duration
9. TC#4 red emergency button OUT
10. Start RUN sequence in LabVIEW code

After Run

1. Red emergency stop button IN
2. Siren OFF
3. Plot all data
4. Water OFF when temp hits 200°F

Shutdown Procedure

1. Red emergency stop button IN
2. DISCONNECT torch
3. ATTACH muffler to fuel tank and VENT fuel tank down
4. Turn off/store camera
5. R2DQ DAQ OFF
6. CLOSE water
7. CLOSE H2 torch bottles
8. CLOSE O2 TC#4 BV#1 wall BV
9. CLOSE N2 TC#4 BV#2 wall BV
10. CLOSE all O2 valves
11. CLOSE all N2 valves
12. CLOSE Shop Air TC#4 wall BV
13. CLOSE Shop Air test stand BV
14. CLOSE fuel tank needle valve once fully drained (gauge reads 0 psig)
15. Cover stand with tarp/shed

B. TEST SPOOL ASSEMBLY/DISASSEMBLY SOP

Adapted from [1]: For all procedures, cleaning should be done with Isopropyl Alcohol. Acetone will degrade Viton o-rings and will destroy engine seals. Only use Acetone away from o-rings when necessary. All steps should be completed in order.

1. Assemble Engine Upper Half

This portion of the engine includes the injector through the film cooling ring.

1. Barium grease Viton O-rings for engine expansion spool and slide onto grooves of the expansion spool.
2. Orient expansion spool such that expansion occurs as the flow moves downstream
3. Orient large SS upper engine section on the table such that flow would be moving upward (from the injector to the expansion section)
4. Sit the expansion spool on the large upper engine section and press with level pressure. Firm, level, constant pressure will allow the spool to slowly slide into place. Spool is in place when the o-rings pop into place. Ensure spool does not wedge and o-rings do not get cut.
5. Flip large upper engine section so injector hole is available
6. Select proper injector for chamber pressure condition (3, 4 or 6 hole for 500, 750, or 1000 psia)
7. Slide Barium-greased Viton O-ring into O-ring groove closest to base of injector
8. Grease CalRes O-ring using high-temp Krytox and slide onto injector into groove closest to downstream injector head
9. Carefully slide injector into injector hole on the large upper engine section ensuring not to cut O-rings. Firm pressure will be required to pop O-rings into slots.

10. Bolt injector into place using ball hex driver and injector bolts. Tighten bolts using a star pattern and tighten to the maximum ability of the ball hex driver. (No need to extra tighten with a ratchet)
11. Insert Torch Assembly:
12. Obtain a spare engine test spool
13. Position spare engine test spool in-front of expansion section
14. Use quick-clamp across injector to spare engine spool to apply pressure and compress the expansion section O-rings. The torch assembly extends into the expansion section and the expansion section must be compressed into place for the torch assembly to enter its slot.
15. Slide torch assembly into torch slot. Tighten Swagelok snug to ensure torch gases are isolated within the engine
16. Obtain SS section downstream of expansion section. Orient on table with flow moving upwards
17. Barium grease O-rings and place into internal O-ring groves on downstream section.
18. Add a small bead of barium grease onto O-rings to allow Copper spool to easily slide into place
19. Place copper spool flush onto downstream section and apply firm, constant, level pressure. Spool will pop into place. Ensure not to wedge the spool or cut any O-rings.
20. Barium grease and insert Viton O-rings into face-seal cuts on both large upper engine section and downstream section. Add a small bead of barium grease onto the O-rings in place to ensure O-ring stays in place during assembly.
21. Place film cooling ring onto table such that the fuel film cooling manifold is exposed and flow would be moving into the table.

22. Barium grease Viton o-ring and place into o-ring groove on film cooling ring.

23. Insert dowel pins into downstream section on side without a face seal o-ring (downstream side)

24. Flip downstream section and place onto film cooling ring ensuring proper orientation such that dowel pins enter dowel pin slots on film cooling ring.

25. Repeat the process adding the large upper engine section onto the stack.

26. Bolt the assembly together tightening with a star pattern. Hand tighten with a ball hex-driver. Then tighten with a ratchet to as tight as possible using one hand at the head of the ratchet.

2. Assemble Engine Lower Half

This section includes the upstream spool, the test spool, the nozzle, and the nozzle retention cap.

1. Ensure SS Test Spool is labeled with an arrow for flow direction.

2. Barium grease Viton o-rings and place into internal o-ring grooves on SS spools of both sections (and nozzle if nozzle is disassembled)

3. With clean gloves and using Acetone, clean off any sharpie marker on Test Spool

4. Over the sink, rinse the Test Spool with Isopropyl Alcohol and place on its side to dry

5. Once the Test Spool is dry, sharpie the outside on the water cooling jacket with Test Spool number and an arrow for flow direction (arrow is arbitrary at this point)

6. Place the SS upstream spool section on the table such that flow moves downward.

7. Place the upstream spool onto the SS section ensuring flow directions match. With firm, level, constant pressure, slide the upstream spool into place ensuring not to wedge the spool or cut any o-rings. The spool will drop in after passing the first o-ring and then after repositioning level on the next o-ring will require firm pressure again to be fully seated in the SS jacket.

8. Repeat the procedure for the Test Spool (and Nozzle if required)

9. Barium grease all Viton face seal o-rings and place into face seal grooves of the Nozzle and Test Spool sub-assemblies. Add a small bead of Barium grease onto the face seal o-rings in the grooves to ensure they stay seated during assembly.

10. Place the Upstream Spool sub-assembly on the table such that flow would move upward (o-ring groove down on the table)

11. Insert dowel pins into Test Spool sub-assembly upstream side (side with face seal o-ring)

12. Flip Test Spool sub-assembly and stack onto Upstream Spool sub-assembly ensuring dowel pins seat into their slots.

13. Repeat the procedure for the Nozzle sub-assembly.

14. Barium grease two Viton o-rings for Nozzle cap and place into o-ring grooves. Add a small bead of grease onto o-rings in grooves.

15. Place nozzle cap onto the stack and ensure all the holes line up.

16. With the two small retainer bolts, bolt the stack together. Ensure the two bolts are tightened evenly by alternating a quarter turn for each bolt. Tighten firmly with ball hex driver (only needs to retain the stack together until it is attached to the stand).

17. Flip the stack such that the nozzle is facing downward. Add last Viton face-seal o-ring ensuring it is Barium greased and there is a bead of grease to hold it in place.

18. Add two dowel pins into the upstream spool to fit with the film cooling ring

3. Assemble Engine on the Stand

This includes fixing the upper and lower half of the engine for testing. This job typically requires 2 people.

Disassembly of the Test Spool configuration is achieved by doing this procedure in reverse. If performing successive Test Spool tests, stop after removing the lower half of the engine.

1. Ensure 4 short engine retaining bolts with washers are in position on the engine stand near the O2 line and ratchet with extension is in place.
2. Ensure O2 line cap is removed.
3. Ensure there is a clear path to position the engine on the stand.
4. Ensure small step ladder is positioned next to the engine stand such that a person can easily access the O2 line and the engine retainer bolts.
5. One member must stand in position on the ladder ready to fix the engine to the stand.
6. The other member must take the engine upper section from the assembly table to the stand.
7. Position the engine such that the torch is vertical.
8. Raise the upper section of the engine into the stand such that the injector slides through the injector hole and into the O2 line.
9. The engine holder must keep firm pressure on the engine upper half and keep the face flush with the engine stand.
10. The other member must two upper bolts through the stand to the engine and tighten with the ratchet.

11. The engine holder may then release pressure.

12. The other member must attach the other two bolts and ensure all four bolts are fully and evenly tight.

13. The other member must tighten the O2 swagelok with a 7/8” wrench. Tighten just past snug (any more will damage the Swagelok seal).

14. **Only 1 person is now required** Attach the film cooling line, main fuel line, and chamber pressure transducer to the engine upper half. Swageloks should only be tightened just past snug. Ensure Swagelok of main fuel line at the pressure transducer T is tightened after swinging the line into place.

15. With the 4 long lower section retaining bolts with washers in hand, take the lower section from the assembly table to the engine.

16. Position the lower section to such that the dowel pins fit into the film cooling ring dowel pin slots.

17. Hold pressure and keep the lower section flush with the upper section

18. Attach one upper bolt with washer to the film cooling ring and hand tighten as much as possible.

19. The lower section should be held in place and pressure can be removed.

20. Attach the other three bolts and ensure even tightening with the ratchet. Bolts with washers should be tightened to the maximum ability of the ratchet using one hand at the head of the ratchet.

21. Attach all water cooling lines to the engine. Water inlet lines are attached to the aluminum header on the left side of the engine stand and water outlet lines are attached to the header on the right side of the engine stand. Each spool (except the large upper engine section) has two water ports on the left side of the engine. Water inlet is the upstream water port, water outlet is the downstream water port.

C. DATA PROCESSING CODE

```
% Rocket data analysis
%
% Originally written by Dave Dausen
% Modified by JCodoni April 02, 2018
% Modified by Zachary Lewis November 08, 2018
% Modified by JCodoni Jan 23, 2019
% Modified by JCodoni Feb 1, 2019
% Modified by ZLewis Feb 1, 2019
% Modified by ZLewis Mar 27, 2019
%%%%% Version History
% % % % Ver 0pt... -- Original iterations to organize
% % % %                inputs/outputs/plots.
% % % % Ver 1pt0 -- First implementation version.
% % % % Ver 1pt1 -- added HS thermocouple data.
% % % % Ver 1pt2 -- Consolidated structures into d, in, r, h. Added dialog
% % % %             box asking for time range to average output
% % % %             calculations over.
% % % % Ver 1pt3 -- Fixed error in CStar calculations. Added curve fit for
% % % %             CStar theory. Calculated and output Eta (efficiency).
% % % % Ver 1pt4 -- Excel outputs (Fuel Venturi and Film Cooling) were
% % % %             mixed between psig and psia, edited the outputs to
% % % %             consistently be psia.
% % % % Ver 1pt5 -- Added water discharge and calorimetry pressure to
% % % %             output excel file.
% % % % Ver 1pt6 -- Allowed plotting of the entire high speed file instead
% % % %             of just the first 12 seconds
% % % %          -- Added progress messages and timing for the longer
% % % %             operations
% % % %          -- Changed the figure save operation to save the figure as
% % % %             displayed on the screen with the SCREEN2PNG function
% % % %          -- Modifications by Mark Fernelius (20180424)
% % % % Ver 1pt7 -- High speed temperature plot axis only adjusted if the
% % % %             temperatures are positive (connected)
% % % %          -- Chamber pressure axis scales with max chamber pressure
% % % %          -- Modifications by Mark Fernelius (20180508)
% % % % Ver 2    -- High Speed Temperatures written in excel and plotted as
% % % %             temperatures vice volts
% % % %          -- Modifications by Zachary Lewis
% % % % Ver 2    -- Film Cooling venturi size changed to 0.020” for higher
% % % %             fuel film cooling mdot runs
% % % %          -- Modifications by Zachary Lewis (20181128)
% % % % Ver 2pt1 -- Added fuel manifold pressure data channel and plot
% % % %          -- Changed fuel venturi cal to reflect new install
% % % %          -- Changed fuel film venturi orifice and cal
% % % %          -- Changed O2 venturi orifice diameter
% % % %          -- Improved subplot speed; it had previously been plotting
% % % %             about 2100 * 0.1 points to make the vertical black line
% % % %          -- Added user-input option to decide whether or not to
% % % %             save a data analysis summary
```

```
% % % % Ver 2pt2 -- Changed fuel fuel venturi back to 0.043” diameter on
% % % %             1/28/19
% % % %          -- O2 venturi changed to 0.1378” diameter on 1/28/19
% % % %          -- Divided O2 venturi CD (0.985) by 0.9426, z-value
% % % %             (compressibility factor)
% % % %          -- JCodoni 2/1/2019
% % % % Ver 2pt3 -- High Speed HT Calculated and written directly to excel
% % % %          -- Cooling Water flowrate calculated
% % % %          -- Low Speed Test Spool HT Calculated
% % % %             and written directly to excel
% % % %          -- Low Speed Calorimetry HT Calculated
% % % %             and written directly to excel
% % % %          -- ZALewis 2/19/2019
% % % % Ver 2pt4 -- Cooling water pressures updated to properly calculate
% % % %             water flowrates. This accounted for line pressure drop
% % % %             and converting from psia to psig
% % % %          -- Cooling water flowrate equatoins updated to accurately
% % % %             reflect new calibrations
% % % % Ver 2pt5 -- Fuel film cooling venturi switched from 0.020” to
% % % %             0.0135”
% % % % Ver 2pt6 -- O2 choke changed from 0.1378” to 0.126” for reprocess
% % % %             of 500 psi data
% % % %          -- Updated Test Spool heat Transfer Plots
% % % % Ver 2pt7 -- Switch from Test Spool to Calorimetry Test
% % % %             configuration
% % % %          -- Added average calculations for EngineChamber,
% % % %             QSpool4(TestSpool), and Downstream Calorimetry Spool
% % % %             (Cal Spool is an average of all 4 cooling channels) and
% % % %             Nozzle
% % % %          -- Updated old O2 choke form 0.125 to 0.126” for
% % % %             reprocessing. True choke size was 0.126”
% % % % Ver 2pt8 -- Swapped back to Test Spool to restart re-processing
% % % %          -- O2 Choke = 0.126,” compressibility factor added
% % % %          -- Fuel Venturis = 0.043” and 0.0135” (27MAR19)
% % % % Ver 2pt9 -- Updated for Calorimetry Testing
% % % %          -- Updated Pressure read-ins to correct naming with real
% % % %             location: WaterDischarge 15-->16, CalManifold 16-->15.
% % % % Ver 2pt10-- Changed Film cooling venturi from 0.0135” to 0.020”
% % % %          -- Changed O2 Choke from 0.126” to 0.1378”
% % % % Ver 2pt11-- Changed from Calorimetry to Test Spool Configuration
% % % % Ver 2pt12-- Changed film cooling venturi from 0.020” to 0.0135”
% % % % Ver 2pt13-- Updated for 1000 psi Test Spool Conditions
% % % %          -- Changed O2 from 0.1378” to 0.154” venturi
% % % %          -- Changed main fuel venturi from 0.043” to 0.052”
% % % %          -- Changed film cooling venturi from 0.0135” to 0.020”
% % % % Ver 3    -- ENS Lewis version of code
% % % %          -- Includes cooling water flowrates, heat transfer
% % % %             calculations, and Chamber pressure and HT averages
% % % %          -- Changed Fuel Venturi from 0.052” to 0.043”
% % % % Ver 3pt1 -- Changed Film Cooling Venturi from 0.020” to 0.0135”
% % % % Ver 3pt2 -- Added correction factor for LS heat flux to account for
% % % %             F to C, multiplied all by 5/9
```

```
% % % %          -- Corrected Calorimetry Area for heat flux from 13.7/4 to
% % % %             3.534 in^2 (05APR19)
% % % % Ver 3pt3 -- Changed fuel venturi from 0.043" to 0.052"
% % % %          -- Changed Film Cooling venturi from 0.0135" to 0.020"
% % % %          -- Changed Nozzle water cooling flowrate cal for
% % % %             calorimetry testing
% % % %          -- Changed fuel venturi from 0.052" to 0.043"
% % % %          -- Changed fuel venturi from 0.043" to 0.052"
% % % %          -- Changed film cooling venturi from 0.020" to 0.0135"
% % % % Ver 3pt4 -- Modified Spool temp calculations to C
% % % %          -- Adjusted plots to reflect SI units
% % % %          -- Modified by Mo Trent (1MAY20)
%
%
%   DATA FORMATTING STRUCTURE
%   d -- data
%   in -- inputs
%   r -- raw data read-ins and calculation constants/variables.
%   h -- handles
%
% SECTIONS: (Can use ctrl + F to go to sections)
%           SECTION01  -- User-defined inputs (default file loc, nozzle,
%           etc.)
%           SECTION02  -- Gas constants
%           SECTION03  -- Import/organize pressure data
%           SECTION04  -- Import/organize temperature data (incl HS)
%           SECTION05  -- Assign pressure data to structure
%           SECTION06  -- Assign temperature data to structure (incl HS)
%           SECTION07  -- Mass flow rate calculations
%           SECTION08  -- Create and show the plots
%           SECTION09  -- Dialog asking for time frame to average over
%           SECTION10 -- Calculate flow averages
%           SECTION11 -- Save Matlab workspace
%           SECTION12 -- Add the average location lines to prev plots
%           SECTION13 -- Save the plots, Excel workbook
%
% Rocket Pressure Channels -- hard coded in SECTION 5
% 1. Time
% 2. O2FeedbackPress
% 3. N2FeedbackPress
% 4. H2FeedbackPress
% 5. C2H4FeedbackPress
% 6. EngineO2ChokePress  %Venturi used not Choke
% 7. TorchO2Press
% 8. FuelTankN2Press
% 9. TorchH2Press
% 10. FuelVenturiPress
% 11. ManifoldO2Press
% 12. EngineChamberPress
% 13. FilmCoolingPress
% 14. WaterCoolingInletManifoldpsigPress
% 15. WaterDischargePress
```

```
% 16. CalInletManifoldPsig

% Rocket Temperature Channels -- hard coded in SECTION 6
% 1. Time
% 2. Spool1Temp
% 3. Spool2Temp
% 4. Spool3Temp
% 5. Spool4Temp
% 6. NozzleTemp
% 7. WaterManifoldTemp
% 8. FilmCoolingTemp
% 9. CalorimetryInletTemp
% 10. T4
% 11. T5
% 12. T6
% 13. T7
% 14. T8
% 15. T9
% 16. T10
% 17. T11

% Rocket Temperature High Speed Channels -- hard coded in SECTION 6
% 1. 1A
% 2. 1B
% 3. 2A
% 4. 2B
% 5. 3A
% 6. 3B
% 7. 4A
% 8. 4B
% 9. 5A
% 10. 5B
% 11. 6A
% 12. 6B
% 13. 7A
% 14. 7B
% 15. 8A
% 16. 8B

close all;clear all;clc
```

%% User-defined inputs for selecting the data file to be analyzed.

%% SECTION01

```
% Run time period to average over default inputs
d.Averages.StartTime = 15;                                          %Seconds
RunTime average for Mar07Run01
d.Averages.EndTime = 25;                                            %Seconds

% Select Run Data folder default location
display('Choose the run folder to analyze.');
```

```
in.loc = uigetdir('C:\RocketData2019\');
% in.loc = uigetdir('C:\Users\Titan\Desktop\Reprocess 27MAR19\RocketData2019\');
drawnow;pause(0.05);

in.data_summary = questdlg('Would you like to create a data analysis summary?',...
    'Data summary','Yes','No','Yes');

% Venturi information
% Oxidizer
%{
% Old choke info
d.Averages.O2VenturiDiam = 0.126;                                    % inch,
Calculated using Isentropic relations
d.Averages.NumO2Chokes = 1;
d.Averages.O2Cd = 0.985/0.9426;                                          %
Discharge coefficient, conservative value
%}
%
d.Averages.O2VenturiDiam = 0.154;                                    % inch,
Calculated using Isentropic relations
d.Averages.NumO2Chokes = 1;
d.Averages.O2Cd = 0.985/0.9426;                                          %
Discharge coefficient, conservative value
%}

%{
d.Averages.O2VenturiDiam = 0.1378;                                   % inch,
Calculated using Isentropic relations
d.Averages.NumO2Chokes = 1;
d.Averages.O2Cd = 0.985/0.9426;                                          %
Discharge coefficient, conservative value, divided by compressibility (z-value) factor
%}

% Fuel
%{
%Calculated using Empiracal measurements
% d.Averages.FuelVenturiDiam = 0.043;                                  OLD CAL PRE
JAN 23 2019 %inch P = 22193*(Mdot, kg/s) - 655.17
% d.Averages.FuelFilmVenturiDiam = 0.0135;                             OLD CAL PRE
JAN 23 2019 %inch P = 204918*(Mdot, kg/s) - 621.00
%}
%
d.Averages.FuelVenturiDiam = 0.052;                                  % 1/25/2019
inch P = 16226.9*(Mdot, kg/s) - 788.9
% d.Averages.FuelFilmVenturiDiam = 0.020;                              % 1/25/2019
inch P = 113890*(Mdot, kg/s) - 1016.6
%}

%
% d.Averages.FuelVenturiDiam = 0.043;                                  % 1/28/2019
inch P = 22193*(Mdot, kg/s) - 655.17
% d.Averages.FuelFilmVenturiDiam = 0.020;                              % 1/25/2019
```

```
inch P = 113890*(Mdot, kg/s) - 1016.6
% d.Averages.FuelFilmVenturiDiam = 0.0135;                                    % OLD CAL
PRE JAN 23 2019 %inch P = 204918*(Mdot, kg/s) - 621.00
d.Averages.FuelFilmVenturiDiam = 0.024;                                      % P =
104440*(mdot)+2184.4
%}

% Nozzle Throat
d.Averages.NozzleDiam = 0.500;                                               %inch

in.loc = strcat(in.loc,'\');
in.Pressure = dir(strcat(in.loc,'\*Pressure*.txt'));
in.temperature = dir(strcat(in.loc,'\*Temp*.txt'));
in.HighSpeed = dir(strcat(in.loc,'\*HighSpeed*.txt'));                       % High speed
temperature.
```

%% Gas constants

%% SECTION02

```
r.RPrime = 8314.4621;                                                        % (kJ/(Mol
K))
r.GammaO2 = 1.397;
r.RO2 = (r.RPrime/31.9988);                                                  % J/(Kg K)
r.GammaH2 = 1.412;
r.RH2 = (r.RPrime/2.016);                                                    % J/(Kg K)

r.O2ChokeArea = (((d.Averages.O2VenturiDiam/2)*0.0254)^2*pi);                % m^2
r.NozzleChokeArea = (((d.Averages.NozzleDiam/2)*0.0254)^2*pi);               % m^2

disp(['Oxygen Venturi (inch): ' num2str(d.Averages.O2VenturiDiam)])
disp(['Fuel Venturi (inch): ' num2str(d.Averages.FuelVenturiDiam)])
disp(['Fuel Film Cooling Venturi (inch): ' num2str(d.Averages.FuelFilmVenturiDiam)])
disp(['Nozzle Throat(inch): ' num2str(d.Averages.NozzleDiam)])
```

%% Import and organize Pressure data

%% SECTION03

```
r.LowSpeedOrigPress = importdata(strcat(in.loc,in.Pressure.name));
r.LowSpeedTimePress = r.LowSpeedOrigPress.textdata;
r.LowSpeedRawDataPress = r.LowSpeedOrigPress.data;

r.DataPointsPress = length(r.LowSpeedTimePress);

r.LowSpeedRawDataPress = r.LowSpeedRawDataPress(1:r.DataPointsPress,:);      % Ensure all
data is same length

r.TimeSecPress = zeros(r.DataPointsPress,1);
r.TimeStampPress = char(r.LowSpeedTimePress{1:r.DataPointsPress});
for K = 1:r.DataPointsPress
    r.TimeSecPress(K,1) = str2double(r.TimeStampPress(K,end-8:end))';
```

```
    if r.TimeSecPress(K) < r.TimeSecPress(1);
        r.TimeSecPress(K) = r.TimeSecPress(K) + 60;
    end
end

r.TimeNormPress = r.TimeSecPress - r.TimeSecPress(1);                    % Normalize
time to start at 0 seconds

r.LowSpeedPressure = horzcat(r.TimeNormPress, r.LowSpeedRawDataPress);   % Concatenate
time and data
r.LowSpeedPressure = sortrows(r.LowSpeedPressure,1);                     % Sort is
completed by time places NaN at end
r.LowSpeedPressure = r.LowSpeedPressure(...
    isfinite(r.LowSpeedPressure(:,1)),:);                                % Delete any
NaN

r.PressureNaN = isnan(r.LowSpeedPressure);
r.NaNPress = find(r.PressureNaN(:,:)==1);
r.NaNPress = isempty(r.NaNPress);
if r.NaNPress==0
    display('Bad P Data')
end
```

%% Import and organize temperature data

%% SECTION04

```
r.LowSpeedOrigTemp = importdata(strcat(in.loc,in.temperature.name));
r.LowSpeedTimeTemp = r.LowSpeedOrigTemp.textdata;
r.LowSpeedRawDataTemp = r.LowSpeedOrigTemp.data;

r.DataPointsTemp = length(r.LowSpeedTimeTemp);

r.LowSpeedRawDataTemp = r.LowSpeedRawDataTemp(1:r.DataPointsTemp,:);     % Ensure all
data is same length

r.TimeSecTemp = zeros(r.DataPointsTemp,1);
r.TimeStampTemp = char(r.LowSpeedTimeTemp{1:r.DataPointsTemp});
for K = 1:r.DataPointsTemp
    r.TimeSecTemp(K,1) = str2double(r.TimeStampTemp(K,end-8:end))';
    if r.TimeSecTemp(K) < r.TimeSecTemp(1)                               % Adjusts the
seconds if clock overlaps from 59 to 00
        r.TimeSecTemp(K) = r.TimeSecTemp(K) + 60;
    end
end

r.TimeNormTemp = r.TimeSecTemp - r.TimeSecTemp(1);                       % Normalize
time to start at 0 seconds

r.Temperature = horzcat(r.TimeNormTemp, r.LowSpeedRawDataTemp);          % Concatenate
time and data
r.Temperature = sortrows(r.Temperature,1);                               % Sort is
```

```
completed by time places NaN at end
r.Temperature = r.Temperature(isfinite(r.Temperature(:,1)),:);

r.TemperatureNaN = isnan(r.Temperature);
r.NaNTemp = find(r.TemperatureNaN(:,:)==1);
r.NaNTemp = isempty(r.NaNTemp);
if r.NaNTemp==0
    display('Bad T Data')
end

% Import high speed temperature data
r.HighSpeedTemp = importdata(strcat(in.loc,in.HighSpeed.name));
```

%% Setup Pressure data structure headers

%% SECTION05

```
d.Pressure.Time = r.LowSpeedPressure(:,1);                                  % seconds
d.Pressure.O2Feedback = r.LowSpeedPressure(:,2)+14.7;                       % psia
d.Pressure.N2Feedback = r.LowSpeedPressure(:,3)+14.7;                       % psia
d.Pressure.O2EngineChoke = r.LowSpeedPressure(:,6)+14.7;                    % psia
d.Pressure.O2Torch = r.LowSpeedPressure(:,7)+14.7;                          % psia
d.Pressure.N2FuelTank = r.LowSpeedPressure(:,8)+14.7;                       % psia
d.Pressure.H2Torch = r.LowSpeedPressure(:,9)+14.7;                          % psia
d.Pressure.FuelVenturi = r.LowSpeedPressure(:,10);                          % Fuel
Venturi uses psig as input for calibration
d.Pressure.O2Manifold = r.LowSpeedPressure(:,11)+14.7;                      % psia
d.Pressure.EngineChamber = r.LowSpeedPressure(:,12)+14.7;                   % psia
d.Pressure.FilmCooling = r.LowSpeedPressure(:,13);                          % Fuel Film
Cooling Venturi uses psig as input for calibration
d.Pressure.WaterCoolingInletManifoldpsig = r.LowSpeedPressure(:,14);        % Test Spool
Inlet Manifold uses psig as input to calculate cooling water flowrate
d.Pressure.WaterDischargePressure = r.LowSpeedPressure(:,16);               % psig
d.Pressure.CalInletManifoldPsig = r.LowSpeedPressure(:,15);                 % Calorimetry
Inlet Manifold uses psig as input to calculate cooling water flowrate
d.Pressure.FuelManifold = r.LowSpeedPressure(:,5) + 14.7;                   % psia; note
-- stole the AI data line that was previously used for the C2H4 tescom pressure, AI3,
line label "MP4"
```

%% Setup high and low speed temperature data structure headers

%% SECTION06 Low speed temperature, deg-F

```
d.TempLS.Time = r.Temperature(:,1);
d.TempLS.Spool1 = r.Temperature(:,2);
d.TempLS.Spool2 = r.Temperature(:,3);
d.TempLS.Spool3 = r.Temperature(:,4);
d.TempLS.Spool4 = r.Temperature(:,5);
d.TempLS.Nozzle = r.Temperature(:,6);
d.TempLS.WaterManifold = r.Temperature(:,7);
d.TempLS.FilmCooling = r.Temperature(:,8);
d.TempLS.CalorimetryInlet = r.Temperature(:,9);
```

```
d.TempLS.T4  = r.Temperature(:,10);
d.TempLS.T5  = r.Temperature(:,11);
d.TempLS.T6  = r.Temperature(:,12);
d.TempLS.T7  = r.Temperature(:,13);
d.TempLS.T8  = r.Temperature(:,14);
d.TempLS.T9  = r.Temperature(:,15);
d.TempLS.T10 = r.Temperature(:,16);
d.TempLS.T11 = r.Temperature(:,17);


% HS Thermocouple data
% Output is in Degrees Celcius
% d.TempHS.Time = (0:1/5000:12-1/5000)';                                    % HS
thermocouple data is sampled at 5 kHz for 60,000 samples.
[HighSpeedLength, HighSpeedWidth] = size(r.HighSpeedTemp);                % Get the
length and width of the high speed data matrix
d.TempHS.Time = (0:1/5000:HighSpeedLength*1/5000-1/5000)';                % Changed the
time matrix to include the whole high speed file and not just the first 12 seconds
d.TempHS.Thermo1A = 0.95039.*(r.HighSpeedTemp(:,1).^3) ...
    -10.715.*(r.HighSpeedTemp(:,1).^2) ...
    +124.02.*r.HighSpeedTemp(:,1) +0.83168;
d.TempHS.Thermo1B = 0.95039.*(r.HighSpeedTemp(:,2).^3) ...
    -10.715.*(r.HighSpeedTemp(:,2).^2) ...
    +124.02.*r.HighSpeedTemp(:,2) +0.83168;
d.TempHS.Thermo2A = 0.95039.*(r.HighSpeedTemp(:,3).^3) ...
    -10.715.*(r.HighSpeedTemp(:,3).^2) ...
    +124.02.*r.HighSpeedTemp(:,3) +0.83168;
d.TempHS.Thermo2B = 0.95039.*(r.HighSpeedTemp(:,4).^3) ...
    -10.715.*(r.HighSpeedTemp(:,4).^2) ...
    +124.02.*r.HighSpeedTemp(:,4) +0.83168;
d.TempHS.Thermo3A = 0.95039.*(r.HighSpeedTemp(:,5).^3) ...
    -10.715.*(r.HighSpeedTemp(:,5).^2) ...
    +124.02.*r.HighSpeedTemp(:,5) +0.83168;
d.TempHS.Thermo3B = 0.95039.*(r.HighSpeedTemp(:,6).^3) ...
    -10.715.*(r.HighSpeedTemp(:,6).^2) ...
    +124.02.*r.HighSpeedTemp(:,6) +0.83168;
d.TempHS.Thermo4A = 0.95039.*(r.HighSpeedTemp(:,7).^3) ...
    -10.715.*(r.HighSpeedTemp(:,7).^2) ...
    +124.02.*r.HighSpeedTemp(:,7) +0.83168;
d.TempHS.Thermo4B = 0.95039.*(r.HighSpeedTemp(:,8).^3) ...
    -10.715.*(r.HighSpeedTemp(:,8).^2) ...
    +124.02.*r.HighSpeedTemp(:,8) +0.83168;
d.TempHS.Thermo5A = 0.95039.*(r.HighSpeedTemp(:,9).^3) ...
    -10.715.*(r.HighSpeedTemp(:,9).^2) ...
    +124.02.*r.HighSpeedTemp(:,9) +0.83168;
d.TempHS.Thermo5B = 0.95039.*(r.HighSpeedTemp(:,10).^3) ...
    -10.715.*(r.HighSpeedTemp(:,10).^2) ...
    +124.02.*r.HighSpeedTemp(:,10) +0.83168;
d.TempHS.Thermo6A = 0.95039.*(r.HighSpeedTemp(:,11).^3) ...
    -10.715.*(r.HighSpeedTemp(:,11).^2) ...
    +124.02.*r.HighSpeedTemp(:,11) +0.83168;
d.TempHS.Thermo6B = 0.95039.*(r.HighSpeedTemp(:,12).^3) ...
```

```
    -10.715.*(r.HighSpeedTemp(:,12).^2) ...
    +124.02.*r.HighSpeedTemp(:,12) +0.83168;
d.TempHS.Thermo7A = 0.95039.*(r.HighSpeedTemp(:,13).^3) ...
    -10.715.*(r.HighSpeedTemp(:,13).^2) ...
    +124.02.*r.HighSpeedTemp(:,13) +0.83168;
d.TempHS.Thermo7B = 0.95039.*(r.HighSpeedTemp(:,14).^3) ...
    -10.715.*(r.HighSpeedTemp(:,14).^2) ...
    +124.02.*r.HighSpeedTemp(:,14) +0.83168;
d.TempHS.Thermo8A = 0.95039.*(r.HighSpeedTemp(:,15).^3) ...
    -10.715.*(r.HighSpeedTemp(:,15).^2) ...
    +124.02.*r.HighSpeedTemp(:,15) +0.83168;
d.TempHS.Thermo8B = 0.95039.*(r.HighSpeedTemp(:,16).^3) ...
    -10.715.*(r.HighSpeedTemp(:,16).^2) ...
    +124.02.*r.HighSpeedTemp(:,16) +0.83168;


% d.TempHS.Thermo1A = 0.95039.*(r.HighSpeedTemp(:,9).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,9).^2) ...
%     +124.02.*r.HighSpeedTemp(:,9) +0.83168;                                  %
Calorimetry spools were switched. If desired to keep 1A as the label for the upstream
thermocouple, use this setup
% d.TempHS.Thermo1B = 0.95039.*(r.HighSpeedTemp(:,10).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,10).^2) ...
%     +124.02.*r.HighSpeedTemp(:,10) +0.83168;
% d.TempHS.Thermo2A = 0.95039.*(r.HighSpeedTemp(:,11).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,11).^2) ...
%     +124.02.*r.HighSpeedTemp(:,11) +0.83168;
% d.TempHS.Thermo2B = 0.95039.*(r.HighSpeedTemp(:,12).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,12).^2) ...
%     +124.02.*r.HighSpeedTemp(:,12) +0.83168;
% d.TempHS.Thermo3A = 0.95039.*(r.HighSpeedTemp(:,13).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,13).^2) ...
%     +124.02.*r.HighSpeedTemp(:,13) +0.83168;
% d.TempHS.Thermo3B = 0.95039.*(r.HighSpeedTemp(:,14).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,14).^2) ...
%     +124.02.*r.HighSpeedTemp(:,14) +0.83168;
% d.TempHS.Thermo4A = 0.95039.*(r.HighSpeedTemp(:,15).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,15).^2) ...
%     +124.02.*r.HighSpeedTemp(:,15) +0.83168;
% d.TempHS.Thermo4B = 0.95039.*(r.HighSpeedTemp(:,16).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,16).^2) ...
%     +124.02.*r.HighSpeedTemp(:,16) +0.83168;
% d.TempHS.Thermo5A = 0.95039.*(r.HighSpeedTemp(:,1).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,1).^2) ...
%     +124.02.*r.HighSpeedTemp(:,1) +0.83168;
% d.TempHS.Thermo5B = 0.95039.*(r.HighSpeedTemp(:,2).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,2).^2) ...
%     +124.02.*r.HighSpeedTemp(:,2) +0.83168;
% d.TempHS.Thermo6A = 0.95039.*(r.HighSpeedTemp(:,3).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,3).^2) ...
%     +124.02.*r.HighSpeedTemp(:,3) +0.83168;
% d.TempHS.Thermo6B = 0.95039.*(r.HighSpeedTemp(:,4).^3) ...
```

```
%     -10.715.*(r.HighSpeedTemp(:,4).^2) ...
%     +124.02.*r.HighSpeedTemp(:,4) +0.83168;
% d.TempHS.Thermo7A = 0.95039.*(r.HighSpeedTemp(:,5).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,5).^2) ...
%     +124.02.*r.HighSpeedTemp(:,5) +0.83168;
% d.TempHS.Thermo7B = 0.95039.*(r.HighSpeedTemp(:,6).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,6).^2) ...
%     +124.02.*r.HighSpeedTemp(:,6) +0.83168;
% d.TempHS.Thermo8A = 0.95039.*(r.HighSpeedTemp(:,7).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,7).^2) ...
%     +124.02.*r.HighSpeedTemp(:,7) +0.83168;
% d.TempHS.Thermo8B = 0.95039.*(r.HighSpeedTemp(:,8).^3) ...
%     -10.715.*(r.HighSpeedTemp(:,8).^2) ...
%     +124.02.*r.HighSpeedTemp(:,8) +0.83168;
```

%% Calculations for mass flow rates

%% SECTION07

```
% O2 Engine choke
r.O2EngineChokePressSI = d.Pressure.O2EngineChoke*6894.757;
r.O2EngineTempSI = 288;                                                    % K
% Fuel engine choke
r.FuelVenturiPressSI = d.Pressure.FuelVenturi*6894.757;                    % Not Needed
as calibration used psig
r.FuelVenturiTempSI = 288;                                                 % K

% Setup time vector for mass flow calculations.
d.calcs.Time = d.Pressure.Time;

% Oxygen mass flow calculation
d.calcs.O2Mdot = d.Averages.O2Cd*d.Averages.NumO2Chokes*((r.O2ChokeArea.*...
    r.O2EngineChokePressSI./sqrt(r.O2EngineTempSI)*...
    sqrt(r.GammaO2/r.RO2*(2/(r.GammaO2+1))^((r.GammaO2+1)/...
    (r.GammaO2-1)))));

% Fuel mass flow calculation
if d.Averages.FuelVenturiDiam == 0.043
    disp('Using 0.043" Fuel Venturi Cal')
d.calcs.FuelMdot = (d.Pressure.FuelVenturi + 655.17)/22193;                % 1/28/2019
For 0.043" Venturi P = 22193*(Mdot, kg/s) - 655.17, Input is Psig
else
    disp('Using 0.052" Fuel Venturi Cal')
d.calcs.FuelMdot = (d.Pressure.FuelVenturi + 788.9)/16226.9;                % For 0.052"
Venturi P = 16226.9*(Mdot, kg/s) - 788.9, Input is Psig
end

% Film cooling mass flow calculation
if d.Averages.FuelFilmVenturiDiam == 0.0135
    disp('Using 0.0135" Film Cooling Venturi Cal')
d.calcs.FuelFilmCoolingMdot = (d.Pressure.FilmCooling + 621.00)/204918;    %OLD, pre- 23
Jan 2019 % Same Pressure for both Venturi, Old P = 202281*(Mdot) + 621, for d=0.0135"
```

```
venturi

elseif d.Averages.FuelFilmVenturiDiam == 0.020
    disp('Using 0.020" Film Cooling Venturi Cal')
d.calcs.FuelFilmCoolingMdot = (d.Pressure.FilmCooling + 1016.6)/113890;     % Same
Pressure for both Venturi, Old P = 113890*(Mdot) + 1016.6, for d=0.020" venturi

else
    disp('Using 0.024" Film Cooling Venturi Cal')
d.calcs.FuelFilmCoolingMdot = (d.Pressure.FilmCooling + 2184.4)/104440;     % Cal
Conducted on 19APR2019, P = 104440*(mdot)+2184.4
end

% Oxygen-fuel ratio calculation in the core
d.calcs.OFCore = d.calcs.O2Mdot ./ d.calcs.FuelMdot;

% Oxygen-fuel ratio calculation for the overall flowfield
d.calcs.OFOverall = d.calcs.O2Mdot ./ ...
    (d.calcs.FuelMdot + d.calcs.FuelFilmCoolingMdot);

% Experimental C-star, core
d.calcs.CStarCore = d.Pressure.EngineChamber / 14.7 * 101300 ...
    * r.NozzleChokeArea ./ (d.calcs.FuelMdot + d.calcs.O2Mdot);             % m/s

% Experimental C-star, overall
d.calcs.CStarOverall = d.Pressure.EngineChamber / 14.7 * 101300 * ...
    r.NozzleChokeArea ./ (d.calcs.FuelMdot + ...
    d.calcs.O2Mdot + d.calcs.FuelFilmCoolingMdot);      % m/s

% C-Star theory calcs, based on third order fit to CEA (not frozen)
d.calcs.CStarTheoryCore = 199.94 * d.calcs.OFCore .^ 3 - 1464.2 * ...
    d.calcs.OFCore .^ 2 + 3500.6 * d.calcs.OFCore - 954.59;                 % C-star fit
= 199.94 * OF ^ 3 - 1464.2 * OF ^ 2 + 3500.6 * OF - 954.59

% C-Star theory for overall flow
d.calcs.CStarTheoryOverall = 199.94 * d.calcs.OFOverall .^ 3 - 1464.2 * ...
    d.calcs.OFOverall .^ 2 + 3500.6 * d.calcs.OFOverall - 954.59;

% Efficiency calculation in the core and overall
d.calcs.EtaCore = d.calcs.CStarCore ./ d.calcs.CStarTheoryCore;
d.calcs.EtaOverall = d.calcs.CStarOverall ./ d.calcs.CStarTheoryOverall;

%----- Convert Pressures to deltaPs

d.Pressure.CalSpoolInletPsig = d.Pressure.CalInletManifoldPsig - 6.5;       % psig,
accounts for pressure drop from manifold to spool inlet
d.Pressure.CalSpoolDeltaP = 0.906360*d.Pressure.CalSpoolInletPsig ...
                            - 0.132509;

d.Pressure.TestSpoolDeltaP = 0.476737 ...
    *d.Pressure.WaterCoolingInletManifoldpsig + 2.893078;                   % psi, Test
Spool Configuration Spool deltaP calibration
```

```
% %---------- If Running Calorimetry
d.Pressure.NozzleDeltaP = 0.445080 ...
    *d.Pressure.WaterCoolingInletManifoldpsig + 2.989997;                    % psi,
Calorimetry Spool Configuration nozzle deltaP calibration

%---------- If Running Test Spools
% d.Pressure.NozzleDeltaP = 0.473661 ...
%     *d.Pressure.WaterCoolingInletManifoldpsig + 2.028298;                    % psi, Test
Spool Configuration Nozzle deltaP calibration

%----- Calorimetry Water Cooling mdot
d.calcs.CalorimetryWaterMdot = 3.785*(0.0219*(d.Pressure.CalSpoolDeltaP)...
    + 1.2721)/60;                                                            % Kg/s =
3.785*(0.0219*(deltaP)+1.2721)/60

%----- Test Spool Water Cooling mdot
d.calcs.TestSpoolWaterMdot = 0.002967*(d.Pressure.TestSpoolDeltaP) ...
                           +0.158799;                                        % kg/s =
0.002967(deltaP)+0.158799

%----- Nozzle Water Cooling mdot

d.calcs.NozzleWaterMdot = 0.003038*(d.Pressure.NozzleDeltaP) ...
                           +0.163580;                                      % kg/s =
0.003038(deltaP)+0.163580

% Convert Fuel Venturi and Film Cooling from psig to psia.
d.Pressure.FuelVenturi = d.Pressure.FuelVenturi + 14.7;
d.Pressure.FilmCooling = d.Pressure.FilmCooling + 14.7;
```

%% Calculations for Heat Transfer

```
%--------- HS Heat Transfer Calcs

x.constants.tw = 0.00254;                                                    % TC recess
distance between ThermoA and ThermoB
x.constants.k = 391;                                                         % Copper
Conductivity
x.HSHT.Time = d.TempHS.Time;                                                 % Time
reference for sheet
x.HSHT.Q1 = (d.TempHS.Thermo1A-d.TempHS.Thermo1B) ...
    .*x.constants.k./x.constants.tw;                                         % Q/A = k/
tw*deltaT
x.HSHT.Q2 = (d.TempHS.Thermo2A-d.TempHS.Thermo2B) ...
    .*x.constants.k./x.constants.tw;
x.HSHT.Q3 = (d.TempHS.Thermo3A-d.TempHS.Thermo3B) ...
    .*x.constants.k./x.constants.tw;
x.HSHT.Q4 = (d.TempHS.Thermo4A-d.TempHS.Thermo4B) ...
    .*x.constants.k./x.constants.tw;
x.HSHT.Q5 = (d.TempHS.Thermo5A-d.TempHS.Thermo5B) ...
```

```
    .*x.constants.k./x.constants.tw;
x.HSHT.Q6 = (d.TempHS.Thermo6A-d.TempHS.Thermo6B) ...
    .*x.constants.k./x.constants.tw;
x.HSHT.Q7 = (d.TempHS.Thermo7A-d.TempHS.Thermo7B) ...
    .*x.constants.k./x.constants.tw;
x.HSHT.Q8 = (d.TempHS.Thermo8A-d.TempHS.Thermo8B) ...
    .*x.constants.k./x.constants.tw;

%---------- LS Heat Transfer Calcs

% Test Spool Mdot

x.calcs.TestSpoolWaterMdot = interp1(d.calcs.Time, ...
    d.calcs.TestSpoolWaterMdot, d.TempLS.Time,'pchip');

x.calcs.NozzleWaterMdot = interp1(d.calcs.Time, ...
    d.calcs.NozzleWaterMdot, d.TempLS.Time,'pchip');

% Test Spool Sizes

x.constants.cpWater = 4185.5;                                          % J/kg-k
x.constants.spoolLength = 2*0.0254;                                    % 2 in spool
length
x.constants.spoolDiam = 1.5*0.0254;                                    % 1.5 in
spool diameter
x.constants.SAspool = pi*x.constants.spoolDiam*x.constants.spoolLength;

% x.constants.NozzleLength = 2.5*0.0254;
% x.constants.NozzleDiam = 1.5*0.0254;                                   % NEED TO
CHECK VALUE
% x.constants.SANozzle = pi*x.constants.NozzleDiam*x.constants.NozzleLength;
x.constants.SANozzle = 12.2093*(0.0254^2);                             % From
SolidWorks model

%----- Test Spool HT

x.LSHT.TestSpool.Time = d.TempLS.Time;
x.LSHT.TestSpool.QSpool1 = (5/9)*x.calcs.TestSpoolWaterMdot.*x.constants.cpWater ...
    .* (d.TempLS.Spool1-d.TempLS.WaterManifold)./x.constants.SAspool;
x.LSHT.TestSpool.QSpool2 = (5/9)*x.calcs.TestSpoolWaterMdot.*x.constants.cpWater ...    %
Q/A = W/m^2
    .* (d.TempLS.Spool2-d.TempLS.WaterManifold)./x.constants.SAspool;
x.LSHT.TestSpool.QSpool3 = (5/9)*x.calcs.TestSpoolWaterMdot.*x.constants.cpWater ...    %
Q/A = mdot*Cp*deltaT/SAspool
    .* (d.TempLS.Spool3-d.TempLS.WaterManifold)./x.constants.SAspool;
x.LSHT.TestSpool.QSpool4 = (5/9)*x.calcs.TestSpoolWaterMdot.*x.constants.cpWater ...
    .* (d.TempLS.Spool4-d.TempLS.WaterManifold)./x.constants.SAspool;
x.LSHT.TestSpool.QNozzle = (5/9)*x.calcs.NozzleWaterMdot.*x.constants.cpWater ...
    .* (d.TempLS.Nozzle-d.TempLS.WaterManifold)./x.constants.SANozzle;


% Calorimetry Mdot
```

```
x.calcs.CalorimetryWaterMdot = interp1(d.calcs.Time, ...
    d.calcs.CalorimetryWaterMdot, d.TempLS.Time,'pchip');

% Cal Spool Sizes

% x.constants.SACalSpoolGroove = 2.3053*(0.0254^2);                          % SA =
2.3053 in^2, Surface Area for 1 individual water groove on the calorimetry spools
% x.constants.SACalSpoolGroove = (13.70/4)*(0.0254^2);                       % INCORRECT
AREA, used for Reprocess 27Mar19, Internal SA of Cal Spool/4 for four water cooling
chanels
x.constants.SACalSpoolGroove = (3.534)*(0.0254^2);                         % Internal SA
of Cal Spool for four water cooling chanels
x.constants.SACalNozzle = 12.2093*(0.0254^2);                              % SA =
12.2093 in^2, Surface Area for Nozzle Water Jacket including flanges

%----- Cal Spool HT

x.LSHT.Calorimetry.Time = d.TempLS.Time;
x.LSHT.Calorimetry.QT4 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T4-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QT5 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T5-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QT6 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T6-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QT7 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T7-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QT8 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T8-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QT9 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T9-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QT10 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T10-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QT11 = (5/9)*x.calcs.CalorimetryWaterMdot ...
    .*x.constants.cpWater.* (d.TempLS.T11-d.TempLS.CalorimetryInlet) ...
    ./x.constants.SACalSpoolGroove;
x.LSHT.Calorimetry.QNozzle = (5/9)*x.calcs.NozzleWaterMdot.*x.constants.cpWater ...
    .* (d.TempLS.Nozzle-d.TempLS.WaterManifold)./x.constants.SANozzle;
```

%% Plotting

%% SECTION08 Figure 1, will automatically be full screen, outputs critical run information such as venturi/chamber Pressure, mass flows, phi, etc.

```
h.fig1=figure('units','normalized','outerposition',[0 0 1 1],...
    'PaperUnits','normalized','PaperPosition',[0 0 1 1]);
h.fig1.NextPlot = 'new';

% Mass flows
subplot(3,2,1,'Parent',h.fig1);
hold on
plot(d.Pressure.Time,d.calcs.O2Mdot,'b','DisplayName','Oxygen Mass Flow');
plot(d.Pressure.Time,d.calcs.FuelMdot,'r','DisplayName','RP-2 Injector Mass Flow');
plot(d.Pressure.Time,d.calcs.FuelFilmCoolingMdot,'k','DisplayName','RP-2 Film Cooling
Mass Flow');
grid on
legend show
title('Film Cooled Rocket Mass Flow');
xlabel('Time (s)');
ylabel('Mass Flow (kg/s)');

% OF ratios
figure(h.fig1);
subplot(3,2,2,'Parent',h.fig1)
hold on
plot(d.Pressure.Time,d.calcs.OFOverall,'b','DisplayName','OF Overall');
plot(d.Pressure.Time,d.calcs.OFCore,'r','DisplayName','OF Core');
grid on
legend show
title('Film Cooled Rocket Mixture Ratio');
xlabel('Time (s)');
ylabel('Mixture Ratio (O/F)');

% N2 Fuel Tank pressure
figure(h.fig1)
subplot(3,2,3,'Parent',h.fig1)
plot(d.Pressure.Time,d.Pressure.N2FuelTank);
grid on
title('Fuel Tank N2 Pressure');
xlabel('Time (s)');
ylabel('Pressure (psia)');

% Fuel venturi & manifold pressures
figure(h.fig1)
subplot(3,2,4,'Parent',h.fig1)
hold on
plot(d.Pressure.Time,d.Pressure.FuelVenturi,'b','DisplayName','Fuel Venturi Pressure');
plot(d.Pressure.Time,d.Pressure.FuelManifold,'r','DisplayName','Fuel Manifold Pressure');
grid on
legend show
ylim([0 round(max(d.Pressure.FuelVenturi)*1.2,-2)]);
title('Fuel Venturi & Mani Pressure');
xlabel('Time (s)');
ylabel('Pressure (psia)');
```

```
% O2 Venturi & Manifold pressures
figure(h.fig1)
subplot(3,2,5,'Parent',h.fig1)
hold on
plot(d.Pressure.Time,d.Pressure.O2EngineChoke,'b','DisplayName','Engine O2 Venturi
Pressure');
plot(d.Pressure.Time,d.Pressure.O2Manifold,'r','DisplayName','Engine O2 Manifold
Pressure');
grid on
legend show
ylim([0 round(max(d.Pressure.O2EngineChoke)*1.2,-2)]);
title('O2 Venturi & Mani Pressure');
xlabel('Time (s)');
ylabel('Pressure (psia)');

%Engine Chamber pressure
figure(h.fig1)
subplot(3,2,6,'Parent',h.fig1)
hold on
plot(d.Pressure.Time,d.Pressure.EngineChamber);
grid on
title('Engine Chamber Pressure');
xlabel('Time (s)');
ylabel('Pressure (psia)');
drawnow

% Figure 5, shows all the temperature information versus the engine chamber
% pressure.
chamberPressureAxisMax = ceil(max(d.Pressure.EngineChamber)/100)*100;
h.fig5 = figure('units','normalized','outerposition',[0.0 0.0 1.0 1.0],...
    'PaperUnits','normalized','PaperPosition',[0 0 1 1]);
% h.fig5.NextPlot = 'new';
% Spool temperatures
figure(h.fig5);
subplot(2,2,1)
[h.Ax,h.Line1,h.Line2] = plotyy(...
    [d.TempLS.Time,d.TempLS.Time,d.TempLS.Time,...
    d.TempLS.Time,d.TempLS.Time,d.TempLS.Time],...
    [d.TempLS.Spool1,d.TempLS.Spool2,d.TempLS.Spool3,...
    d.TempLS.Spool4,d.TempLS.Nozzle,d.TempLS.WaterManifold],...
    d.Pressure.Time,d.Pressure.EngineChamber);
title('Spool Temperatures with Engine Chamber Pressure');
legend('Spool 1','Spool 2','Spool 3','Spool 4','Nozzle','Inlet');
xlabel('Time (s)');
ylabel(h.Ax(1),'Spool Temperatures (deg-F)');                              % Left Axis
Label
ylabel(h.Ax(2),'Engine Chamber Pressure (psia)');                          % Right Axis
Label
ylim(h.Ax(1),[0 200]);
ylim(h.Ax(2),[0 chamberPressureAxisMax]);
h.Line2.Color=[0.5,0.5,0.5];
set(h.Ax(1),'ycolor','k','YTick',0:50:200);
```

```
set(h.Ax(2),'ycolor',[0.5,0.5,0.5],'YTick',0:100:chamberPressureAxisMax);
grid on

figure(h.fig5);
subplot(2,2,2)
[h.Ax,h.Line1,h.Line2] = plotyy(...
    repmat(d.TempLS.Time,1,8),...
    [d.TempLS.T4,d.TempLS.T5,d.TempLS.T6,...
    d.TempLS.T7,d.TempLS.T8,d.TempLS.T9,d.TempLS.T10,d.TempLS.T11],...
    d.Pressure.Time,d.Pressure.EngineChamber);
title('Calorimetry Temperatures');
legend('T4','T5','T6','T7','T8','T9','T10','T11');
xlabel('Time (s)');
ylabel(h.Ax(1),'Calorimetry Temperatures (deg-F)');                          % Left Axis
Label
ylabel(h.Ax(2),'Engine Chamber Pressure (psia)');                            % Right Axis
Label
ylim(h.Ax(1),[0 200]);
ylim(h.Ax(2),[0 chamberPressureAxisMax]);
h.Line2.Color=[0.5,0.5,0.5];
set(h.Ax(1),'ycolor','k','YTick',0:50:200);
set(h.Ax(2),'ycolor',[0.5,0.5,0.5],'YTick',0:100:chamberPressureAxisMax);
grid on

% Film cooling temperature
figure(h.fig5)
subplot(2,2,3)
[h.Ax2,h.Line1,h.Line2] = plotyy(d.TempLS.Time,d.TempLS.FilmCooling,...
    d.Pressure.Time,d.Pressure.EngineChamber);
title('Film Cooling Temperature with Engine Chamber Pressure');
xlabel('Time (s)');
ylabel(h.Ax2(1),'Film Cooling Temperature (deg-F)');                                % Left
Axis Label
ylabel(h.Ax2(2),'Engine Chamber Pressure (psia)');                           % Right Axis
Label
ylim(h.Ax2(1),[0 chamberPressureAxisMax]);
ylim(h.Ax2(2),[0 chamberPressureAxisMax]);
set(h.Ax2(1),'ycolor','k','YTick',0:100:chamberPressureAxisMax);
set(h.Ax2(2),'ycolor',[0.5,0.5,0.5],'YTick',0:100:chamberPressureAxisMax);
h.Line2.Color=[0.5,0.5,0.5];
grid on

% High speed temperatures
figure(h.fig5)
subplot(2,2,4)
r.names = fieldnames(d.TempHS);
r.names=r.names(2:end);
[h.Ax3,h.Line1,h.Line2] = plotyy(repmat(d.TempHS.Time,1,16), ...
    0.95039.*(r.HighSpeedTemp.^3)-10.715.*(r.HighSpeedTemp.^2) ...
    +124.02.*r.HighSpeedTemp +0.83168,...
    d.Pressure.Time,d.Pressure.EngineChamber);
title('High Speed Thermocouple data');
```

```
xlabel('Time (s)');
ylabel(h.Ax3(1),'HS Temperatures (C)');                                    % Left Axis
Label
ylabel(h.Ax3(2),'Engine Chamber Pressure (psia)');                         % Right Axis
Label
ylim(h.Ax3(2),[0 chamberPressureAxisMax]);
if round(max(ylim(h.Ax3(1)))) > 0
    ylim(h.Ax3(1),[0 round(max(ylim(h.Ax3(1))),0)]);
    set(h.Ax3(1),'ycolor','k','YTick',0:max(ylim(h.Ax3(1)))/5:max(ylim(h.Ax3(1))));
end
set(h.Ax3(2),'ycolor',[0.5,0.5,0.5],'YTick',0:100:chamberPressureAxisMax);
h.Line2.Color=[0.5,0.5,0.5];
legend(r.names)
grid on
drawnow

figure(h.fig1)

%----- Heat Transfer Plots

%---- Test Spool

figure;
subplot(3,1,1)
plot(d.TempLS.Time, [d.TempLS.Spool1,d.TempLS.Spool2,d.TempLS.Spool3, ...
    d.TempLS.Spool4,d.TempLS.Nozzle,d.TempLS.WaterManifold]);
title('Test Spool Water Temperatures')
xlabel('Time (s)')
ylabel('Temperature (K)')
legend('Spool1', 'Spool2', 'Spool3', 'Spool4', 'Nozzle', ...
    'InletManifold', 'location', 'best')

subplot(3,1,2)
plot(x.LSHT.TestSpool.Time, [x.calcs.TestSpoolWaterMdot, ...
    x.calcs.NozzleWaterMdot])
title('Test Spool Cooling Water Mass Flow')
xlabel('Time (s)')
ylabel('Mass Flow (kg/s)')
axis([0 45 0 1])
legend('TestSpool', 'Nozzle', 'location', 'best')

subplot(3,1,3)
plot(x.LSHT.TestSpool.Time, [x.LSHT.TestSpool.QSpool1, ...
    x.LSHT.TestSpool.QSpool2, x.LSHT.TestSpool.QSpool3, ...
    x.LSHT.TestSpool.QSpool4, x.LSHT.TestSpool.QNozzle])
title('Test Spool Heat Flux')
xlabel('Time (s)')
ylabel('Q/A (W/m^2)')
axis([0 45 0 1E7])
legend('QSpool1', 'QSpool2', 'QSpool3', 'QSpool4', 'QNozzle', ...
    'location', 'best')
```

```
%---- Calorimetry

figure;
subplot(3,1,1)
plot(d.TempLS.Time, [d.TempLS.T4,d.TempLS.T5,d.TempLS.T6, ...
    d.TempLS.T7, d.TempLS.T8, d.TempLS.T9, d.TempLS.T10, d.TempLS.T11, ...
    d.TempLS.Nozzle,d.TempLS.WaterManifold, d.TempLS.CalorimetryInlet]);
title('Calorimetry Water Temperatures')
xlabel('Time (s)')
ylabel('Temperature (K)')
legend('T4', 'T5', 'T6', 'T7', 'T8', 'T9', 'T10', 'T11', 'Nozzle', ...
    'TestSpoolInletManifold', 'CalorimetryInletManifold', 'location', ...
    'best')
axis([0 40 0 100])

subplot(3,1,2)
plot(x.LSHT.Calorimetry.Time, [x.calcs.CalorimetryWaterMdot, ...
    x.calcs.NozzleWaterMdot])
title('Calorimetry Cooling Water Mass Flow')
xlabel('Time (s)')
ylabel('Mass Flow (kg/s)')
legend('CalorimetryWaterMdot', 'NozzleWaterMdot', 'location', 'best')
% axis([0 40 -2 2])

subplot(3,1,3)
plot(x.LSHT.Calorimetry.Time, [x.LSHT.Calorimetry.QT4, ...
    x.LSHT.Calorimetry.QT5, x.LSHT.Calorimetry.QT6, ...
    x.LSHT.Calorimetry.QT7,x.LSHT.Calorimetry.QT8, ...
    x.LSHT.Calorimetry.QT9, x.LSHT.Calorimetry.QT10, ...
    x.LSHT.Calorimetry.QT11, x.LSHT.Calorimetry.QNozzle])
title('Calorimetry Heat Flux')
xlabel('Time (s)')
ylabel('Q/A (W/m^2)')
legend('QT4', 'QT5', 'QT6', 'QT7', 'QT8', 'QT9', 'QT10', 'QT11', ...
    'QNozzle', 'location', 'best')
% axis([0 40 -1E7 1E7])
```

%% User-defined inputs for run times to average over.

%% SECTION09

```
if strcmp('Yes',in.data_summary)
    in.prompt = {sprintf(...
        'Start time for calculating averages, from %0.2f to %0.2f (s):',...
        min(d.Pressure.Time),max(d.Pressure.Time)),...
        sprintf('End time for calculating averages, from %0.2f to %0.2f (s):',...
        min(d.Pressure.Time),max(d.Pressure.Time)),...
        'Plot the O2/N2 Feedback pressure (T/F)?',...
        'Plot the O2/H2 Toch pressure (T/F)?',...
        'Plot the Filmcooling/H20 Cooling pressure (T/F)?'};
    in.dlg_title = 'User-inputs';
```

```
    in.num_lines = 1;
    in.defaultans = {num2str(d.Averages.StartTime),...
        num2str(d.Averages.EndTime),'F','F','F'};
    in.answer = inputdlg(in.prompt,in.dlg_title,in.num_lines,in.defaultans);
    drawnow;pause(0.05);

    d.Averages.StartTime = str2num(in.answer{1});
    d.Averages.EndTime = str2num(in.answer{2});
    r.plotter1 = in.answer{3};
    r.plotter2 = in.answer{4};
    r.plotter3 = in.answer{5};
end
```

%% Calculate flow averages, add to previously-created plots.

%% SECTION10

```
if strcmp('Yes',in.data_summary)
    d.Averages.O2Mdot = nanmean(...
        d.calcs.O2Mdot((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));            % kg/s

    d.Averages.FuelMdot = nanmean(...
        d.calcs.FuelMdot((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));            % kg/s

    d.Averages.FuelFilmCoolingMdot = nanmean(...
        d.calcs.FuelFilmCoolingMdot((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));            % kg/s

    d.Averages.OFCore = nanmean(...
        d.calcs.OFCore((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));            % kg/s

    d.Averages.OFOverall = nanmean(...
        d.calcs.OFOverall((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));            % kg/s

    % Calculate C-star, core average
    d.Averages.CStarCore = nanmean(...                                         % m/s
        d.calcs.CStarCore((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));

    % Calculate C-star overall average
    d.Averages.CStarOverall = nanmean(...                                      % m/s
        d.calcs.CStarOverall((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));

    d.Averages.EtaCore = nanmean(...                                           % Average
efficiency in the core flow
        d.calcs.EtaCore((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));
```

```
    d.Averages.EtaOverall = nanmean(...
        d.calcs.EtaOverall((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));

    d.Averages.EngineChamberPsia = nanmean(...
        d.Pressure.EngineChamber((...
        d.Averages.StartTime*1000+1):(d.Averages.EndTime*1000+1)));

    d.Averages.QSpool4 = (10^-6)*nanmean(...
        x.LSHT.TestSpool.QSpool4((...
        d.Averages.StartTime*50+1):(d.Averages.EndTime*50+1)));

    d.Averages.QCalSpool = (10^-6)*(1/4)*( nanmean(...
        x.LSHT.Calorimetry.QT8((...
        d.Averages.StartTime*50+1):(d.Averages.EndTime*50+1))) ...
        + nanmean(...
        x.LSHT.Calorimetry.QT9((...
        d.Averages.StartTime*50+1):(d.Averages.EndTime*50+1))) ...
        + nanmean(...
        x.LSHT.Calorimetry.QT10((...
        d.Averages.StartTime*50+1):(d.Averages.EndTime*50+1))) ...
        + nanmean(...
        x.LSHT.Calorimetry.QT11((...
        d.Averages.StartTime*50+1):(d.Averages.EndTime*50+1))));            % Average of
all cooling channels on downstream spool

     d.Averages.QNozzle = (10^-6)*nanmean(...
        x.LSHT.TestSpool.QNozzle((...
        d.Averages.StartTime*50+1):(d.Averages.EndTime*50+1)));

    disp(['Average O2 Mass Flow (kg/s): ' num2str(d.Averages.O2Mdot)])
    disp(['Average Fuel Mass Flow (kg/s): ' num2str(d.Averages.FuelMdot)])
    disp(['Average Fuel Film Cooling Mass Flow (kg/s): '
num2str(d.Averages.FuelFilmCoolingMdot)])
    disp(['Average OF Core: ' num2str(d.Averages.OFCore )])
    disp(['Average OF Overall: ' num2str(d.Averages.OFOverall)])
    disp(['Average C* Core: ' num2str(d.Averages.CStarCore)])
    disp(['Average C* Overall: ' num2str(d.Averages.CStarOverall)])
    disp(['Average Eta Core: ' num2str(d.Averages.EtaCore)])
    disp(['Average Eta Overall: ' num2str(d.Averages.EtaOverall)])
    disp(['Average Chamber Pressure (psia): ' num2str(d.Averages.EngineChamberPsia)])
    disp(['Average Test Spool Heat Flux (MW/m^2): ' num2str(d.Averages.QSpool4)])
    disp(['Average Cal Spool Heat Flux (MW/m^2): ' num2str(d.Averages.QCalSpool)])
    disp(['Average Nozzle Heat Flux (MW/m^2): ' num2str(d.Averages.QNozzle)])
end
```

%% Save Variable space

```
%SECTION11
if strcmp('Yes',in.data_summary)
```

```
    disp('Saving variable space ...')
    warning off
    tic
    save([strrep(strrep(strcat(in.loc,in.Pressure.name),...
        'RocketPressureData',''),'.txt',''),'.mat']);                          % Saves
all of the variables for further post processing
    toc
end
```

%% Add average section lines to the previous plots

%% SECTION12

```
if strcmp('Yes',in.data_summary)
    disp('Adding average section lines and plotting other figures if specified ...')
    tic
    figure(h.fig1)
    subplot(3,2,4,'Parent',h.fig1)
    r.ylim1=ylim;
    plot([d.Averages.StartTime,d.Averages.StartTime],[r.ylim1(1),r.ylim1(2)],'-k');
    plot([d.Averages.EndTime,d.Averages.EndTime],[r.ylim1(1),r.ylim1(2)],'-k');

    figure(h.fig1)
    subplot(3,2,5,'Parent',h.fig1)
    r.ylim1=ylim;
    plot([d.Averages.StartTime,d.Averages.StartTime],[r.ylim1(1),r.ylim1(2)],'-k');
    plot([d.Averages.EndTime,d.Averages.EndTime],[r.ylim1(1),r.ylim1(2)],'-k');

    subplot(3,2,6,'Parent',h.fig1)
    r.ylim1=ylim;
    plot([d.Averages.StartTime,d.Averages.StartTime],[r.ylim1(1),r.ylim1(2)],'-k');
    plot([d.Averages.EndTime,d.Averages.EndTime],[r.ylim1(1),r.ylim1(2)],'-k');
    drawnow

    % Plot, or do not plot, Figures 2, 3, and 4. Based on user-input
    if r.plotter1 == 'T'
        h.fig2 = figure;
        subplot(2,1,1)
        plot(d.Pressure.Time,d.Pressure.O2Feedback);
        grid on
        title('O_2 Feedback Pressure');
        xlabel('Time (s)');
        ylabel('Pressure (psia)');

        figure(h.fig2)
        subplot(2,1,2)
        plot(d.Pressure.Time,d.Pressure.N2Feedback);
        grid on
        title('N_2 Feedback Pressure');
        xlabel('Time (s)');
        ylabel('Pressure (psia)');
    end
```

```
    if r.plotter2 == 'T'
        h.fig3 = figure;
        subplot(2,1,1)
        plot(d.Pressure.Time,d.Pressure.O2Torch);
        grid on
        title('Torch O2 Pressure');
        xlabel('Time (s)');
        ylabel('Pressure (psia)');

        figure(h.fig3)
        subplot(2,1,2)
        plot(d.Pressure.Time,d.Pressure.H2Torch);
        grid on
        title('Torch H2 Pressure');
        xlabel('Time (s)');
        ylabel('Pressure (psia)');
    end
    if r.plotter3 == 'T'
        h.fig4 = figure;
        subplot(2,1,1)
        plot(d.Pressure.Time,d.Pressure.FilmCooling);
        grid on
        title('Film Cooling Pressure');
        xlabel('Time (s)');
        ylabel('Pressure (psia)');

        figure(h.fig4)
        subplot(2,1,2)
        plot(d.Pressure.Time,d.Pressure.WaterCoolingInletManifoldpsig);
        grid on
        title('Water Cooling Pressure');
        xlabel('Time (s)');
        ylabel('Pressure (psia)');
    end
    toc
end
```

%% Saving more stuff

%% SECTION13 Save figures as JPEGs

```
if strcmp('Yes',in.data_summary)
    disp('Saving Figure 1 ...')
    tic
    screen2png(h.fig1,strcat(in.loc,strrep(strrep(...
        in.Pressure.name,'RocketPressureData',''),...
        '.txt','_Pressures_and_FlowRates.png')));
    toc
    disp('Saving Figure 2 ...')
    tic
    screen2png(h.fig5,strcat(in.loc,strrep(strrep(...
        in.Pressure.name,'RocketPressureData',''),...
```

```
        '.txt','_Temperatures.png')));
    toc

    % Write data to Excel file
    % dlg = msgbox('Saving pressure data ...');
    disp('Writing pressure data to spreadsheet ...')
    tic
    writetable(struct2table(d.Pressure),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',1);
    toc

    disp('Writing low speed temperature data to spreadsheet ...')
    tic
    writetable(struct2table(d.TempLS),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',2);
    toc

    disp('Writing high speed temperature data to spreadsheet ...')
    tic
    writetable(struct2table(d.TempHS),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',3);
    toc

    disp('Writing calculation data to spreadsheet ...')
    tic
    writetable(struct2table(d.calcs),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',4);
    toc

    disp('Writing averages to spreadsheet ...')
    tic
    writetable(struct2table(d.Averages),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',5);
    toc

    disp('Writing HS Heat Flux to spreadsheet ...')
    tic
    writetable(struct2table(x.HSHT),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',6);
    toc

    disp('Writing LS Test Spool Heat Flux to spreadsheet ...')
    tic
    writetable(struct2table(x.LSHT.TestSpool),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',7);
```

```
    toc

    disp('Writing LS Calorimetry Heat Flux to spreadsheet ...')
    tic
    writetable(struct2table(x.LSHT.Calorimetry),strrep(strrep(strcat(...
        in.loc,in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'),'FileType','spreadsheet','Sheet',8);
    toc

    % Modify Excel workbook to update tab names.
    disp('Changing spreadsheet tab names ...')
    tic
    h.e = actxserver('Excel.Application');
    h.ewb = h.e.Workbooks.Open(strrep(strrep(strcat(...
        in.loc,'\',in.Pressure.name),'RocketPressureData',''),...
        '.txt','.xlsx'));
    h.ewb.Worksheets.Item(1).Name = 'Pressure';
    h.ewb.Worksheets.Item(2).Name = 'LS Temperature';
    h.ewb.Worksheets.Item(3).Name = 'HS Temperature';
    h.ewb.Worksheets.Item(4).Name = 'Mass Flow Calculations';
    h.ewb.Worksheets.Item(5).Name = 'Average Calculations';
    h.ewb.Worksheets.Item(6).Name = 'HS Heat Flux Calculations';
    h.ewb.Worksheets.Item(7).Name = 'LS Test Spool Heat Flux Calcs';
    h.ewb.Worksheets.Item(8).Name = 'LS Calorimetry Heat Flux Calcs';
    h.ewb.Save
    h.ewb.Close(false)
    h.e.Quit
    warning on
    toc
end
```

Published with MATLAB® R2018a

APPENDIX B. DATA COLLECTION DEVICES

Figure 12, Figure 14, and Figure 15 show ports used to measure pressure and temperature at key points on the assembly. Pressure transducers (Table 5) were operated at 200 Hz and sampled at 1000 Hz. Low speed 3.175 mm (0.125 in) thermocouples (Table 6) were sampled at 50 Hz. "Using the surface area of the spools, the cooling water flow rates (calculated using pressure data), and temperature data from the low speed thermocouples, [Equation 11] can be used to calculate the heat flux into the cooling water and therefore through chamber walls" [1]. The data measurement ports incorporated into the chamber-nozzle were designed to fit these devices (See Chapter IV).

Table 5. Pressure Transducers. Source: [1].

Data Type	Name	Sample Rate (Hz)
Pressure	O2Feedback	1000
Pressure	N2Feedback	1000
Pressure	O2EngineChoke	1000
Pressure	O2Torch	1000
Pressure	N2FuelTank	1000
Pressure	H2Torch	1000
Pressure	FuelVenturi	1000
Pressure	O2Manifold	1000
Pressure	EngineChamber	1000
Pressure	FilmCooling	1000

Table 6. Thermocouples. Adapted from [1].

Data Type	Name	Sample Rate (Hz)
Temperature	Spool1	50
Temperature	Spool2	50
Temperature	Spool3	50
Temperature	Spool4	50
Temperature	Nozzle	50
Temperature	WaterManifold	50
Temperature	FilmCooling	50

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. TEST LOG

Run #	Run Name	Spool 4	Nozzle Diam (in)	Run Time (sec)	Chamber Diam (cm)	O/F Core (Target)	O/F Core (Actual)	Chamber Pres (MPa)	% Film	ηC* (overall)	Water Inlet Temp (C)	Spool 4 Outlet Temp (C)	Spool 4 ΔT (C)	Spool 4 Qdot (MW/m^2)
1	**Oct25 Run03**	Cu-101	0.625	10	3.81	1.80	1.789	4.034	19.015	0.958	24.877	38.069	13.192	2.263
2	**Oct25 Run04**	Cu-101	0.625	10	3.81	1.80	1.841	3.885	19.381	0.957	26.245	40.074	13.830	2.444
3	**Oct25 Run05**	Cu-101	0.625	10	3.81	1.80	1.913	4.062	19.342	0.957	27.601	42.993	15.392	2.662
4	**Nov01 Run02**	Cu-101	0.625	12	3.81	2.20	2.075	4.153	19.692	0.979	16.814	30.773	13.958	2.538
5	**Nov01 Run03**	Cu-101	0.625	12	3.81	1.90	1.912	4.247	19.166	0.987	17.865	31.496	13.632	2.143
6	**Nov01 Run04**	Cu-101	0.625	12	3.81	1.80	1.768	4.251	18.798	0.997	19.190	33.681	14.491	2.414
7	**Nov12 Run01**	Cu-101	0.625	12	3.81	2.00	1.963	4.172	13.369	0.984	15.675	31.554	15.879	3.023
8	**Nov12 Run02**	Cu-101	0.625	12	3.81	1.80	1.782	4.236	13.284	0.992	17.405	33.197	15.792	2.730
9	**Nov20 Run03**	Cu-101	0.500	30	3.81	1.80	1.925	6.871	13.337	1.051	15.697	37.441	21.743	4.293
10	**Nov21 Run01**	ZrO2 coated SS304	0.500	21	3.81	2.18	2.287	6.963	9.651	1.113	15.822	37.769	21.947	4.728

Run #	Run Name	Spool 4	Nozzle Diam (in)	Run Time (sec)	Chamber Diam (cm)	O/F Core (Target)	O/F Core (Actual)	Chamber Pres (MPa)	% Film	ηC* (overall)	Water Inlet Temp (C)	Spool 4 Outlet Temp (C)	Spool 4 ΔT (C)	Spool 4 Qdot (MW/m^2)
11	**Dec04 Run01**	Cu-101/ Ni plated	0.625	15	3.81	2.25	2.234	4.114	13.469	0.996	13.503	34.980	21.477	3.822
12	**Dec10 Run02**	Cu-101/ Ni plated	0.625	15	3.81	1.80	2.168	4.153	13.418	0.998	14.872	40.151	25.279	3.534
13	**Dec10 Run03**	Cu-101/ Ni plated	0.625	15	3.81	2.00	2.271	4.160	13.468	1.010	15.836	37.609	21.773	4.106
14	**Apr08 Run03**	SS 17–4PH 0.100”	0.625	12	3.81	1.80	2.000	4.280	13.332	1.013	14.839	30.671	15.832	3.249
15	**Apr09 Run03**	SS 17–4PH 0.070”	0.625	12	3.81	1.80	2.006	4.291	13.338	1.014	15.877	30.648	14.772	3.237

LIST OF REFERENCES

[1] Z. Lewis, "Heat transfer analysis and regernative limits for a film cooled uni-element liquid rocket engine," M.S. Thesis, Dept. of Mechanical and Aerospace Engineering, NPS, Monterey, CA, USA, 2019.

[2] G. P. Sutton, *History of Liquid Propellant Rocket Engines*, Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2006.

[3] G. P. Sutton and O. Biblarz, *Rocket Propulsion Elements*, 9th ed., Hoboken, NJ: John Wiley & Sons, 2017.

[4] R. Arnold, D. Suslov and O. J. Haidn, "Influence parameters on film cooling effectiveness in a high pressure subscale combustion chamber," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, no. [Online] Available: https://arc.aiaa.org/doi/pdf/10.2514/6.2009-453, Jan. 2009.

[5] P. R. Grandl, "Rapid fabrication techniques for liquid rocket channel wall nozzles," *AIAA Propulsion and Power 2016 Conference*, Jul. 2016.

[6] A. Kwas, "Using additive manufacturing to print a CubeSat propulsion system," in *AIAA/SAE/ASEE Joint Propulsion Conference*, Orlando, 2015.

[7] Desktop Metal, "Deep Dive: Bound Metal Deposition (BMD)," Accessed March 1, 2202. [Online]. Available: https://www.desktopmetal.com/resources/deep-dive-bmd

[8] A. Watson, J. Belding and B. D. Ellis, "Characterization of 17–4 PH processed via bound metal deposition," in *TMS 2020 149th Annual Meeting & Exhibition*, San Diego, 2020.

[9] C. Kirchberger, G. Schlieben, A. Hupfer and H. P. Kau, "Experimental investigation on film cooling in a hydrocarbon/GOX rocket combustion chamber," 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Jan. 2012. [Online]. Available: https://arc.aiaa.org/doi/pdf/10.2514/6.2012-526

[10] C. Kirchberger, "Improved prediction of heat Transfer in rocket combustor for GOX/Kerosene," in *47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition (Disc 1)*, 2009.

[11] I. Martinez, “Fuel Properties,” Termodinámica básica y aplicada, 2020. [Online]. Available: http://webserver.dmt.upm.es/~isidoro/bk3/c15/Fuel%20properties.pdf

[12] MIL-DTL-25576E_NOTICE-1, Military Specification, Jan. 2011. [Online]. Available: http://everyspec.com/MIL-SPECS/MIL-SPECS-MIL-DTL/MIL-DTL-25576E_NOTICE-1_30203/

[13] R. Arnold, D. I. Suslov and O. J. Haidn, “Film cooling in a high-pressure subscale combustion chamber,” *Journal of Propulsion and Power,* vol. 26, no. 3, 2010.

[14] R. Arnold, D. I. Suslov and O. J. Haidn, “Film cooling of accelerated flow in a subscale combustion chamber,” *Journal of Propulsion and Power,* vol. 25, no. 2, 2009.

[15] J. W. Magee, T. J. Bruno, D. G. Friend, M. L. Huber, A. Laesecke, E. W. Lemmon, M. O. McLinden, R. A. Perkins, J. Baranski and J. A. Widegren, “Thermophysical properties measurements and models for rocket propellant RP-1: phase I,” NISTIR 6646, Feb. 2007.

[16] “RP2-1 POSF7688,” Combustion reaction models of liquid fuels Accessed April 10, 2020. [Online]. Available: https://web.stanford.edu/group/haiwanglab/HyChem/fuels/R1_spec.html

[17] P. C. Andersen and T. J. Bruno, “Thermal decomposition kinetics of RP-1 rocket propellant,” *Industrial & Engineering Chemistry Research*, vol. 44, no. 6, pp. 1670–1676, Dec 2005.

[18] C. Tang, “Carbon deposition properties in a film-cooled liquid rocket engine,” M.S. Thesis, Dept. of Mech. And Aero. Eng., NPS, Monterey, CA, USA, 2018.

[19] R. Wright, “Liquid Rocked Engine Redesign 2017 Mk2,” Solidworks, unpublished.

[20] KVA Stainless, “17-4 Stainless Steel,” KVAstainless.com. Accessed April 29, 2020. [Online]. Available: http://www.kvastainless.com/17-4.html

[21] Aalco - Ferrous, “Copper Alloys - Copper C101 Properties, Fabrication and Applications.” Accessed March 31, 2020. [Online]. Available: https://www.azom.com/article.aspx?ArticleID=2850

[22] Aalco - Ferrous, “Stainless Steels - Stainless 304 Properties, Fabrication and Applications.” Accessed April 25, 2020. Available: https://www.azom.com/article.aspx?ArticleID=2867

[23] AmericanElements, “Zirconium Oxide,” AmericanElements. Accessed May 1, 2020. [Online]. Available: https://www.americanelements.com/zirconium-oxide-1314-23-4

[24] Desktop Metal, *Studio System+ Bound Metal Deposition Design Guidelines*, Burlington, MA, USA, 2019.

[25] F. M. White, *Fluid Mechanics*, 7th ed., New York: McGraw-Hill, 2011.

[26] Dassault Systems, *Technical Reference: SolidWorks Flow Simulation 2018*, Waltham, MA, USA, 2018.

[27] Dassault Systems, *Validation Examples: SolidWorks Flow Simulation 2018*, Waltham, MA, USA, 2018.

[28] Dassault Systems, *Tutorials: SolidWorks Flow Simulation 2018*, Waltham, MA, USA, 2018.

[29] Desktop Metal, “DM Studio System Product Overview.”Accessed March 10, 2020. [Online]. Available: https://www.desktopmetal.com/products/studio

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California