Jonas Sømod Ahmed

An Investigation into Hybrid Rocket Injectors

Master's thesis in Mechanical Engineering

Supervisor: James R. Dawson

June 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

PROPULSE NTNU

Jonas Sømod Ahmed

An Investigation into Hybrid Rocket Injectors

Master's thesis in Mechanical Engineering
Supervisor: James R. Dawson
June 2020

Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

Propulse NTNU is a newly started student rocket team who are now trying to build a hybrid rocket. As this is a new organization there is a necessity to gather knowledge about how rockets and their key components function. One of these key components is the injector, whose role is to deliver a certain mass flow of liquid oxidizer as a finely atomized spray to the combustion chamber in the rocket. This thesis has been written to provide a basis for designing injectors that Propulse can rely on in the years to come.

The most important aspects of the injector are the mass flow rate of oxidizer that it delivers as well as other flow characteristics that can improve the performance of the rocket. This has first been assessed through a literature study covering the background theory that is needed to understand how the injector functions in the hybrid rocket engine. Mass flow rate models are then developed to shed light on how the flow rate behaves through the injector. Single-phase models are presented first as a baseline. However, as nitrous oxide has been chosen as the oxidizer, two-phase models are needed as its high vapor pressure will cause vapor to form in the injector. Finally, calculations from these models are used to design a few preliminary injectors with CAD software.

It was found that the flow through the injector may choke for realistic operating conditions, which would decouple the mass flow rate from combustion chamber pressure oscillations and reduce instabilities. Thus, if the tank pressure can be kept constant and the flow is choked, the mass flow rate can be kept constant as well. However, due to the tank dynamics of self-pressurized N_2O, the tank pressure and subsequently the oxidizer flow rate are likely to fall for the duration of the rocket operation. By utilizing mass flow rate models that take the tank dynamics into account, a methodology for designing injectors that should deliver a certain average flow rate has been developed. This resulted in a few preliminary injector designs that Propulse may use, and has shown how they might approach the initial design of injectors in the future. However, injectors usually require some iterative fine-tuning based on experiments to get the exact flow rate that is desired. An experimental campaign that would have been useful for validation was planned, but unfortunately the COVID-19 pandemic stopped it from materializing.

Acknowledgments

I would like to express my gratitude to my advisor, Dr. James Dawson, for his invaluable guidance during this challenging project.

Additionally, I thank the members of Propulse NTNU for providing this interesting task and being available for productive meetings and consultations.

Finally, I wish to let my parents know that their heartfelt encouragement and assistance throughout the year has been greatly appreciated.

Contents

List of Figures

Nomenclature

Symbols and Abbreviations

Δ	Difference
$\dot{m}$	Mass flow rate
$\dot{r}$	Regression rate
γ	Heat capacity ratio
ρ	Density
σ	Surface Tension
τ_b	Characteristic bubble growth time
τ_r	Fluid residence time
θ	Impingement half-angle
A	Area
C_d	Discharge coefficient
C_p	Heat Capacity at constant pressure
C_v	Heat Capacity at constant volume
CO_2	Carbon Dioxide
D	Diameter
E	Internal energy
e	Specific internal energy
F	Thrust force
G	Mass flux
g	Standard gravitational acceleration
H	Height
h	Specific enthalpy
$HDPE$	High-density Polyethylene

HRE Hybrid Rocket Engine

$HTPB$ Hydroxyl-terminated polybutadiene

I_{sp} Specific Impulse

L Length

m Mass

N_2O Nitrous Oxide

O/F Oxidizer-to-fuel

p Pressure

R Gas Constant

r Radius

s Specific entropy

T Temperature

t Time

u Velocity

V Volume

W_e Weber Number

x Two-phase mixture quality

Y Compressibility correction factor

k Non-equilibrium parameter

Subscripts

a Atmospheric

$crit$ Critical value

d Used to denote surface area of port (not cross-section)

e Location at nozzle exhaust

f Fuel

HEM Homogeneous Equilibrium Model

l Liquid

LRO Liquid run-out

ox Oxidizer

p	port
PG	Perfect Gas
sc	Supercharge
SPI	Single-Phase Incompressible
TEM	Transient Equilibrium Model
tot	Total
v	Vapor

Chapter 1

Introduction

The interest in space is growing rapidly at the Norwegian University of Science and Technology (NTNU), with many new organizations under the Space NTNU umbrella. One of the founding members of Space NTNU is Propulse NTNU, a student rocket team that aims to build a rocket each year and compete at the Spaceport America Cup. In 2020, Propulse is going to build a hybrid rocket. Hybrid rockets are a topic of continuous study with growing interest, due to their safety and flexibility. However, their viability has been questioned because they have traditionally not produced enough thrust for many space applications. Despite that, recent advances in hybrid rockets could potentially fix some of the issues.

The injector is a key component in a hybrid rocket engine that can be complicated to design. As Propulse is a very new organization, there is a need to build up an in-house knowledge base on rocket science to design the rockets properly. Therefore, Propulse requested the author to:

"Provide Propulse NTNU with a better understanding of how hybrid rocket engine injectors function, with the goal of developing a method for designing injectors."

This work sets out to address this problem and produces a document that can teach members of Propulse the most important aspects of hybrid rocket injectors.

Specifically, this research investigates how the mass flow rate of oxidizer behaves through the injector, and how the injector can be used to control the mass flow rate. Additionally, how different injector configurations can affect the performance of the hybrid rocket is assessed. These aspects are considered with Propulse NTNU's project in mind, where their use of nitrous oxide is a particularly important factor. This all culminates in the design of a few preliminary injectors.

The problem outlined above will be handled through a rigorous literature study, which will shed light on much of the underlying theory that affects hybrid rocket injectors. This will lead to the programming of mass flow rate models that can then be used to help in the computer-aided design of injectors. Showing the process of using the theory and models to design the injectors will provide Propulse NTNU with a template that they can follow for future injector designs.

1.1 The Hybrid Rocket Concept

Conventional bi-propellant rocket engines are usually either what is known as a liquid rocket or a solid rocket, indicating the phase that the propellants are stored in. A liquid rocket engine has both its fuel and oxidizer separately stored in the liquid phase, with for example the liquid hydrogen-oxygen combination being one of the most widely known combinations. Liquid rockets are very complex, requiring two liquid propellant storage and delivery systems. The usually high combustion chamber pressures and utilization of cryogenic propellants further adds to the complexity. Furthermore, they often require high-performance turbo-pumps to drive the high mass flow rates required, which are powered by a small amount of the propellants running through a separate burner and turbine. Liquid rockets have been the culprits of many of the most spectacular rocket failures, with faulty turbo-pumps often being to blame [1].

Solid rocket engines, on the other hand, are mechanically much simpler than their liquid counterparts. Here, both the fuel and oxidizer are pre-mixed together in the solid phase, eliminating any need for liquid storage, turbo-pumps, and cryogenic cooling. However, storing the oxidizer and fuel together results in an explosive mixture that requires stringent safety precautions during handling, launch, and manufacturing. Additionally, the manufacturing process of the fuel is complex and expensive [1].

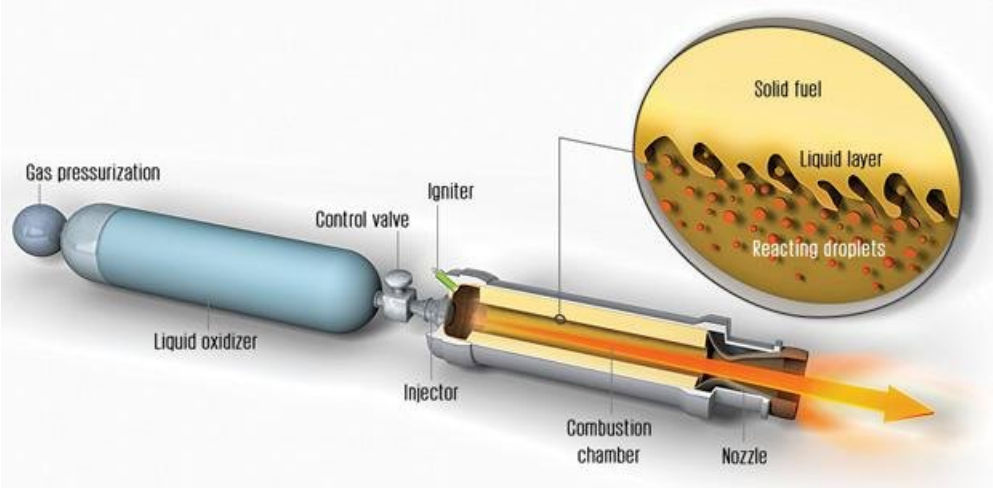


Figure 1.1: Schematic of a hybrid rocket engine [2]

Hybrid rocket engines are, as the name implies, rocket motors where either the oxidizer or the fuel is stored as a liquid, with the other as a solid. This is usually done with a liquid oxidizer and a solid fuel grain, although the reverse has also been done. Figure 1.1[2] shows a schematic of a typical hybrid rocket engine with a pressure-fed propellant feed system. This means that the liquid oxidizer in the tank is pressurized to a level such that when the control valve opens, the oxidizer flows to the combustion chamber. Pump-fed systems are also possible, but are more complex due to the need of turbo-machinery and are usually used for high-performance systems that require high chamber pressures [3]. Pressure-fed systems, on the other

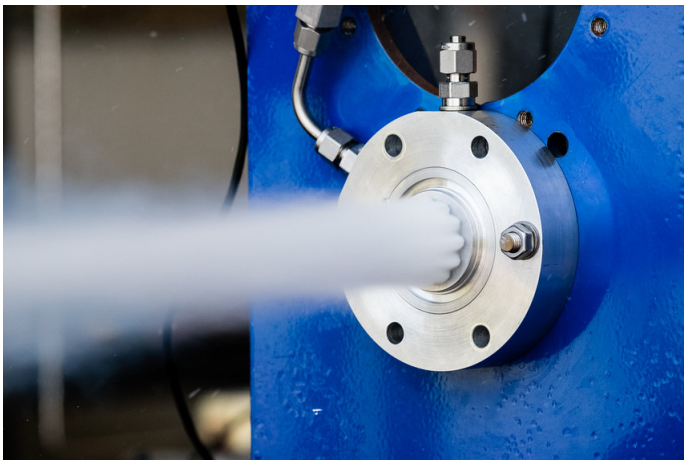


Figure 1.2: Picture of a basic shower-head injector [5]

hand, require heavy propellant tanks as the liquid oxidizer is stored at very high pressures, and offer less performance when compared to systems using turbo-pumps. However, they offer reduced complexity and cost [4]. The remainder of this thesis will focus on pressure-fed systems, as this is what will be most relevant for Propulse NTNU for the time being.

As the oxidizer flows from the tank to the combustion chamber, it goes through the injector. This component disperses the flow into tiny droplets that quickly evaporate to gaseous oxygen due to their high surface area to volume ratio. A picture of an injector can be seen in figure 1.2 [5]. For now, it can be thought of as a simple shower-head, i.e. a plate with multiple very small holes that the liquid runs through - although more complex designs do exist. The injector configuration can have a great effect on the performance of the rocket and will be discussed in chapter 2. The solid fuel grain is located in the combustion chamber and is usually in the shape of a cylinder with a hollowed-out circular section running through it, called the port. The solid fuel will evaporate due to heat in the chamber, and the igniter is needed to begin the combustion process. The vaporized oxidizer and fuel flow through the port, where they mix and burn. Pressure and thermal energy builds in the combustion chamber and is converted to kinetic energy as the gas expands through the nozzle, accelerating the flow and producing a thrusting force that propels the rocket forward as the gas is ejected at high velocities [6].

1.2 Advantages and Disadvantages

As explained above, a hybrid rocket utilizes a motor where the oxidizer is kept as a liquid, while the fuel is kept in the burn chamber as a solid. This configuration provides an inherent safety as there is no risk of explosion when the motor is not firing, which is the principal advantage of hybrid rockets when compared to both their liquid and solid counterparts. Solid rockets have the fuel and oxidizer mixed in one solid compound. Imperfections, cracks, or other disturbances in the fuel grain of a solid rocket can cause uncontrolled combustion and explosions. In a hybrid rocket engine, however, the solid fuel has a non-explosive character as the oxidizer is stored separately. This makes the fuel far easier to fabricate, store, and handle

which in turn reduces costs. Liquid bi-propellant rockets are complex and require flow systems of both liquid fuel and liquid oxidizer. If the liquid fuel and oxidizer mix in an uncontrolled manner due to a pump leak, for example, catastrophic explosions can happen [1].

Another advantage hybrid rockets have over solid rockets is that they generally have a better specific impulse I_{sp}. The specific impulse is a measure of how much total impulse the rocket produces per unit mass of propellant spent, shown in equation 1.1.

$$I_{sp} = \frac{I_{tot}}{m_{ox} + m_f} \tag{1.1}$$

This is somewhat analogous to liters of gasoline per kilometer driven for a car, as the total impulse is what determines the altitude that the rocket can reach. The specific impulse is one of the most important performance indicators for rockets. It is crucial for rocket designers to minimize the amount of propellant weighing down the rocket, in addition to the obvious cost benefits of needing to purchase less propellant. Although liquid motors currently tout the highest specific impulses, hybrid rocket engines may have the ability to get an I_{sp} advantage even over comparable liquid rockets. This is because it is much easier to add performance-enhancing materials such as aluminum powder to a solid fuel grain than a liquid fuel [1].

Hybrid engines also have the ability to smoothly change the thrust over a wide range through throttling, which means to regulate how much propellant is supplied to the engine. Throttling allows them to optimize the trajectory and terminate the thrust on demand. This is easier in a hybrid rocket as there is only one liquid delivery system to worry about. In a liquid rocket engine, the momenta of the liquid fuel and oxidizer streams must match during the mixing process, which is a difficult requirement that the hybrid rocket engine does not have to consider. Solid motors, on the other hand, usually do not have a way to throttle at all [1][6].

There are some disadvantages to hybrid rockets as well. During the burn, the mixture ratio of oxidizer to fuel (O/F ratio) will usually change. This is because the port diameter and thus the inner surface area of the solid fuel grain that is exposed to heat transfer expands during combustion [7]. Variations in the O/F ratio means that there is less control of combustion and will cause the I_{sp} to change as well [6]. This O/F shift will be discussed further in the next section of this chapter.

The primary disadvantage of hybrid rockets and the reason why hybrid rockets have struggled commercially is that the rate of evaporation of traditional solid fuels has historically been too low. This makes it difficult to achieve the high thrust that is needed for many applications. The rate of evaporation is often measured by the regression rate of the solid fuel and will be discussed further in the following sections. However, one advantage of hybrid rockets over solid rockets concerning the regression rate is that in a hybrid rocket the regression rate is usually insensitive to the chamber pressure. This allows the chamber pressure to be a free variable during motor design and allows it to be optimized for the specific mission [1].

1.3 Hybrid Engine Combustion

In hybrid rockets, hot gas is primarily generated from boundary layer combustion, shown in figure 1.3 [8]. Combustion occurs through diffusive mixing of fuel evaporating from the solid fuel grain and the flow of vaporized oxidizer through the port. A diffusion flame forms above the surface of the solid fuel upon ignition and heat transfer from the flame to the fuel grain sustains the combustion by evaporating more fuel [8].

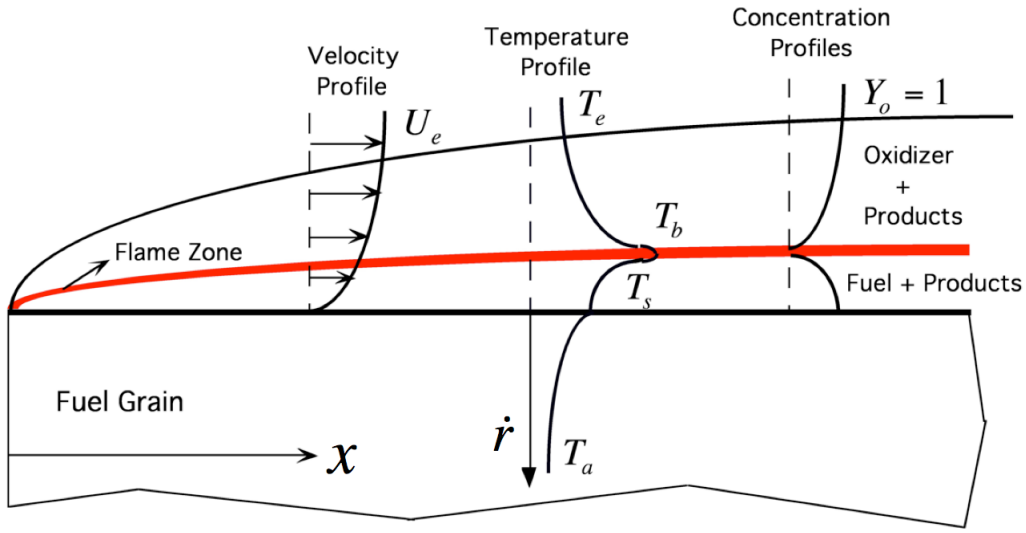


Figure 1.3: Boundary layer combustion [8]

As mentioned previously, the biggest issue for hybrid rockets is the low rate at which the fuel evaporates. This reduces the amount of thrust that the rocket can achieve. A commonly used equation for thrust calculations is given by equation 1.2,

$$F = \dot{m}u_e + (P_e - P_a)A_e \tag{1.2}$$

where $\dot{m}$ is the combined mass flow rate of fuel ($\dot{m}_f$) and oxidizer ($\dot{m}_{ox}$) leaving the nozzle. u_e is the exhaust velocity, and A_e is the nozzle exit area. P_e and P_a are the exhaust and atmospheric pressures, respectively. These are often assumed to be identical, canceling out the term. The mass flow of fuel is closely related to the regression rate by equation 1.3:

$$\dot{m}_f = \rho_f A_d \dot{r} \tag{1.3}$$

Here ρ_f is the density of the solid fuel, while A_d is the surface area of the port, meaning the inner area of the fuel grain that is exposed to heat transfer. $\dot{r}$ is the rate at which the port radius r increases, the regression rate. One should keep in mind that when using this relation, it is assumed that the radius is constant along the axis of the port. In reality, however, there tends to be an uneven burn along the length of the port.

From these equations, it is clear that the regression rate is an important factor for the thrust produced by a hybrid rocket. To understand how the performance of hybrid rockets can be improved, the regression rate must be studied.

1.3.1 Regression Rate

Regression rate studies are usually based on the work presented by Marxman et al.[9]. The significance of the theory is how they identify many of the factors that influence the regression rate and how they are related, showing how the regression rate is governed by turbulent heat and mass transfer in a reacting boundary layer. An important result of the theory is how the regression rate is dependent on the mass flux through the port. Because the mass flow rate increases with the axial distance along the port due to the accumulation of vaporized fuel, the local regression rate and local mass flux become coupled. This means that both variables are dependent on time and space, complicating the analysis [8][1].

A widely used version of the regression rate law that has shown to give accurate results ends up taking the form of equation 1.4,

$$\dot{r} = a\frac{G^n}{x^m} \tag{1.4}$$

with G being the total mass flux through the port:

$$G = \frac{\dot{m}_{ox} + \dot{m}_f}{\pi r^2} = \frac{\dot{m}_{port}}{\pi r^2} \tag{1.5}$$

Here $\dot{m}_{port}$ is the local mass flow, i.e. the mass flow of oxidizer and the accumulated fuel mass flow rate that has been transferred from the solid fuel grain upstream from a location x. The a parameter is an empirically determined constant and depends on the choice of fuel and oxidizer. Its units are

$$[a] = \frac{Length^{2n+m+1}}{Mass^n Time^{1-n}} \tag{1.6}$$

The classical values of n and m that Marxman's theory suggests are 0.8 and 0.2 respectively. However, measurements of n are usually in the 0.3 - 0.8 range and the value for m is usually much smaller than the theory predicts.[8]

While there are more rigorous treatments of the regression rate, including those by Cantwell, Zilliac, and Karabeyoglu [8][10][11], this thesis will consider a commonly used approximation,

$$\dot{r} = aG_{ox}^n \tag{1.7}$$

where a and n are experimentally determined. Note that the port length effect is neglected and the regression rate now can be expressed in terms of the mass flux of the oxidizer only, which is constant along the port. This method works better for higher O/F ratios, above 5 or so [1].

The O/F ratio is an important parameter in any rocket design due to its direct effect on multiple of the performance indicators for a rocket, such as the specific impulse or characteristic velocity. Designers usually want to aim for an optimum O/F ratio with regards to I_{sp}, and in figure 1.4 [8] it can be seen that there is an optimum mixture ratio for a given combination of fuel and oxidizer. Figure 1.4 also highlights the effect of additives. There is a tendency towards more fuel-rich mixtures and higher I_{sp} peaks as aluminum powder is added to the fuel. This can be

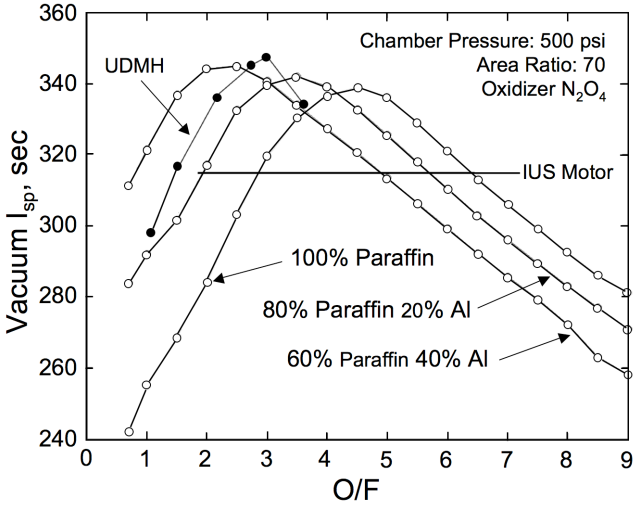


Figure 1.4: O/F ratio to I_{sp} for paraffin with different amounts of Al-additives [8]

beneficial not only to increase the specific impulse, but could also allow for a smaller oxidizer system [8].

Recalling equations 1.3, 1.5 and 1.7, the O/F ratio is given by

$$O/F = \frac{\dot{m}_{ox}}{\dot{m}_f} = \frac{\dot{m}_{ox}}{\rho_f A_d \dot{r}} = \frac{\dot{m}_{ox}}{\rho_f 2\pi r L_{port} a(\frac{\dot{m}_{ox}}{\pi r^2})^n} \tag{1.8}$$

From equation 1.8 the O/F shift of a burn with a fixed oxidizer mass flow that was alluded to earlier becomes clear. For $n > 0.5$ the O/F ratio increases with the radius of the port. This means that the decrease in mass flux dominates the increase of fuel surface area, resulting in a decrease in fuel mass flow. In turn, this will make the thrust decrease during of the burn. This can be a problem, but in some cases might be a desirable feature if the payload has a maximum acceleration constraint. This is in contrast to solid rockets, which tend to increase their thrust throughout the burn. To not lose too much I_{sp} for a configuration with $n > 0.5$ it is recommended to begin the burn at an O/F ratio slightly left of the peak, and terminating slightly to the right. Another interesting result of equation 1.8 is that for $n = 0.5$, the O/F ratio becomes independent of the port radius. For N_2O - paraffin rockets n turns out to be very close to 0.5 [8], which will be of further interest as this is the combination Propulse NTNU is planning to use. In table 1.1, values for a and n for a few common propellant combinations found by Waxman et. al are listed [12]. Note that these values may differ somewhat from system to system, as they are empirically determined. Additionally, certain injection schemes, especially swirl injectors, may result in significant deviations from this standard regression rate theory, including values of n greater than 1 [13].

Fuel	Oxidizer	a	n
HTPB	LOX	$3.043 * 10^{-2}$	0.681
HDPE	LOX	$2.340 * 10^{-2}$	0.62
Paraffin Wax	LOX	$11.70 * 10^{-2}$	0.62
Paraffin Wax	N_2O	$15.50 * 10^{-2}$	0.5

Table 1.1: Empirical Regression rate constants found by Waxman et al. [12]. Note that these are for use with equation 1.7, with G_{ox} in $\frac{kg}{m^2 s}$, but returns $\dot{r}$ in [mm/s].

There have been many suggestions for ways to achieve higher regression rates in hybrid rocket engines. One method is to increase the surface area exposed to heat transfer. To avoid very long port lengths, this is done by introducing multiple ports into the fuel grain as shown in figure 1.5 [14]. This also has the additional effect of increasing combustion efficiency, as a turbulent mixing environment is introduced in the post-combustion mixing chamber downstream of the fuel grain where the different streams from the multiple ports meet. This allows fuel and oxidizer that otherwise would be unburnt to mix and combust, boosting the combustion efficiency [6].

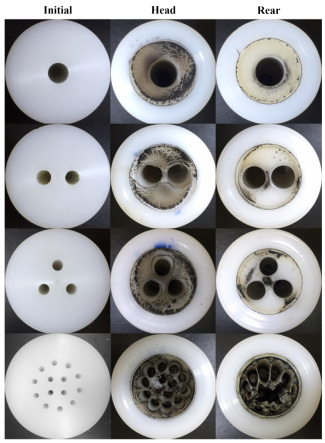


Figure 1.5: Different HDPE port designs [14]

However, there are some issues with the multi-port design. The complexity of multiple ports can be difficult to design and fabricate, and the structural integrity of the grain can become a problem, particularly towards the end of the burn. Each individual port may behave differently, for example in terms of uneven oxidizer flow. This may require dedicated injectors or large pre-combustion chambers, which sacrifices weight and simplicity [15].

Another method one might consider is to simply increase the oxidizer mass flow to achieve higher regression rates, or to a point where the fuel mass flow is a small part of the total mass flow rate that determines the thrust. However, this may come at the cost of performance. As was shown in figure 1.4, the I_{sp} (and other performance indicators) might suffer if the O/F ratio is not at a certain level. On the other hand, some propellants may have their optimum at higher O/F ratios, so this might be sufficient in some cases [15].

The choice of fuel material is an area that shows a lot of promise. The first and most obvious avenue would be to look at fuels that have a low heat of gasification, essentially allowing the same amount of heat from the flame to vaporize more fuel. In practice, however, the variation of the heat of gasification for feasible fuel types is limited. Moreover, there is what is known as a blocking effect; increasing heat transfer to the fuel does increase the rate of evaporation, but the temperature gradient on the fuel surface is reduced due to the increase in mass flux, thus limiting the heat transfer rate [16].

However, Karabeyoglu, Altman, and Cantwell [17] [16] found that paraffin-based fuels could increase the regression rate by a factor of 3-4 compared to traditional polymeric fuels. This is due to the entrainment of droplets from a thin liquid layer that forms on top of the fuel grain, driven by the oxidizer gas flow. It acts as a continuous spray injection along the port, with the droplets convecting between the flame and the melt layer and subsequently vaporizing. While other materials such

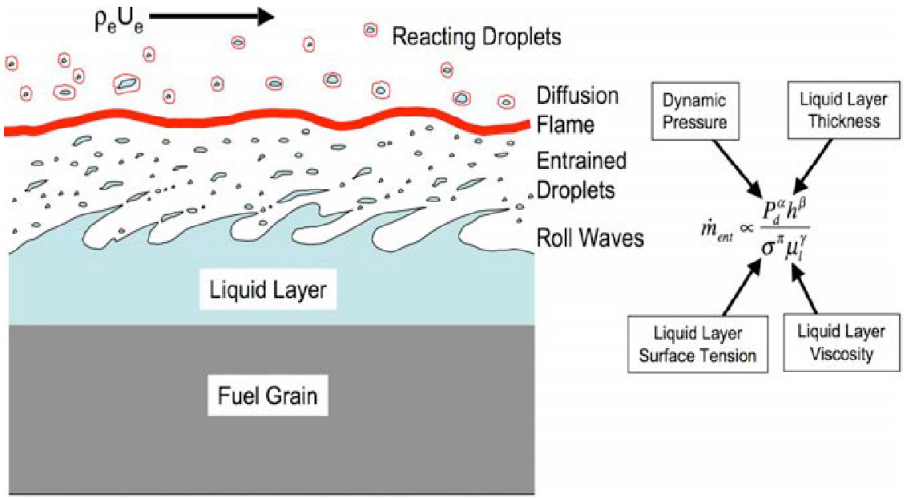


Figure 1.6: Droplets flowing from the liquid layer on the fuel grain [16]

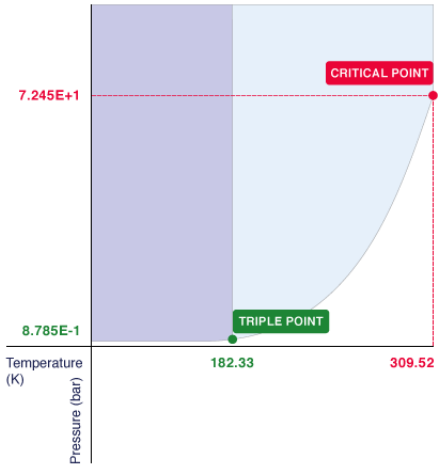
as HDPE also form a liquid layer, the surface tension and viscosity are important factors in determining the mass flow of entrained droplets, as can be seen from the formula on the right in figure 1.6 [16]. Viscosity generally increases with molecular weight but exponentially drops with the temperature of the liquid layer. Apart from for low carbon numbers, this temperature does not change much and as it turns out, paraffin waxes (C_{25-45}) have the best balance between melt layer temperature and molecular weight for substances that are solid at room temperature [16].

Finally, another approach that is being studied is the effect that oxidizer injection has on the regression rate. There are many different injection schemes, and the remainder of this work will be dedicated to gain an understanding of the injector's effect on mass flow rate and performance.

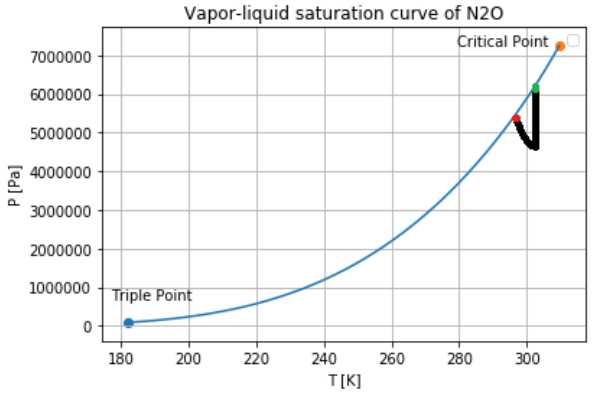
1.4 Nitrous Oxide

Nitrous Oxide (N_2O) has become a popular oxidizer choice for hybrid rocket applications due to its relative ease of handling, low toxicity, and high vapor pressure. The high vapor pressure allows nitrous oxide to be used as a "self-pressurizing" propellant in a pressure-fed rocket engine. This can eliminate the need for a turbo-pump or additional pressurization systems, reducing complexity and cost. For these reasons, Propulse NTNU decided to use N_2O as their oxidizer for the 2020 hybrid rocket. However, the fluid mechanics and thermodynamics of nitrous oxide tank expulsion are complex, making it difficult to obtain an accurate prediction of the mass flow rate [18]. In the following chapters, the injector schemes and flow modeling will be presented within the context of nitrous oxide, and therefore its characteristics will be discussed here.

To gain a basic understanding of how nitrous oxide behaves, it can be useful to observe its phase diagram, seen in figure 1.7a [19]. Here, the solid region is shown in



(a) Phase Diagram of nitrous oxide [19]



(b) Zoomed-in view of the vapor-liquid saturation line. The black line going from green to red describes the behavior as the tank empties

Figure 1.7

dark blue, meaning that for a pressure-temperature combination in this area, N_2O is in the solid phase. Likewise, the light blue region represents the liquid phase while the gaseous phase region is white. When the substance is just on the line between the liquid region and the gaseous region, it is said to be saturated. Liquids at higher pressures or lower temperatures than the saturated conditions are called sub-cooled liquids. Similarly, gases at higher temperatures or lower pressures than saturation are said to be superheated. In 1.7b, the saturation line has been plotted to show more clearly labeled axes. The code for generating this plot can be seen in Appendix A.

The self-pressurizing nature of N_2O as a rocket propellant can be seen from the phase diagram. The oxidizer tank (also known as the run tank) is initially empty, in an upright position and at atmospheric pressure. Then, liquid N_2O is drawn from a external storage tank into the run tank up to a predetermined fill height, introducing the liquid to a low-pressure environment. The phase diagram shows that at

room temperature and low pressures, nitrous oxide is gaseous. Thus, some of the liquid will evaporate. This divides the tank into an upper part containing vapor, known as the ullage, and a lower part containing liquid. As the tank is sealed, this evaporation will raise the pressure in the tank. This continues until the pressure reaches the saturation line that separates the two phase regions. At this point, the phases are in equilibrium, and the rate at which the liquid evaporates is exactly balanced out by the condensation of vapor to liquid. The pressure that the vapor phase exerts at this point is known as the saturated vapor pressure (P_v). Whenever "vapor pressure" is mentioned throughout this work, it will be at saturated conditions. Because its vapor pressure is quite high, close to 60 bar at 25°C, nitrous oxide can reach the pressures required for rocket applications on its own and is thus called a self-pressurizing propellant. By heating or cooling the tank, it is possible to tune the vapor pressure and thus the pressure of the tank. However, one should be careful to not go beyond the critical point, where nitrous oxide becomes a super-critical fluid. This can easily happen when launching in the hot deserts of the US, so using some form of a cooling system should be considered. The use of super-critical N_2O for extraction processes has resulted in explosions in the past and it is generally not used intentionally for hybrid rocket propulsion [20].

When the control valve is opened, liquid begins to flow out of the tank, through the injector and into the combustion chamber. This leads to an expansion of the volume that the vapor occupies in the tank, lowering the pressure. Due to this pressure drop, some of the liquid will evaporate to regain the pressure and reach equilibrium once again. This is known as a "VaPak" system and can be seen in figure 1.8 [4].

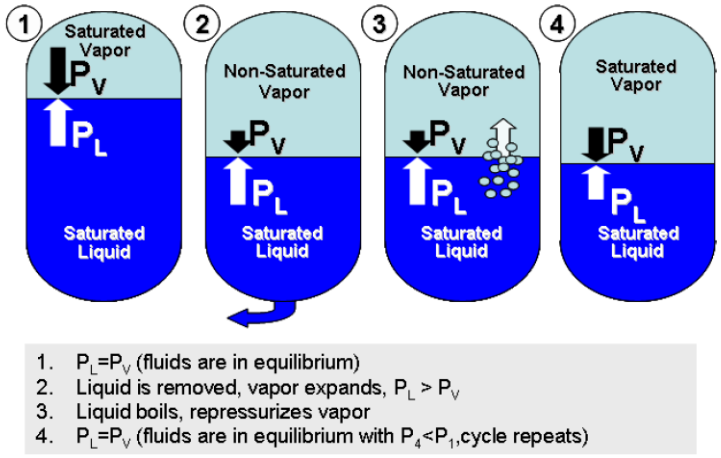


Figure 1.8: Tank being emptied from initially saturated conditions [4]

In figure 1.7b, the black line going from the green to the red point represents an exaggerated, simplified way of how the nitrous oxide in the tank would behave when operating as a VaPak system for as long there is liquid remaining in the tank. When some liquid is removed, the vapor expands, lowering the pressure. Evaporation brings the N_2O back to saturation, but at a slightly lower pressure and temperature than previously due to heat loss. This will be further expanded upon in chapter 3, where the tank dynamics are modeled.

Typical pressure and temperature time histories for a carbon dioxide tank being emptied in this manner, known as a blowdown mode, can be seen in figure 1.9 [18].

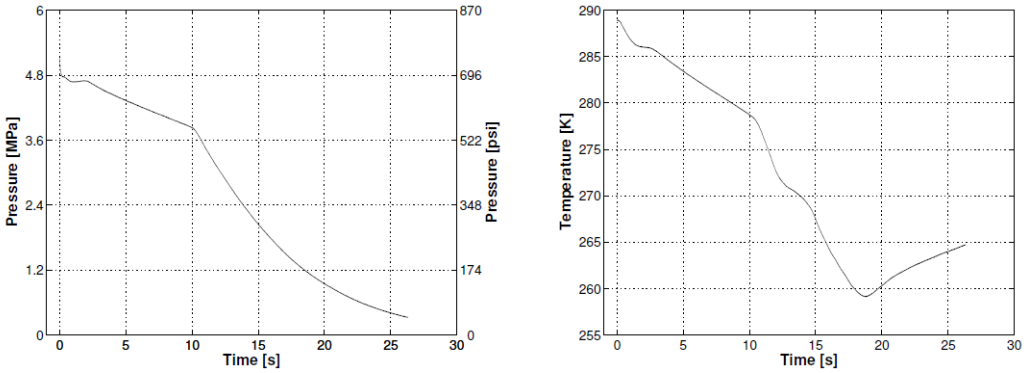


Figure 1.9: Pressure and Temperature time histories for a cold-flow test using initially saturated CO_2 [18]

Carbon dioxide is often used as an analog for nitrous oxide during testing due to the similarities in their thermodynamic properties and safety concerns. Following some initial transient behavior, the pressure time history follows an approximately linear drop up until about the 10-second mark of this test. The sharp cusp and increase in slope at this point happens when all the liquid is removed from the tank and there is only gas flowing out [18]. The temperature plot shows a similar curve, and when comparing the two graphs, it seems that the CO_2 in the tank stays quite close to saturated conditions for the duration of liquid expulsion.

1.4.1 Operating Modes and Safety

So far, it has been shown that nitrous oxide has the ability to reach high pressures from its vapor pressure alone. However, it is not unusual to use helium or other inert gases to pressurize the tank beyond the vapor pressure to sub-cool the liquid. This is known as supercharging, and the supercharge pressure upstream of the injector can be defined as shown in equation 1.9, with P_1 denoting the upstream pressure.

$$P_{sc} = P_1 - P_v \tag{1.9}$$

This operating mode can be used to get higher pressure drops and thus mass flow rates of the oxidizer at a given temperature. Its primary purpose, however, is to address safety concerns regarding various decomposition events of nitrous oxide vapor. While these events are rare, they have resulted in lethal accidents. Nitrous oxide has a positive heat of formation, and the heat released during decomposition to gaseous oxygen and nitrogen can potentially cause runaway reactions that raise the tank pressure very rapidly and even ignition. It is extremely important to avoid any contamination of hydrocarbons in the tank and feed system, and if parts made of hydrocarbons (rubber, etc.) are used it must be checked that they are not soluble in N_2O. Detailed modeling of nitrous oxide decomposition was done by Karabeyoglu et al. [21], and it is highly recommended to heed the guidelines presented in

the conclusion of their work. Another paper on N_2O handling considerations was also written by Thicksten, et.al [22], which should also be useful to Propulse. One recommendation includes supercharging the nitrous oxide in the tank ullage, which reduces the risks of decomposition events in the tank and reduces the chance of cavitation in the feed system. When N_2O is in a saturated state, small changes in pressure or temperature can cause the liquid to evaporate, as was shown earlier. This can be problematic if significant flash vaporization of liquid N_2O occurs in the feed system due to pressure losses as the nitrous oxide flows from the tank to the injector. In addition to the potential decomposition events from the presence of vapor in the feed system, two-phase flow in this area could make accurate mass flow modeling very difficult as liquid-vapor mixture properties in the pre-injector volume are not readily obtainable. Furthermore, venturi flow rate measurements are unreliable for two-phase flows, making it more difficult to gather reliable mass flow rate data. Sub-cooling the liquid by supercharging it makes it possible for a pressure drop to happen without inducing a phase change.

That being said, the most common approach is to use nitrous oxide at saturated conditions. This will be the focus of this work as the members of Propulse NTNU have decided to not use supercharging for their 2020 hybrid rocket, primarily due to cost and simplicity. While this does increase the risk of decomposition events, these are still rare and should be avoidable as long as strict cleaning processes and other handling considerations of N_2O are followed [22].

Another point of note for both supercharged and non-supercharged operating modes is the gas-only flow of the nitrous oxide towards the end of the tank expulsion. While it is technically possible to use the gas-only flow as an oxidizer source to get more total impulse for the rocket, this is generally not recommended. This is because the gains are usually small due to a fall in combustion chamber pressure when operating in this mode [23] and because the gas-only flow of nitrous oxide is particularly hazardous [21]. The decomposition events associated with nitrous oxide vapor have already been mentioned. Additionally, as was shown in figure 1.9, the pressure drops very rapidly in the tank when only gas is flowing out. While the chamber pressure also falls, this still increases the chance of hot gases from the combustion chamber propagating back up the feed system. This is known as blow-back and can result in catastrophic explosions. Therefore, it is recommended to close the control valve at, or slightly before, the point of liquid run-out.

1.4.2 Two-Phase Flow

Another important characteristic of nitrous oxide to consider is that two-phase flow has a high chance to develop inside the injector orifices. This is because the liquid N_2O upstream of the injector is often very close to or at the vapor pressure. When the propellant accelerates through the injector, local static pressures inside the injector can fall below the vapor pressure. This results in the formation of a significant amount of vapor [12]. Observing the liquid-vapor dome diagram of nitrous oxide, shown in figure 1.10 [24], can be useful to visualize this. The figure shows how the liquid goes from a supercharged state upstream of about 8 MPa to a target chamber pressure of about 4 MPa. It is clear that as the nitrous oxide goes from the injection point to the chamber conditions, it passes through the liquid-vapor dome. Thus, some vaporization from liquid to vapor is likely [24].

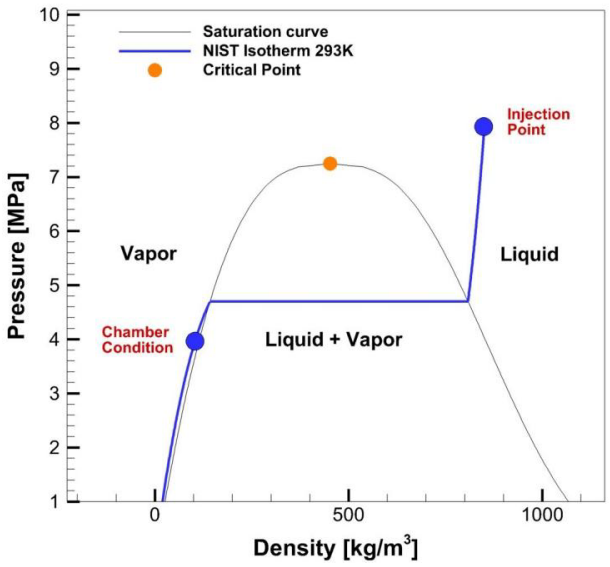
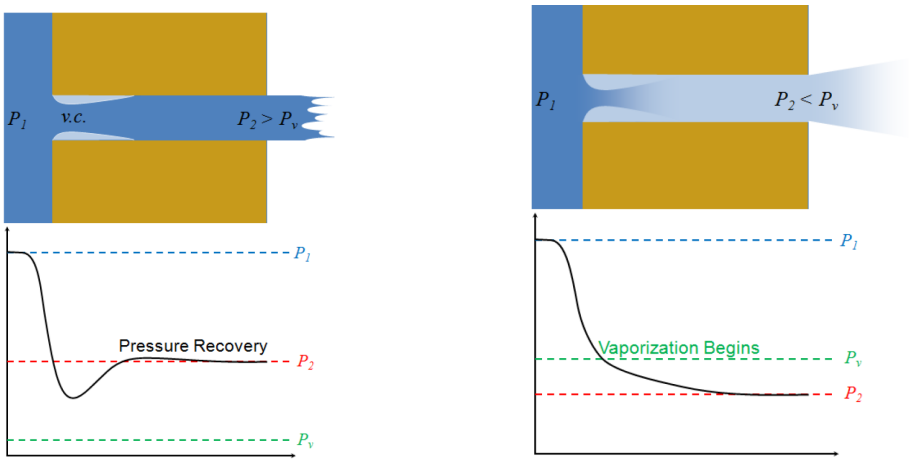


Figure 1.10: Pressure-density vapor dome diagram for nitrous oxide injection [24]



(a) Low vapor pressure (b) High vapor pressure

Figure 1.11: Injector pressure history for a low vapor pressure and a high vapor pressure propellant, originally made by Dyer et al. [12]

In figure 1.11 [12] the behavior of a high vapor pressure propellant and a low vapor pressure propellant are compared. For the low vapor pressure case, the fluid loses pressure as it is accelerated, but it recovers towards the downstream chamber pressure as the flow approaches the exit. Flow separation causes the vena contracta, denoted by v.c. For the high vapor pressure case, the chamber pressure is now lower than the vapor pressure. Thus, as the bulk pressure drops below P_v, a significant amount of vapor forms, limiting the mass flow.

For a given upstream pressure, experiments show that the flow rate reaches a maximum as the downstream pressure drops below a certain value. This is known as choked or critical flow and must be accounted for when attempting to model the mass flow rate. Choking is usually associated with gaseous flows but also occurs for two-phase flows. Neuterium.net [25] is a knowledge base on engineering topics that explains the concept concisely:

"As a compressible fluid reaches the speed of sound, pressure changes can no longer be communicated upstream as the speed of which these pressure changes are propagated is limited by the speed of sound. In a nozzle or restriction this has the effect of isolating the upstream side from the downstream side at the throat. Because of this effect any reduction in downstream pressure will have no effect on the flow rate, as the increased pressure differential is not 'felt' upstream of the restriction" [25].

The threshold for the onset of critical flow depends on many factors in addition to the downstream pressure, especially the length-to-diameter ratio of the injector orifice. Experiments on high L/D ($\sim 10 - 15$) injectors used with supercharged nitrous oxide by Waxman et. al [12] resulted in a useful criterion for the onset of critical flow:

$$P_2 < 0.8P_v \tag{1.10}$$

Do note that while this could be helpful, none of the models presented in chapter 3 will be able to account for L/D ratio effects. However, this criterion can be useful as a comparison tool while validating the models and highlights the idea that lower L/D ratios will require lower P_2 for choked flow to occur.

The critical flow phenomenon will be very important throughout this work. It may be possible to utilize choked flow to maintain a more stable mass flow rate despite pressure oscillations in the burn chamber, which tend to happen during combustion. Choked flow could help to reduce or eliminate feed system coupled instabilities. Combustion instabilities are considered one of the most difficult issues to overcome during rocket design. The term is used to describe unwanted, often violent pressure and thrust oscillations that can occur when firing the engine. Choking the flow can help eliminate one of these types of instabilities, and is an interesting option that should be considered [12]. Additionally, none of the flow rate models will automatically account for choked flow, so it is important to be aware that choked flow must be imposed to reflect the actual physics of the flow.

1.4.3 CO_2 as an Analog to N_2O

While nitrous oxide is relatively safe, it remains an energetic oxidizer and there are some safety risks associated with its use, particularly with regards to the decomposition events mentioned earlier. Strict cleaning procedures must be followed to minimize the possibility of explosions. However, accidents still happen and therefore nitrous oxide is not always ideal to use in academic settings [12].

Carbon dioxide, on the other hand, is an inert gas. Therefore it is much safer to handle than N_2O, and multiple groups have identified that CO_2 can be used as an analog to nitrous oxide for fluid flow studies and cold-flow testing. This is because most of their thermodynamic properties are very similar, as can be seen in table 1.2 [12].

Property	Units	N_2O	CO_2	% difference
Molecular Weight	amu	44.013	44.010	-0.007
Critical point Pressure	MPa	7.25	7.38	+1.8
Critical point Temperature	°C	36.5	31.1	-1.74
Critical point Density	$\frac{kg}{m^3}$	452	467.6	3.45
Critical point Compressibility Factor	—	0.273	0.274	+0.366
Triple point Pressure	MPa	0.09	0.52	+491
Triple point Temperature	°C	-90.2	-56.6	+18.8

Table 1.2: Comparison of thermodynamic properties of N_2O and CO_2 [12]

Although most of them are very similar, there are some deviations in the thermodynamic properties of the two substances. This is particularly evident in the triple point properties. However, the triple point is unlikely to be reached for self-pressurizing propellant tank blowdown and should not get in the way of using CO_2 as an analog [12]. One possible issue, though, could be for cold-flow testing to ambient conditions. Atmospheric pressure is lower than the triple point pressure of CO_2, which could potentially result in a solid-vapor mixture leaving the injector. A potential remedy to this could be to use a pressurized chamber downstream of the injector for cold-flow testing, which will be discussed further in chapter 4.

Experiments performed by Waxman et. al [12] confirm that CO_2 and N_2O can produce similar mass flow rates. In figure 1.12 [12] tests at different supercharge levels are shown. The data shows how the mass flow rates of the two different fluids are quite similar at similar supercharge levels. More specifically, most cases show that the mass flow rates are essentially equal during the non-choked region, while CO_2 tends to have slightly higher values of the critical mass flow rate. Because P_{sc} and the temperatures are different for the N_2O and CO_2 tests in figure 1.12 one cannot conclusively say that the analogy is valid. However, if one plots the critical mass flow rate against the supercharge level, the similarity of the two substances can be seen more clearly. The results are presented in figure 1.13 [12] and show that CO_2 and N_2O have similar critical flow rates that are at least within 10% of each other.

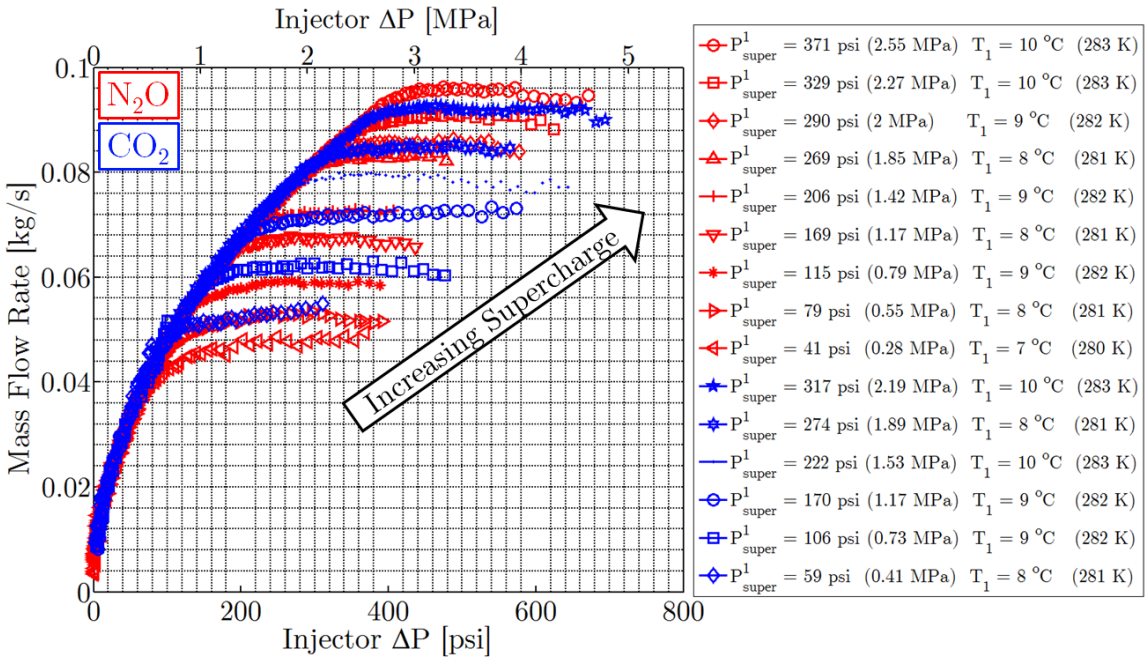


Figure 1.12: Mass flow rates of CO_2 (in blue) and N_2O (in red) for different tests [12].

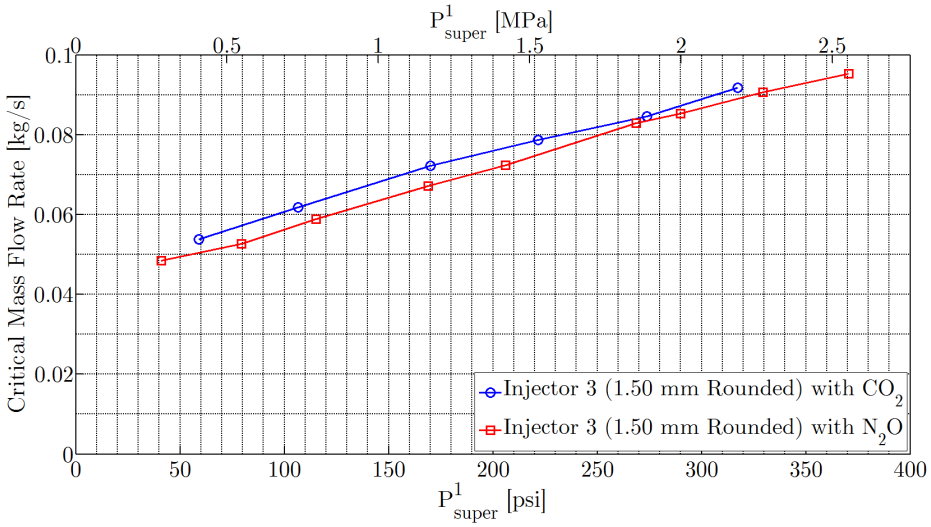


Figure 1.13: Critical flow rate N_2O and CO_2 for varying supercharge levels [12]

Chapter 2

Atomization & Injection schemes

Many different injector schemes can be used to enhance the atomization of the oxidizer and introduce other advantageous flow characteristics. Atomization is important as smaller droplets have a larger surface-area-to-volume ratio, allowing for faster vaporization of the oxidizer, which then can take part in combustion. If too many liquid droplets hit the surface of the fuel grain, the flame can be extinguished or unstable combustion can occur. Minimizing the amount of unburnt propellant obviously boosts the combustion efficiency as well. Other important functions of the injector are to provide an even distribution of oxidizer and ensure good mixing of the fuel and oxidizer. This is largely dependent on the flow pattern and turbulence of the flow [6], and also enhances combustion stability and efficiency [26].

The different injector configurations that are presented in this chapter have been selected because they seem to be the most promising options for Propulse NTNU to use in terms of their effect on performance and feasibility.

2.1 Introductory atomization theory

There are two main modes of atomization that are the most relevant to discuss for this thesis. The first is the mechanical breakup mode, also known as pressure atomization. In liquid jets of high velocity and small diameter, such as the ones caused by straight-holed injector orifices, instabilities tend to occur. These instabilities make the jet to break up into small droplets. This mode is generally governed by viscous, aerodynamic, and shear effects [27] [28].

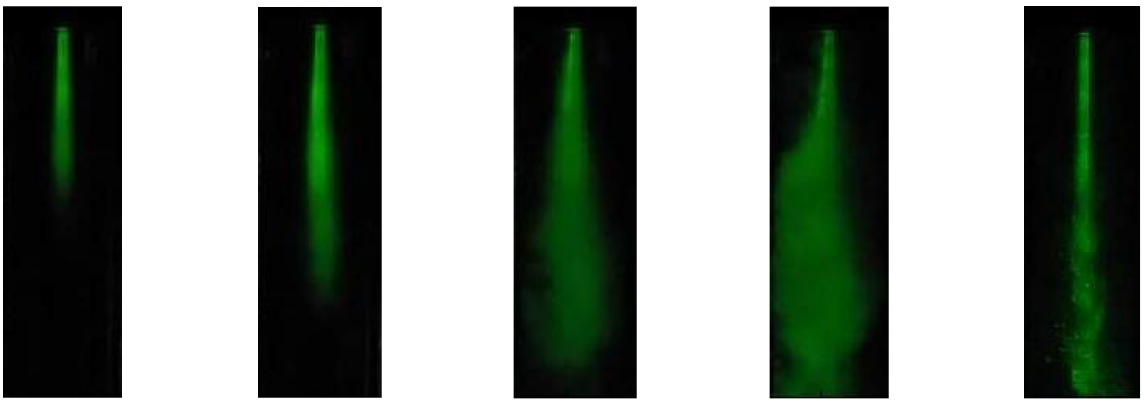
An important non-dimensional number that is often used to characterize liquid jet sprays is the Weber number, seen in equation 2.1:

$$W_e = \frac{\rho u^2 D}{\sigma} \tag{2.1}$$

The Weber number gives the ratio of the inertia of a fluid to its surface tension. Higher Weber numbers provide a higher degree of atomization, considering that the surface tension that holds the fluid together is then small compared to the inertial forces. Some sources say that Weber numbers greater than 50 are needed to create small droplets [26]. For liquid nitrous oxide in rocket applications, it is common to

have Weber numbers greater than 10^4 due to high fluid velocities and low surface tension [27].

The second mode is a result of flash vaporization in the jet. The sudden pressure drop across the injector turns the jet to a metastable, superheated state. With enough superheating and the presence of bubbles or nucleation sites, rapid bubble growth can make the jet break in a more abruptly and produce much smaller droplets than the mechanical breakup mode does. Experiments by Mojtabi et al. [29] on gasoline injection found that the degree of superheating needed primarily depends on the temperature difference and the vapor pressure at the injector exit, but is also affected by the surface finish of the injector orifices, the Weber number of the liquid jet, and the L/D ratio of the orifices. It has already been shown that two-phase flow tends to develop when injecting high vapor pressure propellants. Waxman, Cantwell, and Zilliac [27] found that this mode is indeed the dominant mode of atomization for the majority of cases during their experimental campaign. In figure 2.1 [27] pictures taken from CO_2 tests on the same injector for the two different modes are shown, highlighting the differences between the two modes.



(a) (b) (c) (d) (e)

Figure 2.1: Flash vaporization mode with P_{sc}=350kPa in figures a)-d), with ΔP=35kPa, 350kPa, 700kPa, 1.4MPa respectively. Mechanical breakup mode in e), with P_{sc}=2MPa,ΔP=35kPa [27].

The flash vaporization mode is identified by the creation of an aerosol cloud downstream of the injector. The only cases that exhibited the mechanical mode were when the supercharge pressure was very high and the pressure drop was very small. In 2.1e, an aerosol cloud is not visibly created and one can see the jet break up into larger droplets when compared to the other cases.

2.1.1 Showerhead Injectors

The atomization theory that has been shown so far will be relevant for any injector design, including simple straight holed injector orifices. These types of injector orifices are usually used in the most basic injector scheme, the showerhead injector. This injector configuration can simply be a collection of straight-holed orifices of diameter D and length L on a plate, as shown in figure 2.2 [30]. The orifices typically have a diameter of 1-2mm, with L/D ratios in the range of 3-15 [12]. Showerhead injectors are the easiest injector type to design and manufacture. Considering how the previously shown theory and experiments indicate that even straight-holed injector orifices can produce an aerosol spray of quite fine droplets when using nitrous oxide, this design may be sufficient for some applications. Moving forward, the showerhead injector design will be used as a baseline for comparisons with the more advanced impinging and swirl designs.

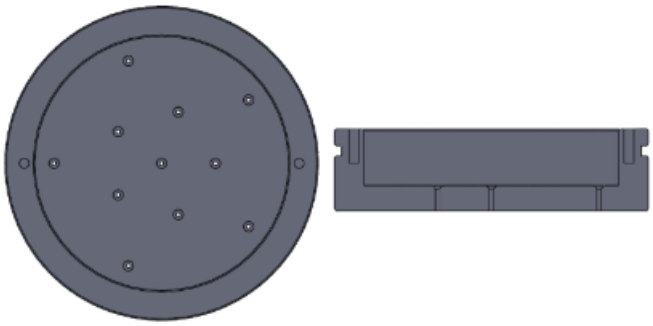


Figure 2.2: CAD of a showerhead injector [30]

2.2 Impinging Injectors

This injector type utilizes orifices that are angled against each other so that the jets they produce impinge onto each other, as shown in figure 2.3 [31]. In simple terms,

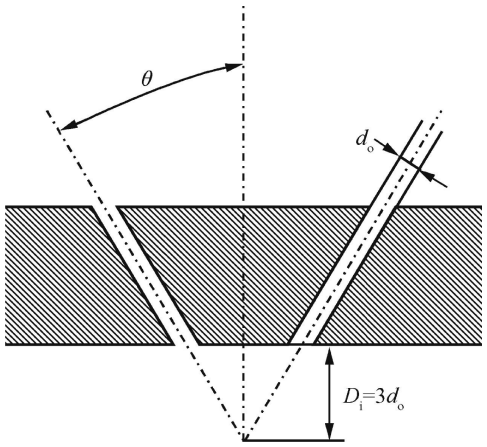


Figure 2.3: A doublet impinging injector. Note that the expression for D_i shown is only valid for this particular case [31]

impinging jet streams onto each other forms a thin, unstable sheet of liquid that disintegrates into small droplets [32]. The different breakup regimes of impinging jets will not be discussed in detail here, as this is a complicated topic and a field of study on its own. For the purposes of this thesis, it will be sufficient to simply state that the impingement of jet streams is commonly used to aid atomization.

The impingement half-angles θ that are seen in the literature usually range from 22.5-45 degrees. 45 degrees seems to be particularly popular, but the reasoning behind the choice of angle is not often reported, unfortunately [27][6][33]. However, NASA researchers from Lewis Research Center [34] have conducted some studies on the effects of impingement angle. In 1961, they performed experiments on injectors using water with impingement angles (2θ) ranging from 10 to 90 degrees. The general takeaway was that for higher injection angles, the mean droplet size was smaller. Additionally, a 1972 study by the Lewis Research Center [35] found that increasing the impingement half-angle up to 45 degrees gave increased combustion stability for a liquid rocket using an oxygen-hydrogen propellant combination. Another potential reason for using 45 degrees could be that the impingement point of the streams would be closer to the injector, allowing for a shorter pre-combustion chamber.

Most of the literature regarding impinging injectors is related to liquid rockets, and there are many different impinging injector varieties. They are often characterized into like-on-like and unlike injectors. Like-on-like injectors are also known as self-impinging, refers to injectors where oxidizer impinges against other oxidizer jets, and likewise for fuel-against-fuel. In contrast, unlike injectors have fuel jets impinge onto oxidizer jets, which can be used to enhance their mixing. Additionally, another means of characterization is how many jets impinge on each other. In figure 2.4 [36] sprays from both a doublet and triplet impinging injector element are shown - the name indicating how many streams are impinging on each other. Experiments by

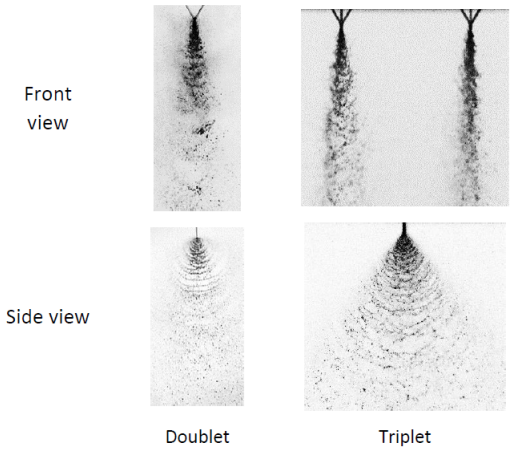


Figure 2.4: Sprays from a doublet and triplet injector with water [36]

Indiana et. al [36] on the sprays in figure 2.4 found that the addition of the central jet in the triplet configuration did not significantly affect the spray topology. Furthermore, the triplet configuration generally produced slightly smaller Sauter mean diameters for the droplets. For a liquid rocket with ethyl alcohol and hydrogen peroxide, they found that the triplet injector produced more homogeneous mixtures.

Combustion efficiency was similar between doublets and triplets for long combustion chambers, but the triplet was more efficient for shorter chambers. For liquid rockets, Sutton [6] recommends using the unlike doublet configuration for cases where the volume flow (i.e. orifice diameter) is equal for the oxidizer and fuel streams, and the triplet configuration is better for uneven flow.

However, impinging injectors for hybrid rockets are naturally like-on-like, as only the oxidizer is stored as a liquid. Therefore, it is primarily the atomization features of impingement that is relevant. Research on impinging injectors for hybrid rockets is limited, but the atomization characteristics described for liquid rockets should be valid for hybrid rockets as well. For the case of self-pressurizing propellants, even less literature exists. However, the experiments by Waxman et. al [27] show that the impingement of CO_2 jets enhances atomization. When comparing the pictures in figure 2.5 [27], where impinging injectors are used, to figure 2.1 it is clear that impingement indeed produces a higher degree of atomization. The figure

(a) (b) (c) (d) (e)

Figure 2.5: Impinging doublet: Flash vaporization mode with P_{sc}=350kPa in figures a)-d), with ΔP=35kPa, 350kPa, 700kPa, 1.4MPa respectively. Mechanical breakup mode in e), with P_{sc}=2MPa,ΔP=35kPa [27].

shows impingement also gives enhancements for the flash vaporization mode. This is supported by research done by Kuo et. al [37], which suggested that impinging flash atomizers could result in a wider spray with finer, more evenly distributed droplets. Cold-flow experiments by Gamper and Hink [26] also found that the injection of nitrous oxide using this method provides a relatively homogeneous distribution while atomizing the oxidizer well. As such, impinging injectors can be a good option for hybrid rockets, especially in cases where sufficient atomization is critical or proves difficult to achieve.

2.3 Swirl injectors

Swirl injectors are a very promising class of injectors. A swirl injector element can be seen in figure 2.6 [38]. A full injector will usually consist of multiple swirl elements on the injector plate. These injectors have tangential inlets that go into a

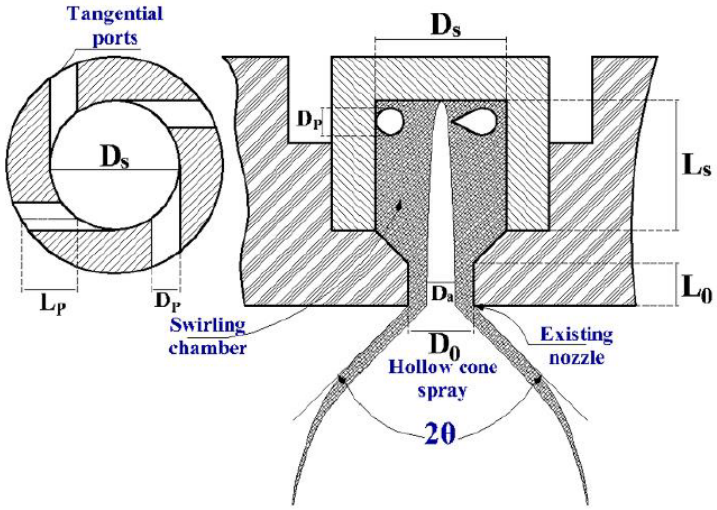
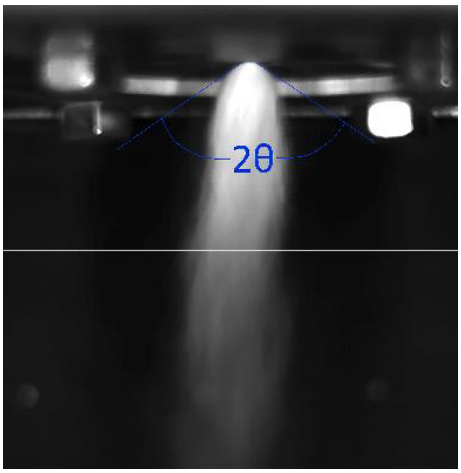
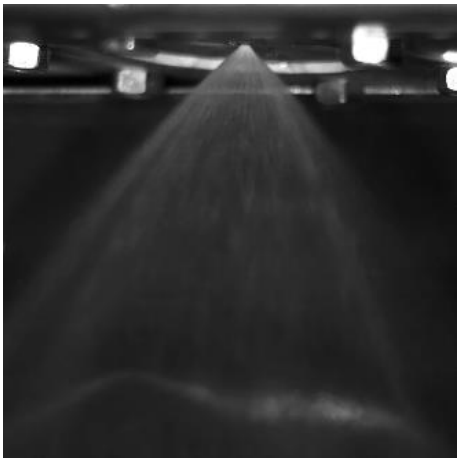


Figure 2.6: Schematic of a swirl injector element [38]

swirl chamber, where the flow swirls around the walls with an air core in the center. The rotational momentum of the flow results in a cone-shaped vortex sheet spray. In figure 2.7 [38], pictures of the exiting flow pattern from a swirl injector element illustrate this more clearly.



(a) Swirl injector exit using N2O



(b) Swirl injector exit using Water with a pressure drop of 30 bar

Figure 2.7: Swirl flow pictures taken by Bouziane et al. [38]

Many hot-fire tests show that swirl injectors can increase regression rates significantly when compared to a traditional axial injection with the same mass flow rate of oxidizer. The increase varies greatly between different tests, with cases reporting

everything from a 16% to a 700% increase [13] [39]. The latter is an extreme case and should probably be taken with a grain of salt, but multiple cases report the regression rate increasing by factors of 2-3. Unfortunately, there does not seem to be much literature on how to design swirl injectors to achieve a specific regression rate increase, which is likely dependent on several factors. What is clear from the various experiments throughout the literature, though, is that introducing swirl will increase the regression rate. Part of the reason for this is that the centrifugal force of the swirling flow will drive the flame closer to the surface of the solid fuel grain, increasing heat transfer to the fuel grain [39]. An additional reason that has been suggested for the increased regression rates is that with a tangential velocity component, the effective velocity that governs the "apparent" oxidizer flux is increased. This may somewhat alter the regression rate equations that were shown earlier, but the general point is that as the apparent oxidizer flux increases, the regression rate does the same as it is dependent on the oxidizer flux [13]. Another advantage swirl injectors have over the standard axial injectors is that the flow downstream of the injector may form a recirculation zone that protects it from heat transfer. This means that using a swirl injector could allow for a shorter pre-combustion chamber, as the high temperatures of the chamber are not going to affect the injector as much [39]. Furthermore, swirl injectors have been shown to improve combustion stability for some cases, which is also attributed to the recirculation zone. The recirculation could let the oxidizer be pre-heated and stabilize the flame sheet, preventing flame-holding instabilities [40].

While designing these injectors, there are a few things to keep in mind in terms of the geometry of the swirl element shown in figure 2.6. $\frac{L_s}{D_s}$ should be minimized to avoid friction losses, but needs to be bigger than 0.5 to stabilize the liquid flow and generate a uniform vortex sheet. For proper design, a recommended value of this ratio is 1. To minimize friction losses at the exit, $\frac{L_0}{D_0}$ should also be reduced. The $\frac{L_p}{D_p}$ ratio should also be larger than 1.3, as short inlet orifices may cause an unstable spray. As is clear, these swirl injectors have more "sources" of friction losses than simple orifices. Additionally, the existence of the air core makes the estimation of the discharge coefficient quite different for these injectors. The discharge coefficient (C_d) is a friction loss parameter that is very important for mass flow rate modeling and will be discussed in further detail in chapter 3. There are a few different empirical formulas for a swirl injector's C_d. A convenient one is shown in equation 2.2, with the discharge coefficient being primarily influenced by $0.19 < \frac{A_p}{D_s * D_0} < 1.21$ and $1.41 < \frac{D_s}{D_0} < 8.13$ [38][41].

$$C_d = 0.35 * \left(\frac{D_s}{D_0}\right)^{0.5} * \left(\frac{A_p}{D_s * D_0}\right)^{0.25} \tag{2.2}$$

While using swirl injectors can be an efficient way to increase the regression rate with some added benefits, the rocket designers must consider if this is necessary. Depending on the optimal O/F ratio, higher regression rates may not always be desired. This can be particularly true whilst using high regression rate fuels such as the paraffin wax that Propulse intends to use. Too high regression rates could lead to decreased performance due to the change in the O/F ratio. Burning through the fuel grain too fast could even damage the chamber walls or the structural integrity of

the fuel grain itself. A highly fuel-rich mixture could also lead to significant amounts of unburnt fuel exit the nozzle. However, for traditional fuel materials such as HTPB that have historically had low regression rates, swirl injectors should be particularly useful. In any case, it could be wise to experiment with showerhead or impinging injectors first, to determine whether the regression rate needs further increases.

2.3.1 Vortex Injectors

Before moving forward, it should be mentioned that the swirl injectors from the previous section are sometimes referred to as vortex injectors in the literature. Although they are in many ways similar, a distinction is made here. See figure 2.8 [38] for a schematic of a vortex injector.

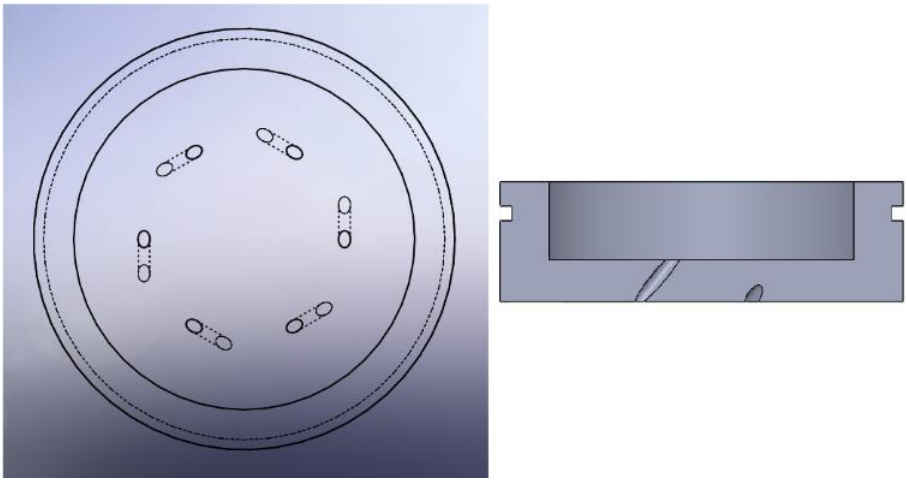


Figure 2.8: Schematic of a vortex injector with 45°inclined orifices [38]

The vortex injectors also introduce a swirling or vortex flow into the combustion chamber, but instead of using a tangential inlet and a swirl chamber, they simply have inclined outlets. In the example from the figure above, the whole orifice is inclined as well. Thus, the flow gains a tangential velocity component and results in the flow pattern that can be seen in figure 2.9 [38]:

(a) Vortex injector exit using N_2O

(b) Vortex injector exit using Water with a pressure drop of 30 bar

Figure 2.9: Vortex flow pictures taken by Bouziane et al. [38]

Of note is that when comparing the water cold flow pictures of the swirl and vortex injectors, it seems as if the swirl variety shows a higher degree of atomization. However, in 2.9a, the flow seems well atomized. If sufficient atomization can be obtained with this method, it could be an alternative that has many of the benefits of swirling flow while being easier to manufacture and design than the swirl injectors.

Additionally, the vortex injector is much more similar to the showerhead and impinging designs than the swirl injector is. As the work now will focus on mass flow rate modeling, it will be much easier to adapt the models to the vortex design than the swirl injector. This shows once again that the vortex injector might be a lot simpler to work with than swirl injectors, while still providing some of the same benefits. For an inexperienced group like Propulse NTNU, this could be particularly important.

Chapter 3

Oxidizer flow modeling

While the effect different injector configurations can have on the rocket's performance is certainly interesting, the most important function of the injector is their role in determining the mass flow rate of oxidizer. From the previous chapters, it should be clear that understanding the mass flow rate of oxidizer is crucial as it is a deciding factor in many of the performance indicators of any hybrid rocket design. The injector consists of very small orifices. Due to their low cross-sectional area, they will normally act as the flow limiter, thus controlling the mass flow rate. In some cases, a cavitating venturi could be placed upstream of the injector to limit the mass flow rate with a different approach. Cavitating venturis are a constricted part of the pipe that can choke the flow to limit the flow rate, typically in the form of a converging-diverging section. As mentioned in section 1.4, however, nitrous oxide vapor is generally undesirable in the feed-lines. This would also require an additional component, so allowing the injector to be the flow limiting device is beneficial in terms of weight and system complexity. Waxman et al. [12] developed a novel injection scheme where the injector orifice itself is a cavitating venturi, forcibly choking the flow. This design could be interesting, but will not be discussed further in this thesis as it was deemed difficult to manufacture with Propulse's current capabilities.

Many of the calculations that are performed when designing the rocket, such as thrust calculations, often assume a constant oxidizer flow rate. This was also done by various Propulse members during the design of the engine as it was assumed that this would be the case. Achieving a constant flow rate would be ideal for the stability and predictability of the rocket's behavior and could prove beneficial to minimize issues such as the O/F shift mentioned in chapter 1. Certain design choices and operating conditions may allow for a constant oxidizer flow rate, and the models and theory presented in this chapter will try to shed light on whether it can be done or not. In any case, being able to predict the oxidizer flow rate would be very useful when designing injectors so that they can provide the appropriate flow rate without requiring much modification after testing.

There are a variety of models that can be used to predict what the oxidizer mass flow rate that is injected into the combustion chamber will be. For many traditional oxidizers, such as liquid oxygen, the modeling can be relatively simple as they often can be assessed as incompressible liquids with reasonable accuracy. However, for high vapor pressure oxidizers like the N_2O that Propulse intends to use, the onset

of two-phase flow in the injector and the effect of phase changes on tank dynamics complicates the analysis. That being said, the choking aspect of two-phase flow can be beneficial, as was discussed in section 1.4.

That being said, the simplest models can provide an invaluable baseline and will be presented first, before moving towards the more complex two-phase models. In order of appearance, the mass flow rate models that will be shown are:

- The single-phase incompressible liquid model
- The single-phase perfect gas model
- The two-phase homogeneous equilibrium model
- The two-phase Dyer model (non-homogeneous, non-equilibrium)

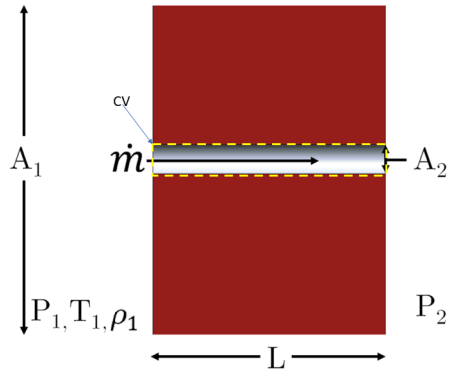


Figure 3.1: Schematic of a simple straight-hole injector orifice [12]

While exploring the different models, straight-holed circular orifice injectors will be considered for the sake of simplicity, as shown in fig 3.1 [12]. However, the models can potentially be adapted for different injector hole geometries. In some cases, this adaptation could be as simple as adjusting the discharge coefficients, a friction loss parameter that will be discussed further throughout this thesis. The subscripts 1 and 2 in figure 3.1 represent the locations upstream and just downstream of the injector, respectively. The variables that are shown in the figure are the ones that are generally known when attempting to calculate the mass flow rate. The pressure downstream in the combustion chamber will usually be chosen by the rocket designers to achieve a certain thrust level. The initial upstream conditions must be chosen such that flow from location 1 to 2 can be established and maintained until the tank runs out of liquid. The models of this chapter will initially calculate the mass flow rate for a constant P_1 over a range of different P_2. This can be useful for cases where the upstream conditions do not change much as the tank is being emptied and to assess choked flow.

Then, the models will be adapted to account for varying P_1. As was shown in section 1.4, the upstream pressure in Propulse's case is likely to drop as the tank is emptied, and this will be an important factor for the mass flow rate. Therefore, the goal of having a constant mass flow rate may not be possible for Propulse's

current design. However, an average flow rate can be found as a substitute if the upstream changes can be modeled, and will be the goal for the varying upstream cases. Alternatively, Propulse could consider using an external pressurizing gas to maintain the upstream pressure, although this would somewhat defeat the purpose of using the self-pressurizing N_2O. As this is not currently in Propulse's plans, the upstream pressure changes will be assessed in two different ways. The results of the first variation will be shown as each mass flow model is presented, while the second will be detailed at the end of the chapter.

In the first variation, the upstream pressure is simply assumed to drop linearly for the duration of the operation, from predesignated initial and final values of P_1. To avoid blow-back it is very important that the choice of the final P_1, which represents the pressure when the control valve is closed, is such that it is sufficiently above P_2. One should try to be as accurate as possible though, to avoid having too much unused propellant by closing the control valve while there is still a significant amount of liquid left in the tank. It is also possible to choose a final P_1 that is above P_2, but still too low, as gas-only flow can still happen in that case. Assuming a linear pressure drop in this manner is a simple method, but as shown in figure 3.2 [18] and figure 1.9, the upstream pressure does tend to fall almost linearly during liquid expulsion. Therefore, this method is a simple way to obtain reasonable results based on empirical data.

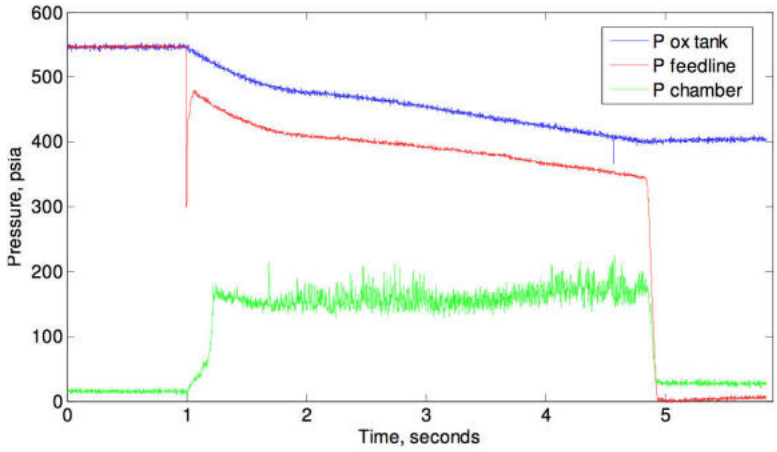


Figure 3.2: Pressure histories of the tank, feedline and combustion chamber for a hybrid motor test firing [18]

The upstream variations will be assessed in two different ways, the operating downstream pressure will be assumed constant for the duration of the burn for both methods. This means that even as the tank empties and the upstream pressure falls, the mass flow rates will always be calculated at the same operating P_2. This assumption of constant downstream pressure may not be entirely accurate, as one could expect it to be affected by for example changes in mass flow rate. Unfortunately, it is difficult to measure the pressure in the extreme conditions of the burn chamber, so the available data is limited. However, judging from the test data shown in figure 3.2, the assumption could be reasonably valid. The combustion chamber pressure

seems to oscillate around some value just short of 200psi, while the tank and feedline pressures fall almost linearly, as expected. However, when assuming a constant P_2 the models cannot capture initial transient behavior, as the chamber pressure will in actuality be atmospheric to begin with. When the control valve opens, the oxidizer flows into the burn chamber and combustion starts, quickly pressurizing the chamber.

Another important takeaway from figure 3.2 is that the feedline pressure is lower than the tank pressure, indicating a significant pressure loss in the feed system. This has not been accounted for in any of the models presented in this work. Instead, the common assumption that the feedline/pre-injector pressure P_1 is equal to the tank pressure has been used. To avoid having too much of a pressure difference between the tank and the pre-injector volume, the front bulkhead that the injector is attached to in Propulse's rocket has been designed with a diffuser just upstream of the injector. A diffuser is simply a diverging section so that the cross-sectional area of the flow increases. This lowers the flow velocity coming from the feedline before it reaches the injector, allowing potential pressure drops as the oxidizer flows through the feed system to be somewhat regained. The significance of feed line pressure loss likely varies from system to system, and in some cases assuming zero losses may cause the flow rate to be over-predicted as P_1 is actually lower than the tank pressure. A potential remedy to this could be to simply choose a slightly lower initial P_1 than the expected tank pressure in the calculations.

Both the assumption of a linear upstream pressure drop and the more advanced model that will be shown at the end of the chapter are variations on what is known as an equilibrium model for the tank dynamics. What this means is that the propellant in the tank is assumed to be in phase equilibrium. Thus, the nitrous oxide is in saturated conditions at any given time. This is physically equivalent to assuming that the flow out of the tank is slow when compared to the heat and mass transfer between phases [42]. As the tank loses liquid and the tank pressure falls as a result, some of the remaining liquid in the tank "instantly" evaporates to get back to saturation. However, due to heat loss, the new point of saturation is now at a slightly lower pressure and temperature. Recalling the phase diagram, the nitrous oxide in the tank will follow the saturation line towards lower pressures and temperatures as the tank is emptied. It also assumed that the only interface of phase change is on the liquid surface inside the tank, meaning that any N_2O arriving at the injector is as a saturated liquid. To make the assumption of "instant" evaporation more valid, a large tank diameter can be useful as this would increase the area available to heat and mass transfer. Although it appears that this is a plethora of assumptions, similar models have produced decent results in the past and it was deemed that this approach is the most practical for Propulse NTNU's purposes and capability [42].

The following models have all been programmed in Python 3.7 using the Spyder environment, with thermodynamic properties determined through the use of the CoolProp package. CoolProp includes highly accurate equations of state based on Helmholtz energy formulations, which makes determining thermodynamic properties much easier [43]. The code for each of the models can be viewed in Appendix A. The different models will be assessed with Propulse NTNU's 2020 rocket in mind. In

table 3.1, some of the design choices they have made that affect the injector design have been listed to provide the reader with some context.

Fuel/Oxidizer	Paraffin/N_2O
Tank pressurization method	N_2O Self-pressurization only
Initial P_1	6MPa
Operating P_2	3MPa
Desired $\dot{m}_{ox}$	2.6 $\frac{kg}{s}$
Desired $\dot{m}_{fuel}$	0.52 $\frac{kg}{s}$
Desired regression rate	0.004 m/s

Table 3.1: Some of the most important design choices affecting injector design

3.1 Single-Phase Models

For many different applications and designs of both liquid and hybrid rocket engines, injectors are typically operated with the fluid flow in a single-phase regime as either purely liquid or exclusively gaseous flow. The models for these kinds of flows are relatively simple and can be quite accurate, making them very useful for certain cases. They will also serve as a baseline when moving towards the two-phase models that are likely to be necessary for nitrous oxide considerations.

3.1.1 Single-Phase Incompressible Model

The simplest model used to analyze liquid flow through an injector orifice is the single-phase incompressible model. Typical rocket propellants such as liquid oxygen, liquid hydrogen, and kerosene can often be modeled to an adequate degree of precision using the SPI model, making it a useful tool [12].

The analysis begins from the continuity equation, assuming steady-state flow:

$$\dot{m} = \rho_1 u_1 A_1 = \rho_2 u_2 A_2 \tag{3.1}$$

A is the cross-sectional area and u is the velocity of the liquid. As the flow is assumed incompressible, the densities are related by $\rho_1 = \rho_2 = \rho$ and equation 3.1 can be rewritten as:

$$u_1 = u_2 \frac{A_2}{A_1} \tag{3.2}$$

The next step is to use the well-known steady Bernoulli equation, shown in equation 3.3.

$$P_1 + \frac{1}{2}\rho u_1^2 + \rho g H_1 = P_2 + \frac{1}{2}\rho u_2^2 + \rho g H_2 \tag{3.3}$$

P is the pressure, g is the standard gravitational acceleration, and H is the height relative to some reference. Neglecting the height difference upstream and at the exit of the injector, the Bernoulli equation can be further simplified to:

$$P_1 + \frac{1}{2}\rho u_1^2 = P_2 + \frac{1}{2}\rho u_2^2 \tag{3.4}$$

By inserting equation 3.2 into 3.4 and rearranging the terms a useful expression for the exit velocity is obtained, as shown in equation 3.5.

$$u_2 = \sqrt{\frac{2(P_1 - P_2)}{\rho[1 - (\frac{A_2}{A_1})^2]}} \tag{3.5}$$

This result can then be used to express the theoretical mass flow rate through the injector:

$$\dot{m}_{SPI} = \rho A_2 u_2 = A_2 \sqrt{\frac{2\rho(P_1 - P_2)}{[1 - (\frac{A_2}{A_1})^2]}} \tag{3.6}$$

Equation 3.6 is usually augmented by a discharge coefficient C_d. This is because the actual flow rate will be affected by frictional losses at the inlet and along the length of the injector hole, in addition to vena contracta effects. Vena contracta is a phenomenon that makes the effective exit area smaller than A_2 and is caused by flow separation. In addition, the denominator in equation 3.6 will often approach unity because the ratio $\frac{A_2}{A_1}$ tends to be very small. Therefore one can choose to incorporate this denominator into the discharge coefficient, resulting in the "C_dA" equation, where the pressure drop across the injector $P_1 - P_2$ is written as ΔP.

$$\dot{m}_{SPI} = C_d A_2 \sqrt{2\rho\Delta P} \tag{3.7}$$

The value of the discharge coefficient depends on many factors and is highly dependent on injector geometry. It is usually determined experimentally and is calculated by dividing the experimentally measured mass flow rate with the theoretical mass flow given by equation 3.7 if $C_d = 1$. Water testing is often used to give a reasonable estimate, and the value of C_d usually ranges from 0.6 to 0.9 for straight hole injector orifices [12]. In Rocket Propulsion Elements, Sutton [6] recommends a value of 0.65 for sharp-edged, straight holed orifices. Chamfering or rounding the inlets can increase C_d, while using angled orifices may lower it [12].

The code for the SPI model can be seen in Appendix A. For ease of reading, a flowchart showing the basic functions of the code is provided in figure 3.3. The flowcharts for each of the models are provided to support the reader's understanding of the model, but may still require some programming knowledge to be understood properly. The input parameters for this model are:

- Fluid (i.e. 'N2O')
- Discharge coefficient C_d
- Injector orifice diameter D_2
- A vector containing upstream pressures P_1
- Operating P_2
- Number of iteration steps for mass flow rate calculation

Note that here, "vector" means a one-dimensional array in Python that contains scalars. The upstream pressure input is taken as a vector so that cases of different

P_1 can be modeled. This is also used to apply the assumption of a linear upstream pressure drop, by making this vector range linearly from predesignated initial and final values. The operating P_2 input is used to assess the flow rate at the operating conditions of the burn chamber, but do keep in mind that for each P_1 in the vector, the mass flow rate will be calculated for all $0 < P_2 < P_1$. The number of iteration steps determines how many different P_2 will be used in this calculation. Once the flow rate has been calculated for each P_2, they are plotted against the pressure difference for the current P_1 case. Only then is the flow rate at the operating P_2 extracted from the result. This procedure is repeated for all the P_1 values in the input vector, and then the operating flow rates that have been extracted for each case are plotted against P_1. The code also returns the average flow rate of this plot. This general procedure will be used for most of the models throughout this thesis.

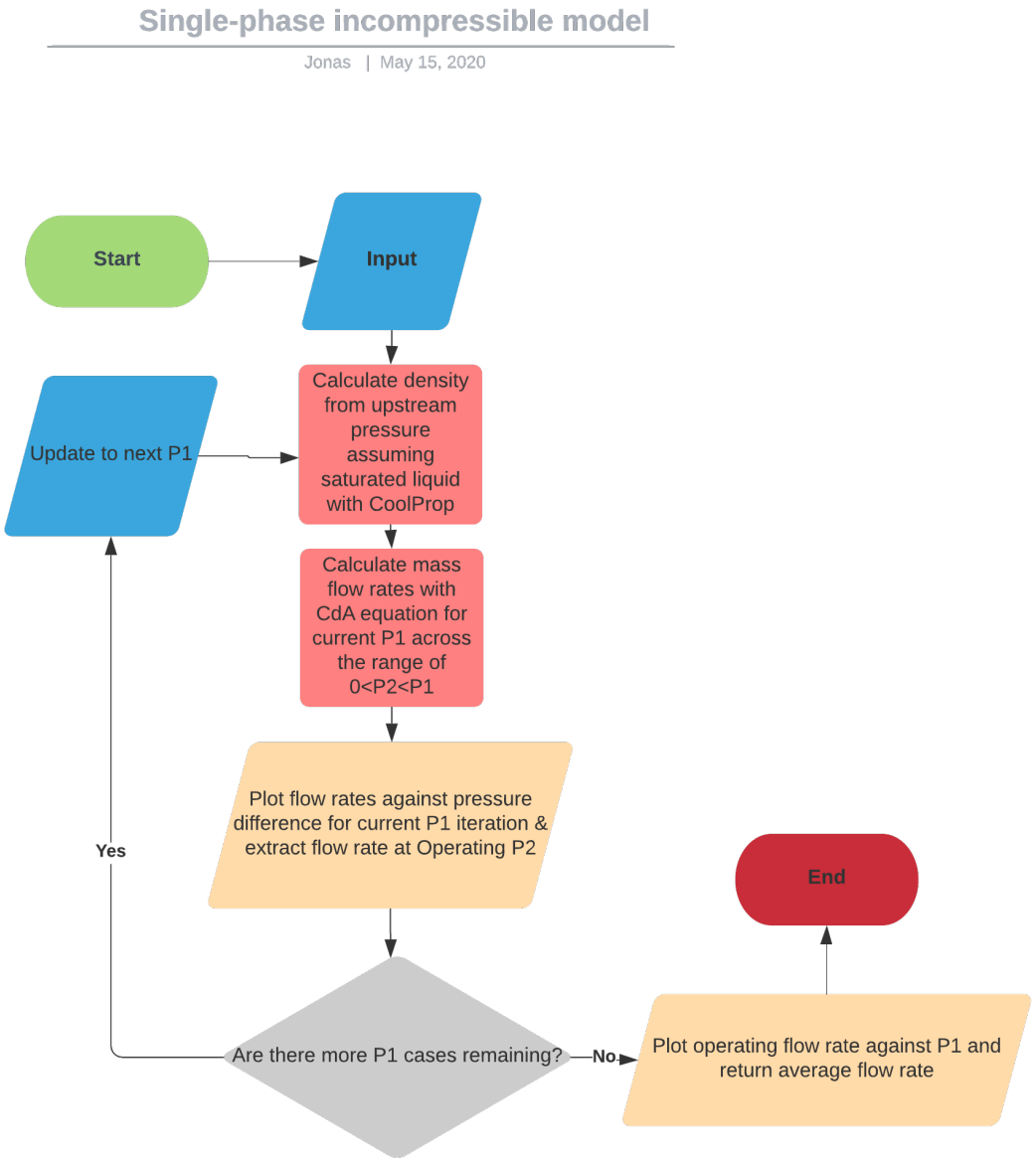
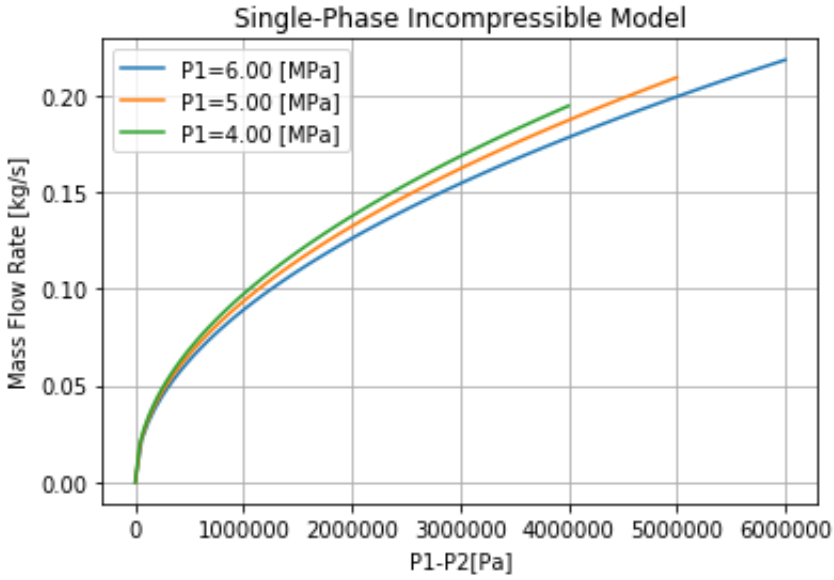
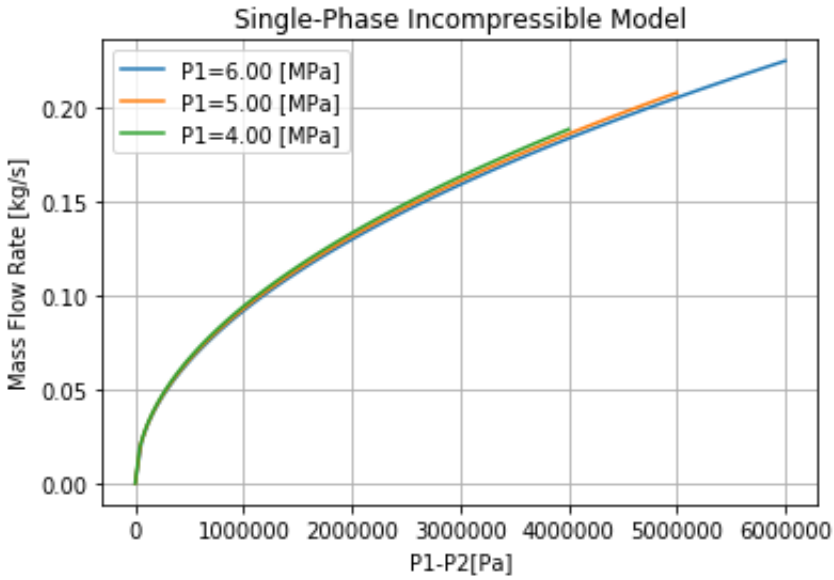


Figure 3.3: Flowchart for the SPI model



(a) SPI model with liquid N_2O.



(b) SPI model with liquid H_2O.

Figure 3.4: SPI model results. $C_d = 0.75$, D_2=2mm, 100 iteration steps

Figure 3.4 shows a few example calculations using the SPI model for a variety of downstream pressures with constant upstream pressure, plotting the mass flow rate against the pressure difference. One thing to note is that despite this being an incompressible model, the different P_1 cases show slightly different mass flow rates for equal ΔP. This is because of the density, which is calculated using the upstream pressure assuming saturated liquid with CoolProp. Thus, different P_1 will produce slightly different densities. Only then is the incompressible assumption used, such that the upstream and downstream densities are set to be equal. This variation is much less apparent in the example using water, indicating that nitrous oxide density is more sensitive to pressure changes, which will be touched upon at the end of this

section. All the different graphs exhibit similar behavior. An initial sharp spike in flow rate as a pressure drop is established, but further increases do not raise the flow rate as quickly. At the same time, the flow rate does not show any sign of choking and will continue to increase as P_2 gets smaller. This is consistent with the fact that choking is a property of gaseous flows. If the downstream pressure changes or oscillates, the flow rate will not remain constant even for constant upstream pressure. However, if the downstream pressure oscillations are small, the flow rate could be reasonably stable.

In figure 3.5 the mass flow rate of N_2O at the operating P_2 is plotted against different P_1 that vary linearly. Note that the vector containing P_1 goes from 6 to 4 MPa in 100 steps for this calculation rather than the three cases shown in figure 3.4, for increased resolution. This plot highlights how upstream pressure changes will result in decreasing mass flow rates, as the pressure difference changes. The decrease is significant in this case, with a fall of approximately 35%. With such a great variation, using the average flow rate and assuming that the oxidizer flow is constant for thrust calculations could result in significant deviations from the actual performance. As such, Propulse may need to reconsider their design or work around this in some other manner.

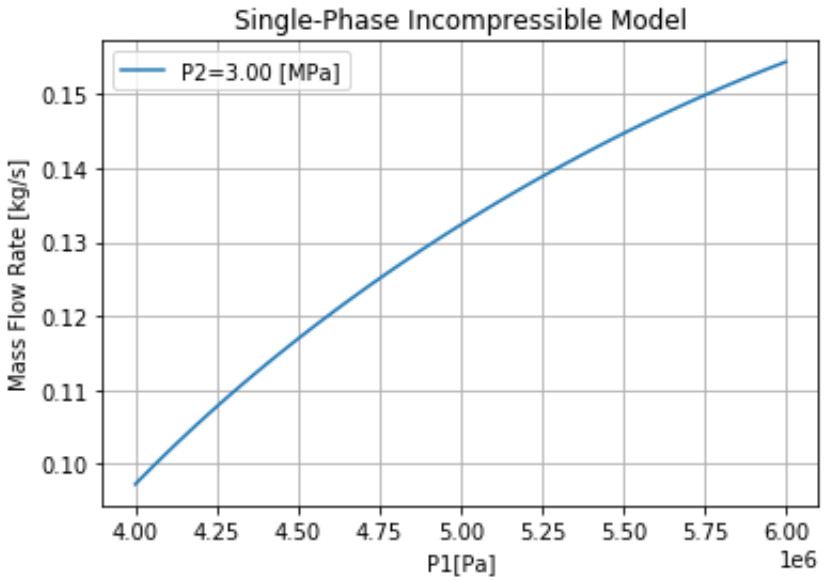


Figure 3.5: SPI model with linear upstream pressure drop with N_2O. $\dot{m}_{SPI,avg.} = 0.13kg/s$

Compressible liquid

While equation 3.7 is often used to model traditional liquid propellants, some errors can occur due to the effects of compressibility, especially when considering the injection of high-vapor pressure oxidizers. This is particularly important close to the critical point of the fluid, and therefore equation 3.7 can be modified with a compressibility correction factor Y:

$$\dot{m} = C_d A_2 Y \sqrt{2\rho\Delta P} \tag{3.8}$$

To determine the compressibility factor for compressible liquid, one can follow the approach by Zimmerman et al. [18]. Here, however, only the results of figure 3.6 [12] will be discussed briefly. Figure 3.6 shows that the compressibility factor becomes

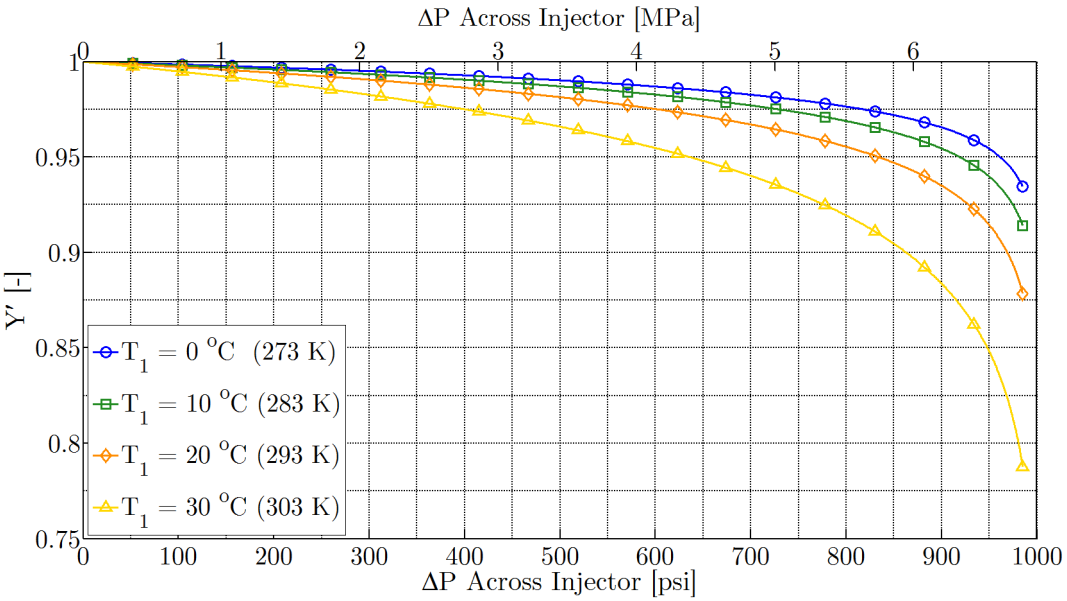


Figure 3.6: Compressible liquid correction factor of N_2O plotted against ΔP for different temperatures with $P_1 = 6.89 MP_a$. The critical temperature of N_2O is 309 Kelvin [12].

more important as the pressure drop increases and the temperature approaches the critical value. Propulse intends to operate quite close to the critical point, so one would think that this is an important consideration. However, two-phase flow effects will typically make compressible liquid considerations insignificant for nitrous oxide and therefore the treatment of compressible liquids will be ended here [12].

3.1.2 Perfect Gas Model

Single-phase gaseous flow will be assessed using the perfect gas model, which is very commonly used to predict the flow of gases, such as gaseous oxygen, through injectors. While the assessment of N_2O in section 1.4 indicates that gas-only flow of this oxidizer is generally unwanted, the perfect gas model has been included here as it might prove useful for Propulse in the future. Additionally, this model can be useful to gain a better understanding of N_2O behavior, can illustrate the choked flow concept, and serves as a natural bridge before moving on to the two-phase models.

In this model it is assumed that the gas is thermally and calorically perfect, allowing for the use of the ideal gas law:

$$P = \rho RT \tag{3.9}$$

This assumption also dictates that the heat capacities C_p and C_v are constant, resulting in simple relations for the specific enthalpy h and the specific energy e:

$$h = C_p T \tag{3.10}$$

$$e = C_v T \tag{3.11}$$

Equation 3.12 can be obtained from the first and second laws of thermodynamics, and is a form of the famous "Tds equations".

$$Tds = dh - \frac{dp}{\rho} \tag{3.12}$$

If the flow then is assumed to be isentropic and the ideal gas law is utilized, integration of equation 3.12 can provide the power-law relations for an isentropic perfect gas[44]:

$$\frac{P_2}{P_1} = \left(\frac{\rho_2}{\rho_1}\right)^{\gamma} = \left(\frac{T_2}{T_1}\right)^{\frac{\gamma}{\gamma-1}} \tag{3.13}$$

Here, γ is the heat capacity ratio:

$$\gamma = \frac{C_p}{C_v} \tag{3.14}$$

It also assumed that the flow is stationary upstream of the injector, and wall friction is neglected. The latter assumption will be accounted for later in the analysis by introducing a discharge coefficient.

Then, both continuity and the energy equation for calorically perfect gases are applied:

$$\dot{m} = const. = \rho_2 u_2 A_2 \tag{3.15}$$

$$C_p T_1 = C_p T_2 + \frac{1}{2} u_2^2 \tag{3.16}$$

Rearranging the terms in equation 3.16, an expression for u_2 can be obtained,

$$u_2 = \sqrt{2C_pT_1\left(1 - \frac{T_2}{T_1}\right)} \tag{3.17}$$

which is then inserted into the continuity equation. Then, by using the ideal gas law and the isentropic power pressure-density relation, the mass flow rate can be written as:

$$\dot{m_{PG}} = C_d A_2 \rho_1 \sqrt{2C_pT_1\left[\left(\frac{P_2}{P_1}\right)^{\frac{2}{\gamma}} - \left(\frac{P_2}{P_1}\right)^{\frac{\gamma+1}{\gamma}}\right]} \tag{3.18}$$

This equation is very useful because it models the mass flow based only on the upstream thermodynamic conditions and the ratio of the downstream to upstream pressure. C_d has been added here to account for frictional losses, as mentioned earlier [12].

Equation 3.18 has a maximum at what is called the critical pressure ratio,

$$\left(\frac{P_2}{P_1}\right)_{crit} = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{3.19}$$

and when the pressure ratio drops below this value $\dot{m}$ becomes independent of downstream conditions due to the flow becoming sonic. This chokes the flow, as was mentioned in section 1.4.

The code for this model can be found in Appendix A, and a flowchart outlining it is shown in figure 3.7. The input variables are the same as was shown for the SPI model, with one important addition. Now the upstream temperature is also an input variable, and it must be chosen so that the fluid is in the gaseous region of the phase diagram (or as saturated vapor) for all the P_1 in the vector. The number of iteration steps will now determine how many different pressure ratios the mass flow calculation will done be for in each P_1 case. Then the density and heat capacities are found with CoolProp, with a condition enforcing the gaseous state. With these variables, the critical pressure ratio and critical flow rate are calculated using equations 3.19 and 3.18.

Following that, the mass flow rate is calculated for each pressure ratio using equation 3.18. However, when the pressure ratio gets below the critical ratio, the mass flow rate is set to be equal to the critical flow rate to account for choked flow. Once the entire pressure ratio vector has been iterated through, the code outputs a plot of the mass flow rate against the pressure ratio. If any more P_1 cases are to be assessed, the process repeats until there are no remaining P_1 in the input vector. This allows for multiple cases of P_1 to be shown in the same plot, similar to what was done for the SPI model. However, this model was not made to assess a linear pressure decrease. Recalling figures 3.2 and 1.9, this assumption was only valid during the liquid expulsion. When the tank is empty of liquid N_2O, the pressure falls much more rapidly than before and as the gas-only flow is generally unwanted, it was deemed unnecessary to model. If Propulse wants to use a gaseous oxidizer in the future, the tank dynamics should be reassessed with that in mind.

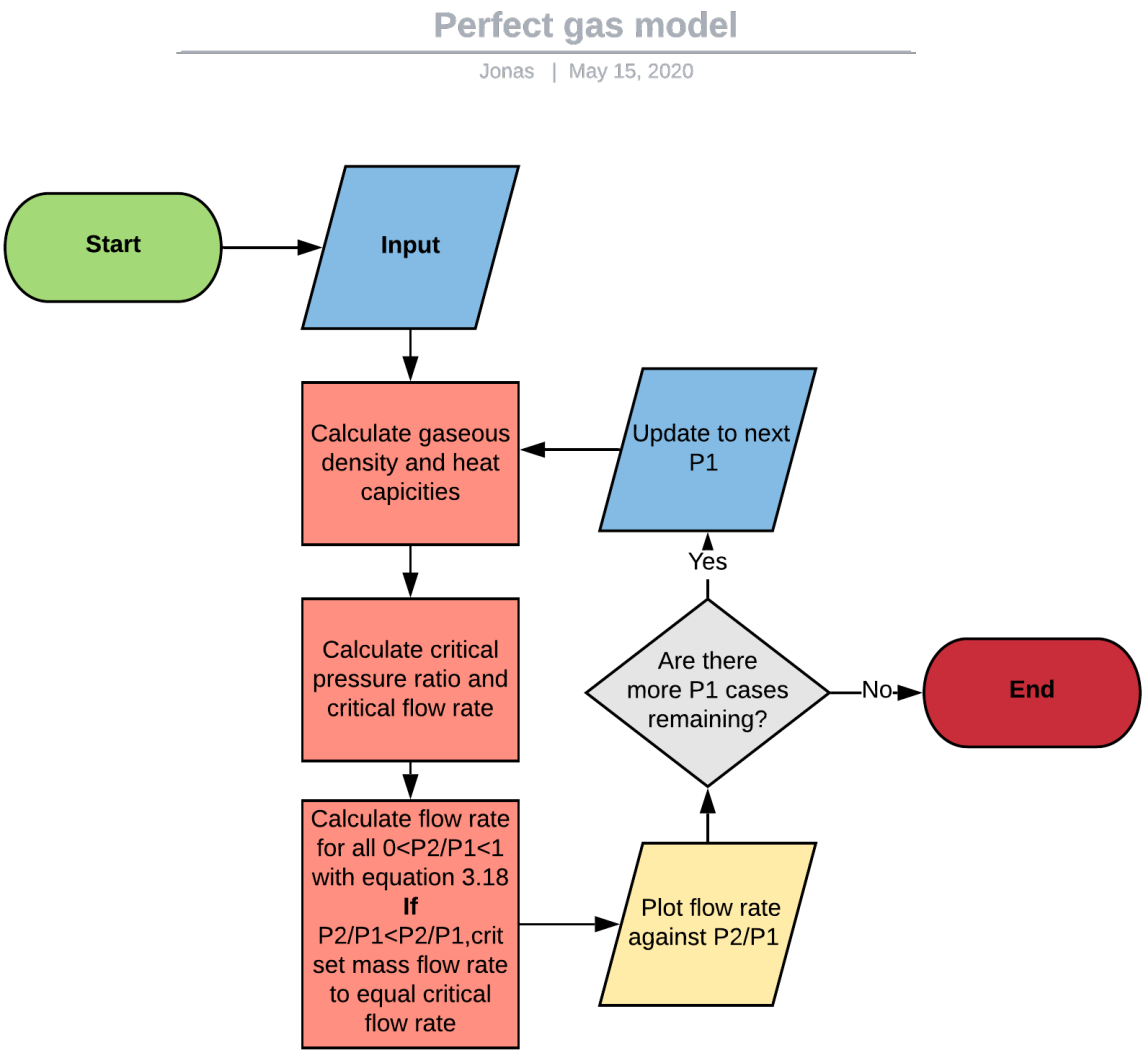


Figure 3.7: Flowchart for the perfect gas model

In figure 3.8 a few example cases with the perfect gas model are plotted. The dotted lines represent what the equation predicts for these small pressure ratios, but as explained, choked flow occurs at the maximum for each graph. Therefore, the continuous line is made to stay at its maximum to reflect the physical flow. One can observe that higher upstream pressures produce higher mass flow rates for the same pressure ratio. This makes sense, as the ideal gas law tells us that the density will be significantly larger. Additionally, for an identical pressure ratio, ΔP will be larger if P_1 is larger. While equation 3.18 uses the pressure ratio, one can intuitively infer that a greater pressure difference would cause a greater "force" that drives the flow.

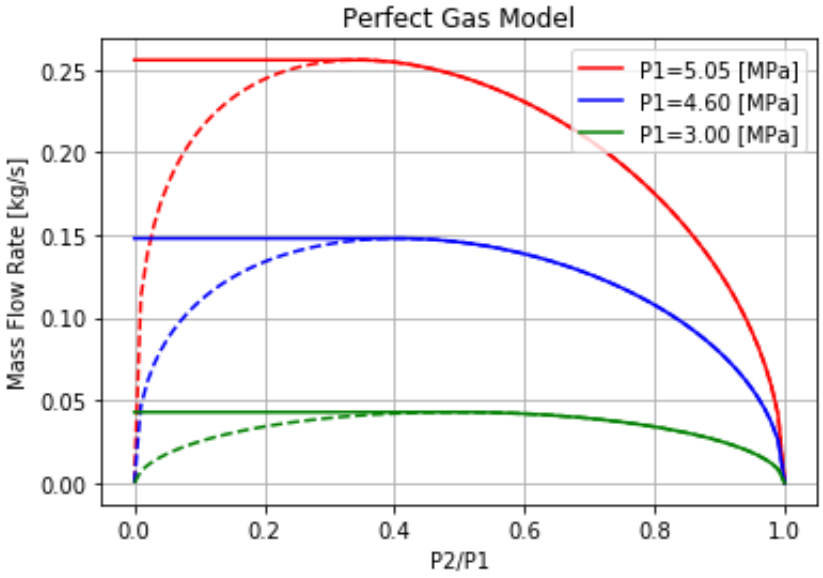


Figure 3.8: Mass flow rate calculations for gaseous N_2O using the perfect gas model. $D_2 = 2mm$, C_d=0.75, T_1=293K

3.2 Two-Phase Models

As was discussed in section 1.4.2, nitrous oxide will likely develop two-phase flow as it goes through the injector. As such, the single-phase models that have been shown so far are not likely to accurately predict the mass flow rate, and two-phase models must be developed. Two-phase models can generally be broken down into models that assume thermodynamic equilibrium between the two phases as the flow expands, and those who do not. Thermodynamic equilibrium implies that the temperatures of the two phases are equal and that pressure and temperature can be related by the thermodynamic saturation curve inside the injector. The models can then further be broken down into homogeneous or non-homogeneous models, where the former assumes that the liquid and gaseous phase have the same velocity [12]. There are many different two-phase mass flow rate models but for this thesis, only two have been selected. One is the simplest of the two-phase models, while the other is the most popular due to its decent performance when compared to most other models.

3.2.1 Homogeneous Equilibrium Model

The first two-phase model that will be assessed is the simplest of them - the homogeneous equilibrium model. In this model, it is again assumed that the orifice cross-sectional area A_2 is much smaller than A_1, which is valid for most hybrid rocket engine injectors. The flow is also assumed isentropic. The analysis begins from equations 3.20 and 3.21, the continuity and energy equations. Note that the fluid can contain a mixture of liquid and vapor at location 2. The vapor mass fraction is defined in equation 3.22, where the subscripts v and l denote vapor and liquid states [12].

$$\dot{m} = const. = \rho_2 u_2 A_2 \tag{3.20}$$

$$h_1 = h_2 + \frac{1}{2}u_2^2 \tag{3.21}$$

$$x_2 \equiv \frac{\dot{m}_v}{\dot{m}_l + \dot{m}_v} \tag{3.22}$$

After rearranging the energy equation to isolate u_2 and inserting this into the continuity equation, an expression for the mass flow rate can be obtained. A discharge coefficient is then added, resulting in equation 3.23.

$$\dot{m}_{HEM} = C_d \rho_2 A_2 \sqrt{2(h_1 - h_2)} \tag{3.23}$$

To find the downstream thermodynamic properties of the liquid/vapor mixture, the isentropic assumption is used alongside the downstream pressure with the CoolProp package. Therefore, as long as the upstream conditions and the chamber pressure are known, the mass flow rate can be calculated.

In figure 3.9 a flowchart provides the basic outline of the code for the homogenous equilibrium model. Note that the input variables used here are the same as for the SPI model, so they will not be detailed here.

CoolProp is used to find the upstream enthalpy h_1 and entropy s_1. The downstream entropy then takes the same value as s_1 from the isentropic assumption. Then the code goes into its first for loop, which updates P_2 and iterates for the specified number of iteration steps. The density and enthalpy at the injector exit of each iteration can be found from s_2 and the downstream pressure of the current iteration. When the density and enthalpy are known, equation 3.23 is used to calculate the mass flow rate, which is stored in a vector containing the mass flow rate of each iteration. This continues for all P_2 in this P_1 case. One important thing to note is that now, the lowest P_2 is not 0 as it was in the SPI model. This is because CoolProp runs into an issue with the isentropic assumption if the downstream pressure goes lower than the triple point pressure, likely due to the solid phase becoming relevant at this point. Therefore, P_2 is set to range from slightly above the triple point pressure P_{trip} to P_1 instead. This does not change much for nitrous oxide, which has a triple point pressure below $1atm$, but carbon dioxide has $P_{trip} = 5.2atm$. Therefore, the flow rate while venting to atmospheric conditions can not be modeled. This could potentially be troublesome for cold flow testing, although choked flow may make it not matter. This problem will be elaborated upon in chapter 4.

Once all the mass flow rates for the different P_2 in the current P_1 case have been calculated, the maximum flow rate is determined. Choked flow is then imposed by mandating that the flow rate must be equal to the maximum for iterations where P_2 is lower than the critical value. Just like in the perfect gas model, choked flow is not directly predicted by the HEM and therefore must be imposed. When this is done, the flow rate is plotted against the downstream pressure. After the flow rate at the operating P_2 is extracted, P_1 is updated and the process repeats.

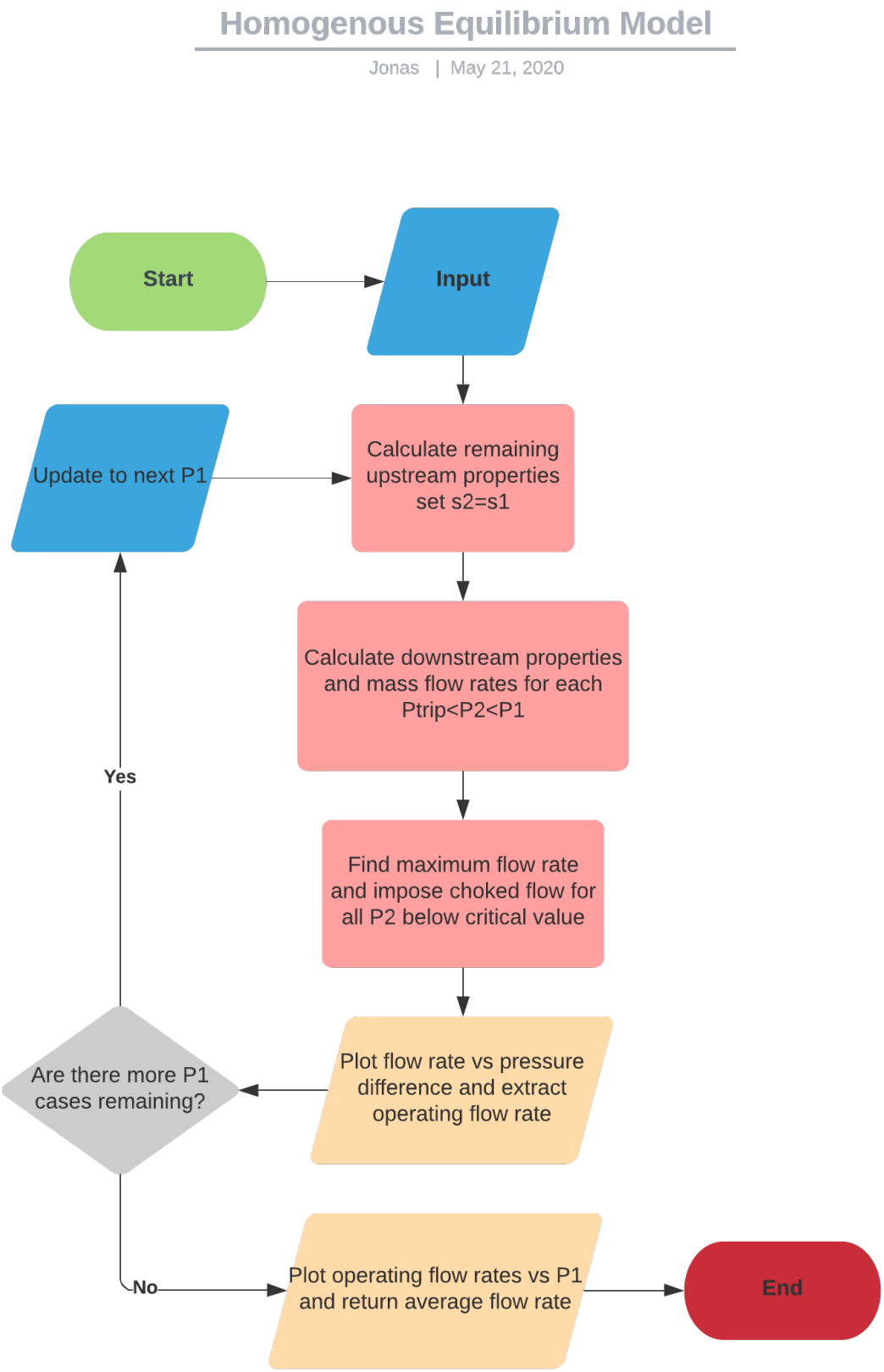


Figure 3.9: Flowchart for HEM

The imposing of choked flow can be seen in figure 3.10, as the model will not predict choking but follow the dotted line. An interesting point to note is that for Propulse NTNU's desired operating pressures of 6 MPa upstream and 3 MPa downstream, the model predicts that critical flow will be reached. When comparing these results to the ones from the SPI model in figure 3.4, which used the exact same input values, some trends of HEM can be observed. From $\Delta P = 0$ to about 1 MPa, HEM appears to predict flow rates relatively close to what the SPI model predicts for this case. This is to be expected, as for high P_2 the liquid phase should dominate. However, as the pressure difference continues to increase, the HEM curve begins to flatten out. SPI also sees a decrease in slope, but it is not as significant as for HEM. This leads to the transition region from 1-2 MPa pressure difference, where the two models deviate further from each other, but $\dot{m}_{HEM}$ has not yet choked. For the HEM model, some vapor has now formed. The bulk density decrease is enough to cause deviation from the SPI model, but not enough vapor has formed to choke the flow. When P_2 falls below the critical value, the model predicts that a significant amount of vapor is exiting the injector, enough for the flow to choke. The model predicts choked flow to occur for slightly lower P_2 than the criterion that was shown in equation 1.10, but is quite close to agreeing with it.

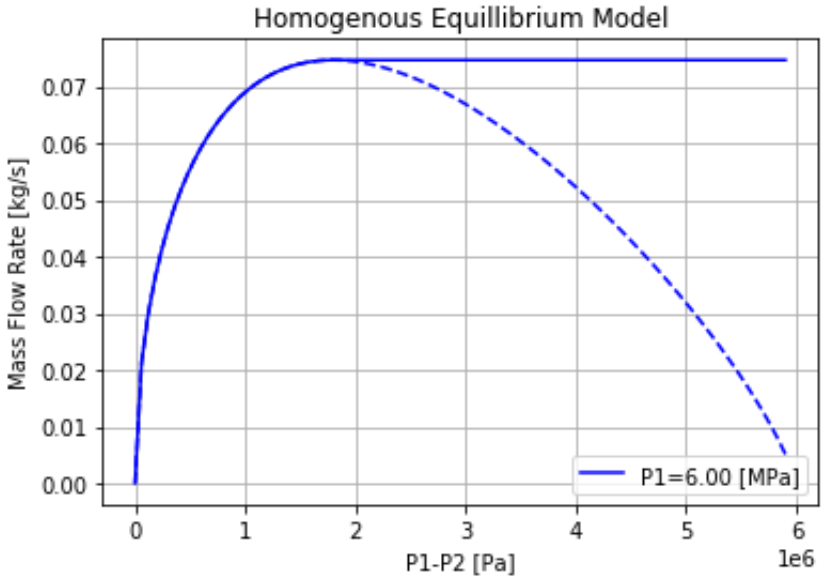


Figure 3.10: Example calculation for $\dot{m}_{HEM}$ of N_2O, plotted against ΔP. $C_d = 0.75$, $D_2 = 2mm$, 100 iteration steps.

Once all the upstream pressures in the P_1 vector have been iterated through, the model outputs the average flow rate and plots both the operating flow rate and the choked flow rate against the upstream pressure. In figure 3.11, an example calculation from HEM with this tank model is shown. It is clear that when the tank dynamics are taken into account the mass flow rate does not remain constant at all, but shows a decrease of around 20%. The rocket engine designers must be careful to take this into account. One thing to note is that the blue and orange graphs overlap, indicating that the model predicts the flow to remain choked for all of the relevant P_1 when the downstream pressure is 3MPa. It also highlights that when the flow stays choked, the mass flow rate decreases relatively linearly with the upstream pressure. This implies that while Propulse will not be able to maintain a constant

mass flow rate, it may be possible to operate at choked conditions for the duration of the burn. This could be advantageous to improve combustion stability. While

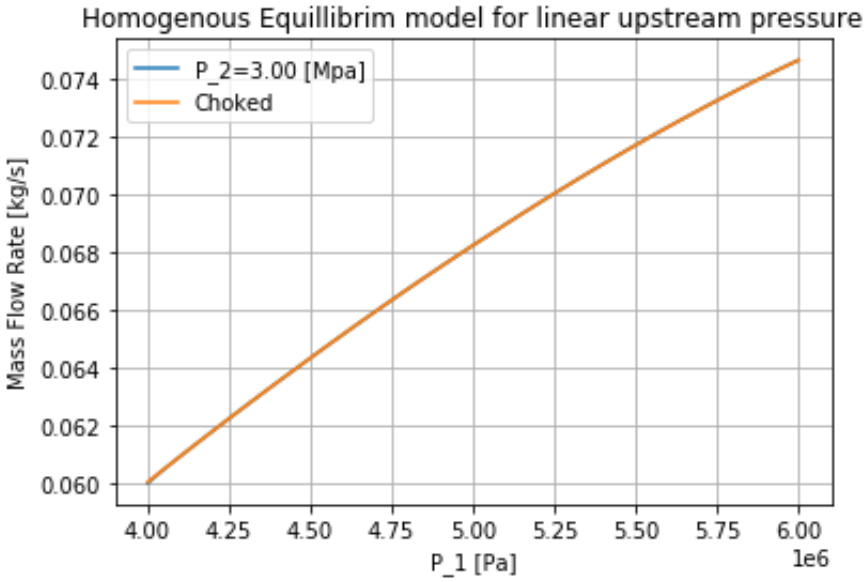


Figure 3.11: Average $\dot{m}_{HEM} = 0.068 \frac{kg}{s}$

HEM is relatively easy to implement and can provide adequate results at times, the assumption of thermodynamic equilibrium is not necessarily always valid. The homogeneous equilibrium model will only provide a lower-bound estimate for the critical flow rate, and will in many cases produce a significant under-prediction when compared to experimental data. This is likely due to non-equilibrium effects, which the final flow rate model presented in this work will incorporate [12].

3.2.2 The Dyer Model

The Dyer model was developed by Dyer, et al.[45] and then corrected by Salomon [46]. It is a non-homogeneous, non-equilibrium model, and does this by allowing the mass flow to smoothly transition between values predicted by the HEM and SPI models. It is one of the most popular models used to assess two-phase nitrous oxide flows and has been shown to predict flow rates better than many other alternatives [12].

Dyer postulated that the non-equilibrium effects were largely due to superheating of the liquid during expansion and finite vapor bubble growth rates [12]. Therefore, Dyer defined a characteristic bubble growth time,

$$\tau_b = \sqrt{\frac{3}{2}\frac{\rho_l}{P_v - P_2}} \tag{3.24}$$

where P_v is the vapor pressure and ρ_l is the liquid density.

The amount of vapor that can form within an injector is dependent on the ratio of this bubble growth time to the fluid residence time, a measure of how long the fluid stays in the injector. The fluid residence time can be defined as in equation 3.25:

$$\tau_r = \frac{L}{u} = \frac{L}{\sqrt{\frac{2\Delta P}{\rho_l}}} = L\sqrt{\frac{\rho_l}{2\Delta P}} \tag{3.25}$$

By comparing these two characteristic times, Dyer introduced a non-equilibrium parameter k, shown in equation 3.26.

$$k = \sqrt{\frac{P_1 - P_2}{P_v - P_2}} \propto \frac{\tau_b}{\tau_r} \tag{3.26}$$

This then leads to a weighted expression for the mass flow rate, given in equation 3.27:

$$\dot{m}_{Dyer} = \frac{k}{1+k}\dot{m}_{SPI} + \frac{1}{1+k}\dot{m}_{HEM} \tag{3.27}$$

The equation has been set up such that if the bubble growth time is much larger than the residence time, very little vapor will be formed and so the single-phase assumption is weighted more heavily. If it is the residence time that is larger, the fluid would have time to reach equilibrium. Then the flow rate should lean towards the homogeneous equilibrium model and approach the value predicted there [12]. This suggests that the reason why high L/D ratio injectors develop choked flow more easily could be due to a high fluid residence time.

Note that in the work done by Salomon, C_d and A are included outside the brackets in the equation above [46], and in the work done by Waxman et. al [12] A is included. However, it seems that these are most likely typos, as A should be included within the SPI and HEM mass flow rate expressions. Multiplying an area again here would cause the units to be wrong, although the discharge coefficient could potentially be placed here and set to 1 inside the HEM and SPI models.

In figure 3.12, a flowchart provides the basic outline of the code for the Dyer model. Once again, the input variables are identical to what was used for the HEM and SPI models, as the Dyer model is a combination of the two.

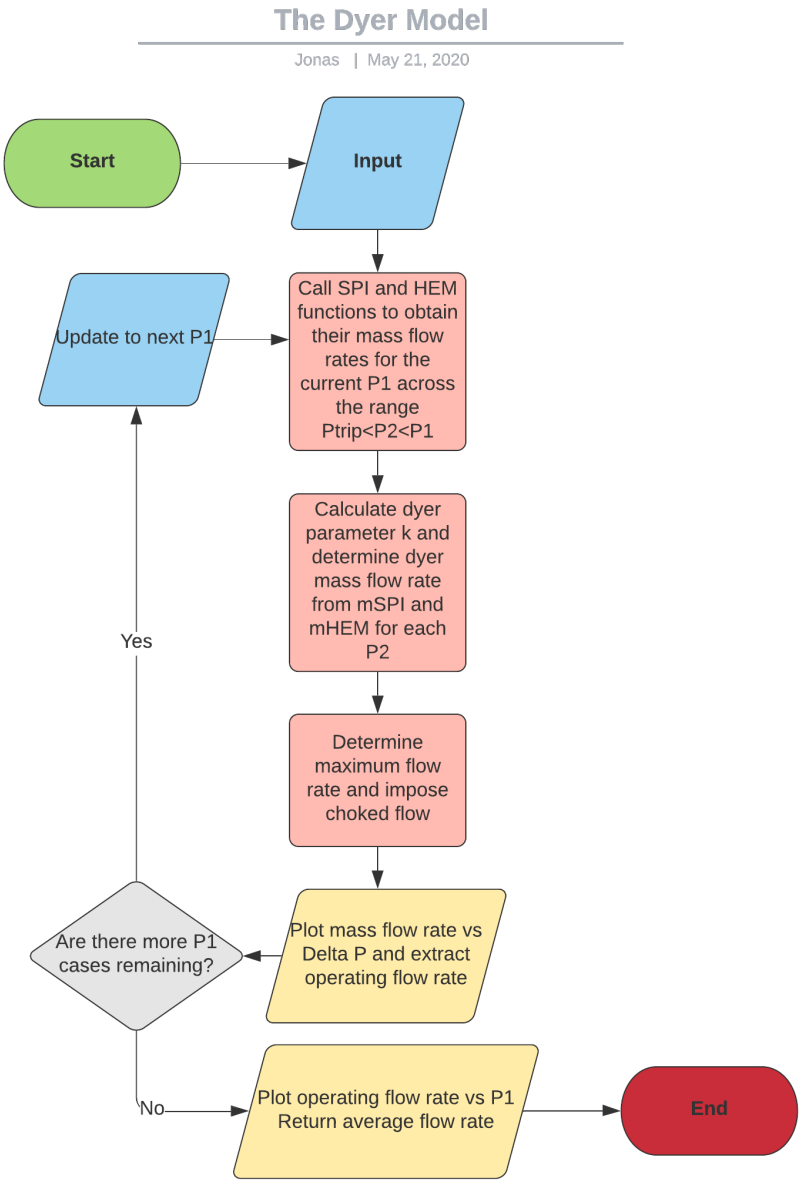


Figure 3.12: Flowchart for the Dyer model

As the code for SPI and HEM have already been made, the first operation in this model is to simply call the functions from SPI and HEM that return vectors containing SPI and HEM mass flow rates for a given P_1. Simplified versions of the SPI and HEM models that only assess one P_1 case at a time were made so that they could be used by the Dyer model more easily. They can be seen in Appendix A. The vapor pressure is then set to be equal to the upstream pressure, as only the self-pressurized saturated case is considered.

Following that, the Dyer parameter k is calculated with equation 3.26. Note that since the upstream pressure equals the saturated pressure, k will always be equal to 1. Then, the Dyer mass flow rate for this iteration is calculated using equation 3.27. The procedure is repeated for every P_2 for this case, and once all the iterations are complete, the maximum flow rate and the index in the mass flow rate vector where it is located is determined. Then, choked flow is accounted for by setting the flow rate to be equal to the maximum for any indexes below the critical index. Once this for loop is finished, the code outputs a plot of the Dyer mass flow rate against P_2 and the operating flow rate is extracted. The code then moves on to the next P_1 case and repeats the process until there are no more P_1 to be assessed. Finally, the operating flow rate is plotted against the upstream pressure and the average operating flow rate is returned.

The results obtained for this model can be seen in figure 3.13. It is clear that

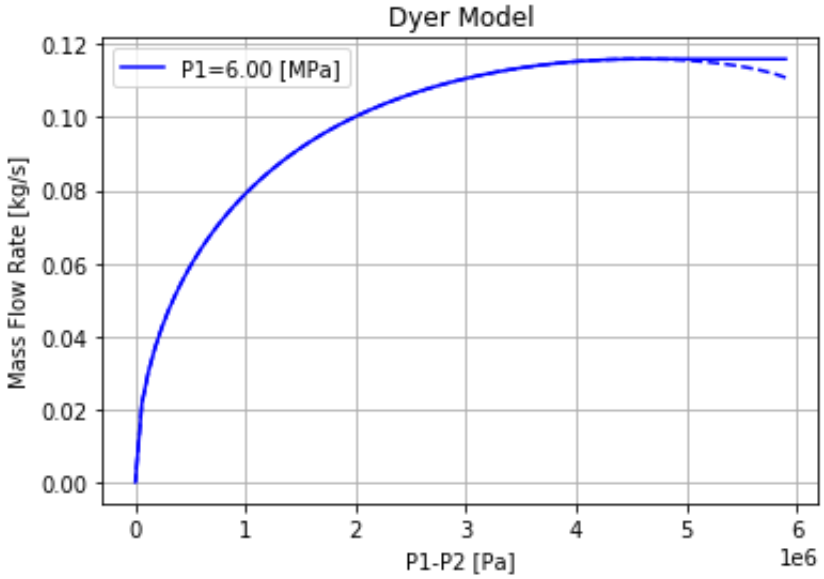


Figure 3.13: $\dot{m}_{Dyer}$ of N_2O plotted against ΔP. $C_d = 0.75$, D_2=2mm, 100 iteration steps

the Dyer model predicts significantly lower downstream pressures required for critical flow than the homogeneous equilibrium model did, thus also deviating from the $P_2 < 0.8P_v$ criterion of equation 1.10. The value of the critical flow rate is also higher when using the Dyer model versus HEM. For Propulse's desired operating conditions, the Dyer model does not predict choked flow in the calculation performed here. However, the Dyer model is similar to HEM in that it predicts flow rates close to the SPI model for large P_2, with the deviation becoming greater as the Dyer curve flattens when P_2 decreases.

In figure 3.14, the operating flow rate is plotted against P1, showing how the mass flow rate changes while the tank empties. In stark contrast to the HEM model, the Dyer model does not predict choked flow at any point during tank expulsion. If the Dyer model indeed is the most accurate, then this could entail additional combustion instabilities. Similarly to HEM, though, the choked flow rate does seem to fall relatively linearly when compared to the actual flow rate. The actual flow

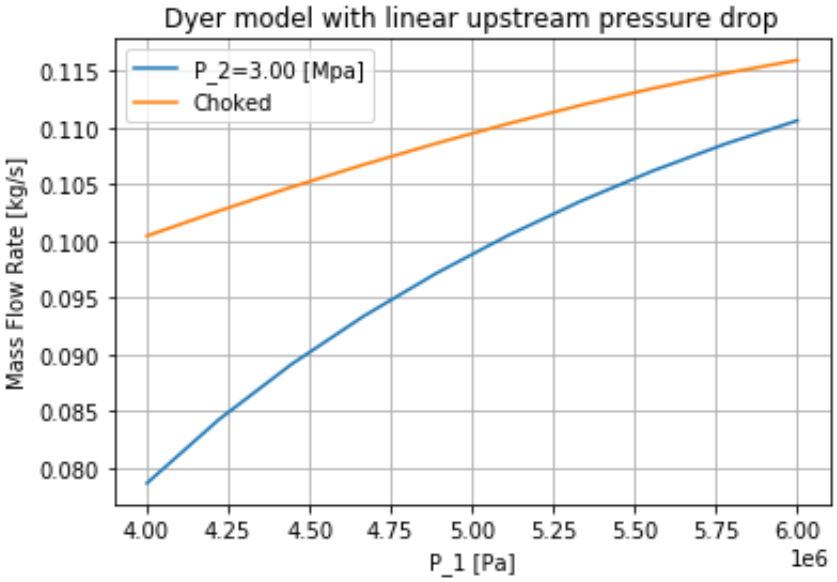


Figure 3.14: Average $\dot{m}_{Dyer} = 0.097 \frac{kg}{s}$

rate drops faster as the upstream pressure decreases, and from 6 to 4 MPa the mass flow rate drops almost 30%.

While experiments done by Waxman et. al [12] on supercharged nitrous oxide show that the Dyer model can usually give quite accurate results and is in contention for being the best current model, some data points still show a deviation of up to 15 percent when compared to experiments. Additionally, there is not full clarity as to which cases the Dyer model can be applied, particularly because of the k-parameter. If the upstream pressure is set to be the saturated liquid pressure, $P_1 = P_v$ and thus k will always be equal to 1. As a result, the Dyer model simply becomes the average of the SPI and HEM models. There are some examples where the Dyer model is used with saturated conditions that have produced seemingly reasonable results [42] [46]. However, they do not address this issue, so the saturated case remains somewhat unclear.

Another uncertainty about the use of this model stems from some calculations done by Waxman et al. [12]. It appears that $P_2 > P_v$ for some of Waxman's cases using supercharge, which should result in an imaginary value for k. It is unclear how they have gotten around this problem, but it is expected that the SPI model should be more heavily weighted in this case. These conditions are not expected for Propulse NTNU, so it should not be of great importance for the time being.

Despite these uncertainties, the Dyer model does seem to be the most accepted model for nitrous oxide two-phase flows. Therefore, when the alternative model for the tank dynamics is introduced in the next section, it has been programmed to use the Dyer model for the mass flow rate.

3.3 Transient Equilibrium Tank Dynamics

While assuming that the tank pressure follows a simple linear curve appears to agree decently with experimental data, it is still a relatively crude way of assessing the tank dynamics. In the linear model, it was necessary to estimate a final upstream pressure to create the linearly decreasing P_1 vector. If the final P_1 is guessed as a higher value than the actual pressure at the time of liquid run-out, P_{LRO}, the calculation will be made for a case where there is still liquid in the tank at the end. Depending on the over-prediction of the final P_1, it could lead to significant amounts of unspent oxidizer as the control valve is closed too early. On the other hand, if the final P_1 is under-predicted, it could lead to gas-only flow as the tank empties of liquid some time before the valve is set to close. This would also disrupt the calculations that assume that there is always saturated liquid upstream of the injector. Furthermore, the linear pressure assumption model has not been made to assess how the conditions develop in time directly. Therefore it does not predict how long the burn time would be. The model that will be presented here has been named "transient" as it will be able to track how the variables such as pressure, temperature, and mass flow rate develop with time, which should be very useful for rocket designers. Additionally, the final P_1 will now be calculated rather than guessed, avoiding the potential issues outlined earlier. It is worth noting that despite the name, this model will still not take the initial transient behaviors into account.

The transient equilibrium model that is presented here also assumes phase equilibrium in the tank, as the name implies. Therefore, it is still assumed that the nitrous oxide in the tank remains saturated at all times. The way that this model has been made stems from tracking the mass and internal energy of the nitrous oxide in the tank as it is being emptied. The equation for tracking the tank mass is simply related to the flow rate and can be seen in equation 3.28.

$$\frac{dm_{tot}}{dt} = -\dot{m} \tag{3.28}$$

For tracking the internal energy, it is assumed that there is no net work done on or by the oxidizer and that the tank is adiabatic so that no heat transfer between the tank walls and propellant is happening. Essentially, this is assuming that the tank is well isolated. This results in equation 3.29, which says that the change in internal energy of the propellant in the tank is equal to the enthalpy of the liquid that has left the tank:

$$\frac{dE_{tot}}{dt} = -\dot{m}h \tag{3.29}$$

The final equation that is needed is the volume constraint, where the volume of the tank is a known constant:

$$V_{tank} = m_{tot}\left[\frac{1-x}{\rho_l} + \frac{x}{\rho_v}\right] \tag{3.30}$$

Here, x is the vapor quality of the tank contents, while ρ_l and ρ_v are the liquid and vapor densities respectively. The expression for the vapor quality is given by equation 3.31.

$$x = \frac{\frac{E_{tot}}{m_{tot}} - e_l}{e_v - e_l} \tag{3.31}$$

What is important to note is that the specific internal energies e_l and e_v, as well as the densities in equation 3.30, can all be found using CoolProp as long one thermodynamic variable is known alongside the saturated liquid/vapor assumption. In other words, these variables can all be expressed in terms of T_1. Thus, if the volume, mass, and internal energy in the tank are known, it is possible to set up this volume constraint equation with T_1 as the only unknown and solve for the temperature.

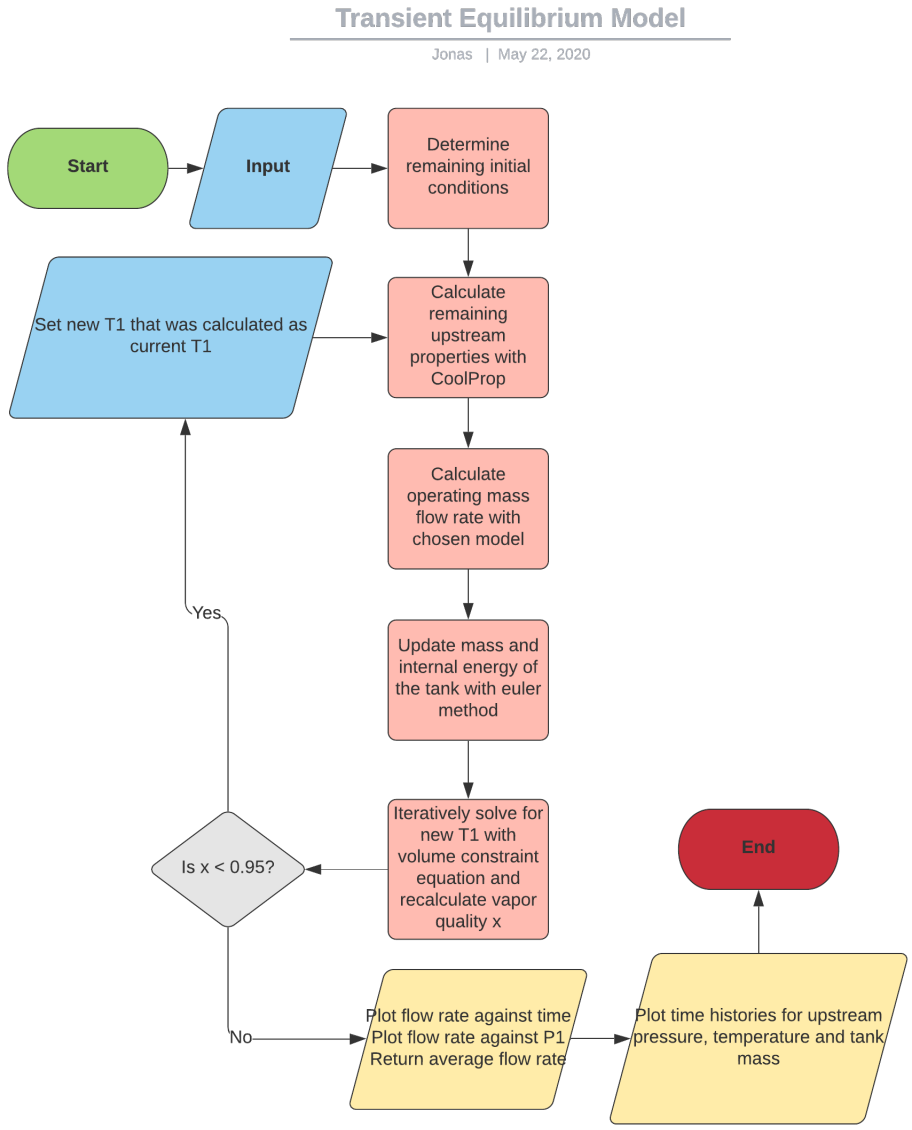


Figure 3.15: Flowchart for the Transient equilibrium model

A flowchart of the code is provided in figure 3.15, but due to the complexity of the model, it would be wise to study the code itself as well. The input variables can be seen in table 3.2. Many of the input variables that are used in this model have already been discussed for previous models, but there are now quite a few new ones introduced. The initial mass of the oxidizer and the tank volume should be known and are taken as inputs. The time-step that is introduced determines the resolution

of the solver and allows for the time histories of the different variables to be found. The number of orifices on the injector must now also be included, as it is the total mass flow rate from all the orifices combined that determines how quickly the tank empties. Finally, it can be observed that now the initial T_1 is taken as an input rather than P_1. This is a somewhat arbitrary choice, but it was thought to be more convenient to solve for temperature when using the volume constraint equation. If the designers have a certain initial tank pressure that is desired, it is a simple task to find the temperature that corresponds to this pressure at saturated conditions using CoolProp or the phase diagram. Also note that since T_1 is updated by calculation, inputting only the initial temperature rather than a vector is sufficient.

The model starts by performing some initial calculations. The density is determined by simply dividing the initial mass with the tank volume. The initial specific internal energy e of the gas-vapor mixture is found with CoolProp using the temperature and density. e is then multiplied by the mass to find the total internal energy E_{tot} of the propellant. The initial vapor quality x can now be calculated with equation 3.31. Additionally, an iteration counter "j" and a time tracker starting at 0 are initialized.

The code then moves into a while loop with a vapor quality condition, meaning that the iterations will continue as long the vapor quality is lower than some value. The value that has been chosen is 0.95, so it stops when the tank only contains 5% liquid. To avoid gas-only flow, 0.95 was chosen instead of 1 to have a small safety factor, but this could be changed if desired. The first operation inside the loop is to find the saturation pressure from T_1, and then record P_1, T_1, m_{tot}, and the time of this iteration. The mass flow rate at the operating P_2 is then found in the same way that was described for the previous models. It was decided to use the Dyer model for the mass flow rate with this tank dynamics model, but changing the code to use the SPI or HEM models instead could be done if desired.

Once the flow rate is determined, it is time to find the state of the tank for the next iteration. First, the total flow rate is calculated by multiplying the number of orifices by $\dot{m}_{Dyer}$. The mass is then recalculated from equation 3.28 by using the time-step input variable. Similarly, the internal energy is found with equation 3.29, with the enthalpy calculated using CoolProp. These have been solved with a basic Euler method as shown in the equations below:

$$m_{new} = m_{old} - \dot{m}\Delta t \tag{3.32}$$

$$E_{new} = E_{old} - \dot{m}h\Delta t \tag{3.33}$$

The volume constraint equation is then used to iteratively solve for the new T_1 corresponding to this combination of mass and internal energy using the previous T_1 as an initial guess. Once the new T_1 is found, the vapor quality is recalculated and the time-step is added to the total time.

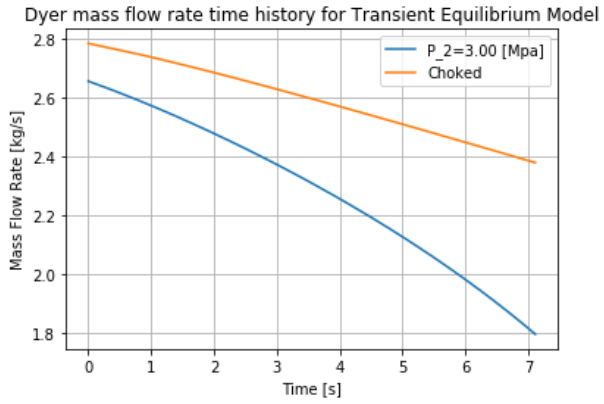
The while loop continues in this way until the vapor quality condition is met. At that point, the code outputs time history plots for $\dot{m}_{ox}$, P_1, T_1, and the tank

mass. The single-orifice flow rate is also plotted against P_1, so it can be more easily compared with other models. Finally, the average flow rates for both a single orifice and all orifices combined are outputted.

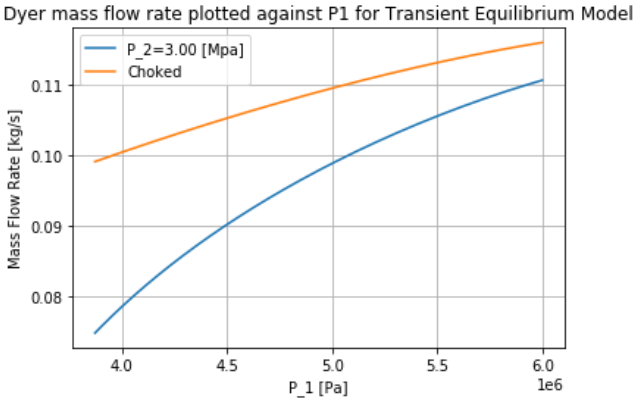
In the following figures, these plots can be seen for an example calculation. Note that the Dyer model was chosen as the mass flow rate model, but the model could potentially be adapted to use the other flow rate models as well. The input values that were used are displayed in table 3.2. The values were chosen based on Propulse's 2020 Hybrid rocket, with the initial tank temperature corresponding to a saturated vapor pressure of 6 MPa.

Fluid	N_2O
Cd	0.75
D_2	2 mm
Number of Orifices	24
V	$0.03255m^3$
Initial propellant mass	20 kg
T_1	300.86 K
Operating P_2	3 MPa
Iteration steps for flow rate calculation	100
Time-step	0.01 s

Table 3.2: Input values used in this example calculation



(a) Total mass flow rate plotted against time. Average flow rate of 2.284 kg/s.



(b) Flow rate (1 orifice) plotted against P_1. Average flow rate of 0.095 kg/s.

Figure 3.16: Expulsion time = 7.12 s. Final P_1 = 3.869 MPa.

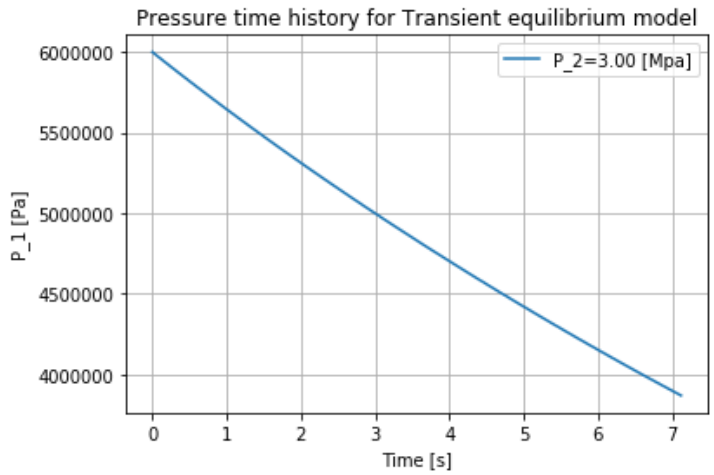
In figure 3.16a the mass flow rate has been plotted against the time, resulting in an average flow rate of 2.284 kg/s and a burn time of 7.12 seconds. The burn time is a result that is unique to this model and could be very useful when calculating the total impulse of the rocket, for example. The mass flow rate decreases around 30% during this time, similar to the decrease that was observed for the linear upstream pressure assumption. The choked flow rate falls quite linearly with time, but the flow does not choke for these conditions with the Dyer model. At first glance, figure

3.16b is very similar to 3.14, indicating that the linear pressure assumption gives similar results to the transient model. However, there are some differences. Most notably, the final P_1 is lower than what was guessed for the calculation in figure 3.14, even if the guess was quite good. Having a better estimate of the pressure at which the control valve should close would mean that less liquid propellant would remain unused in the tank. The average mass flow rate of the transient model is lower than for the linear model as a result. However, even when using the final pressure found from the transient model as input in the linear model, the transient model still predicts slightly lower flow rates. This is likely due to the slight non-linearity in the pressure time history shown in figure 3.17a, where the slope is larger at higher pressures. While the differences between 0.095 and 0.097 kg/s may seem very small, the impact of the deviation could become more significant as the number of orifices increases. If the guess used for the final P_1 in the linear model did not happen to be so close to the calculated value here, the results would have been even more differentiated.

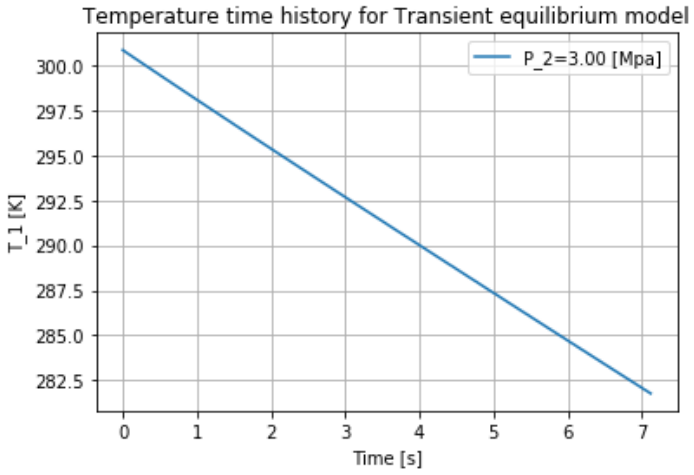
The mass time history is also deceptively linear at first glance. However, it does show some non-linearity with the slope decreasing with time. This is expected, as the mass flow rate plot clearly shows that it should decrease. In this case, the vapor quality reached 0.95 when there was still 3.74 kg left in the tank, meaning that close to 0.2 kg of liquid is unused. Considering the delay in the actuators that close the control valve, it seems that using the $x < 0.95$ condition should be a viable way to try to avoid gas-only flow while utilizing as much propellant as possible. Producing mass time histories in this manner can be very useful for making decisions about how much propellant should be filled into the tank. It could also be useful for flight calculations that must use the current mass of the rocket.

The temperature time history falls linearly, as expected. The final temperature can be important to assess as one would like to avoid freezing temperatures. It does indeed appear that the nitrous oxide triple point temperature of 182.33 K is very unlikely to be reached for reasonable operating conditions, as this would require around a 100 degree fall in temperature as the oxidizer flows through the injector. However, temperatures below the freezing point of water could potentially be problematic due to air moisture, especially during cold flow testing to ambient air.

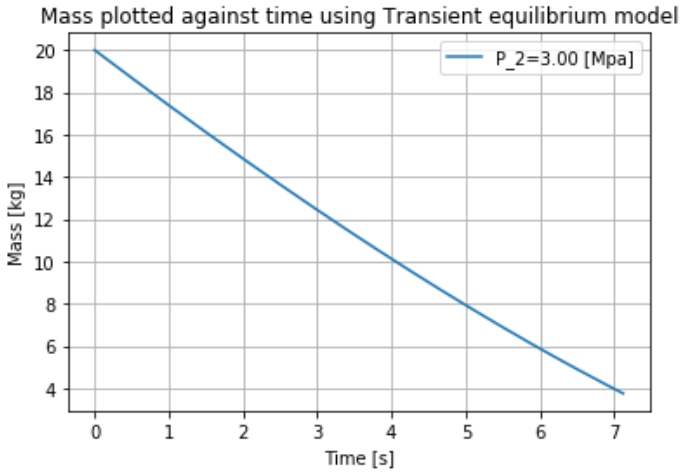
As a final note on oxidizer flow rate modeling, the various models have so far all shown that the mass flow rate will not remain constant as the upstream pressure drops. Propulse initially thought that the upstream pressure would remain relatively constant, but experimental data and the transient equilibrium model both show that this is unlikely. However, there is still a possibility that Propulse's initial assumption holds some weight if the upstream conditions do not change as quickly as the model predicts. TEM uses the enthalpy loss from the removed liquid as the only change in internal energy, which in turn determines how the upstream temperature develops. It could be that E will be affected by other factors as well, potentially leading to a slower drop in P_1 and T_1. This could result in the mass flow rate not dropping very much after all. At the end of the day, experiments need to be conducted to validate the models, which will be discussed in chapter 4.



(a) P_1 time history, ranging from 6MPa to 3.869 MPa.



(b) T_1 time history, ranging from 300.86 K to 281.76 K



(c) Tank propellant mass time history, ranging from 20 kg to 3.74 kg ($x_{final} = 0.95$)

Figure 3.17: Pressure, Temperature and mass time histories from Transient Equilibrium model

Regression rate modeling

As the transient equilibrium model now models how the oxidizer flow develops in time, it is possible to use this to model the regression rate. This was done by adding a few additional inputs to the model:

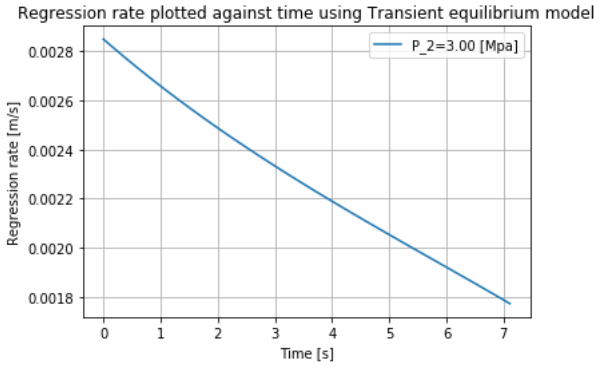
- Initial fuel grain port radius r_p
- Fuel grain port length L_p
- Regression rate equation parameters a and n
- Solid fuel density ρ_f

Recalling the regression rate theory of chapter 1, the calculations needed for regression rate modeling can now be performed. This takes place just after the operating mass flow rate is calculated for every iteration in the model and begins by calculating the cross-sectional area of the port A_p and the inner surface area of the fuel grain A_d. The oxidizer flux through the port is then calculated by dividing the flow rate with A_p. This allows equation 1.7, the regression rate equation, to be solved. Once the regression rate is found, the fuel mass flow rate is calculated with equation 1.3. Finally, the port radius for the next iteration is found with equation 3.34, similarly to how the mass and internal energy were updated:

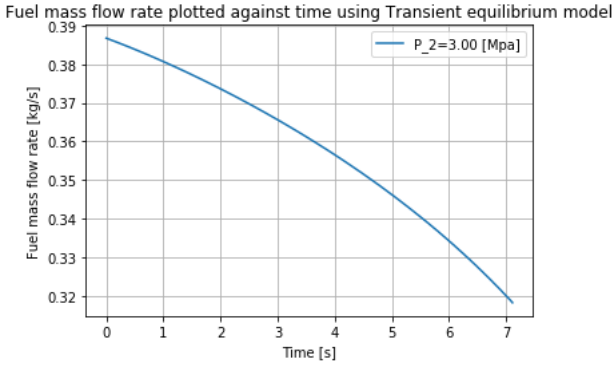
$$r_{new} = r_{old} + \dot{r}\Delta t \tag{3.34}$$

The regression rate and fuel mass flow rate of each iteration is stored in arrays, so that their averages may be found and that they may be plotted against time. Additionally, the array containing the oxidizer flow rates is divided by the one containing the fuel flow rates, so that the O/F ratio can be plotted once the iterations are over. In figure 3.18, these plots are shown. They have been made using the same input parameters as in table 3.2, with the additional input parameters shown alongside the plots.

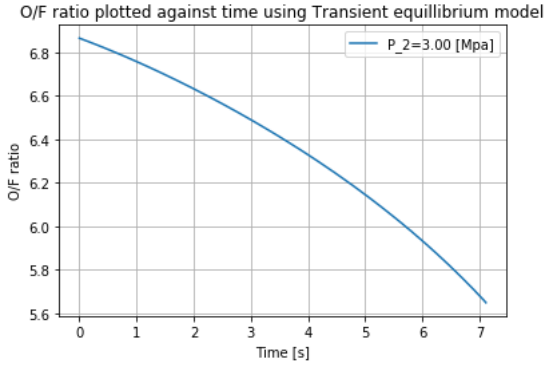
The regression rate plot shows that the regression rate will fall significantly during the burn, which is expected as it has been shown that the oxidizer mass flow rate also falls. This will lower the oxidizer flux, which in turn lowers the regression rate. One can note that even if the oxidizer flow rate was constant, the increase in port radius for the duration of the burn would also contribute towards a fall in oxidizer flux. The fuel mass flow rate also declines over time, although the O/F ratio plot indicates that the oxidizer flow rate is falling more rapidly than the fuel flow rate. Interestingly, an O/F shift similar to what was described in chapter 1 happens here despite the propellant combination of N_2O and Paraffin wax having $n = 0.5$. However, one must recall that having this exponent equal to 0.5 would eliminate the O/F shift only if the oxidizer flow rate was constant. As this is not the case here, the O/F ratio falls. That being said, the recommendation of trying to begin with slightly higher O/F ratios than the ideal and finish with slightly lower O/F ratios could potentially be possible.



(a) Regression rate time history. $\dot{r}_{avg} = 0.00227m/s$



(b) $\dot{m}_f$ time history. $\dot{m}_{avg} = 0.358kg/s$



(c) O/F ratio time history

Figure 3.18: $r_{p,initial} = 0.05m$. $L_p = 0.48m$. $\rho_f = 900kg/m^3$. $n = 0.5$. $a = 15.5 * 10^{-5}$. Note that a has been converted from the value in table 1.1 to output $\dot{r}$ in m/s.

The results that have been shown for this model are for a calculation on an injector with 24 orifices, corresponding to Injector 2 - a design that will be shown in chapter 4. The calculations here give a lower mass flow rate than the desired 2.6 kg/s. If a new calculation is made with the same input variables, except that the number of orifices is changed to 28, then $\dot{m}_{ox,avg.} = 2.66kg/s$. However, this calculation that provides the "correct" oxidizer flow rate still predicts significantly lower regression rates and fuel flow rates than Propulse's desired values from table 3.1. This suggests that Propulse may need to consider utilizing swirl injectors or other means of boosting the regression rate to achieve their goals.

Chapter 4

Injector Design & Experiments

While the models that have been shown throughout chapter 3 might be useful design tools, they still use a lot of assumptions. As was mentioned, even the best models tend to deviate from experimental data. Therefore, injectors are often iteratively designed with the help of both cold-flow and hot-fire testing. The models can potentially provide Propulse a better understanding of how the engine works and act as a starting point for injector design, but experiments are usually still necessary. For this reason, the final part of this thesis was originally intended to detail an experimental campaign on Propulse's hybrid rocket with regards to the injector. The experimental results could be used for validation of the mass flow models, allow for iterative modification and comparisons of different injectors, and assess the overall performance of the engine. However, due to the COVID-19 pandemic, Propulse's activities have unfortunately been suspended. Both finishing the construction and operating the test bench are endeavors that would require multiple Propulse members, and thus it has not been possible for the author to independently see the experimental campaign through. Therefore, this chapter will instead show the design process of a few preliminary injectors and describe proposed experiments while discussing how the testing results could potentially be used.

4.1 Preliminary injector designs

A few different injector designs that have been made with Propulse NTNU's projects in mind will be proposed in this section. SolidWorks has been used for computer aided design. These designs are based on the mass flow models of chapter 3 and the findings of chapter 2 on different injector configurations. Propulse can follow the approach shown here as a method for initial injector design.

The desired mass flow rate that Propulse intends to have is 2.6 $\frac{kg}{s}$. This was based on thrust calculations done by other members of the organization, and will not be discussed in further detail here. As has been shown in the previous chapter, getting a constant oxidizer mass flow rate with the current design of the oxidizer system does not seem to be possible. As such, the injectors have been designed while attempting to get the average mass flow rate to 2.6 $\frac{kg}{s}$. A showerhead, an impinging, and a vortex design have been made. Although they are all different, certain aspects were kept similar for each injector so that they would be easier to compare.

All the injectors have an orifice diameter of 2mm, which is on the larger side of the 1-2mm standard [12]. This was decided as the lateral space on the injector face is limited - a constraint from the feed system is that the part of the injector's area that is in contact with oxidizer has a diameter of 100mm. For more complex injector designs with angled holes, this can make it difficult to fit the needed number of orifices for the desired mass flow rate. With larger diameter holes, each hole gives a higher flow rate. Fewer orifices are then needed, saving space on the injector plate. This diameter was also chosen for the showerhead injector so that test data from this configuration could be compared to and be useful for other designs. The injector thickness L was initially set to 30mm to obtain a high L/D ratio that would increase the likelihood of choked flow. However, the manufacturer was unable to drill straight holes that deep, and the thickness was cut in half. A L/D of 7.5 is still quite high and this also halves the weight of the injector. In that sense, this is possibly a positive even if the likelihood of choked flow may be somewhat reduced.

Aluminum was chosen as the material for its low weight and machinability. All the injector plates also have 8 M6 clearance holes around their outer edge for fastening. This was determined through a few simple calculations, beginning with a calculation for the force that will be applied to the injector. This is found from the maximum pressure difference during operation, which will be a ΔP of approximately 60 bar at the beginning of any test.

$$F = \Delta P * A_{contact} = 60 * 10^5 * \pi * 0.05^2 = 47123.88N \tag{4.1}$$

This force is then multiplied by a burst safety factor of 2.5 and divided equally among the 8 screws. This results in a force of 14726 N per M6 bolt. According to engineeringtoolbox.com [47], this implies that M6 bolts of grade 8.8 or higher are needed as they have a minimum ultimate tensile load strength of 16100N. This method should ensure that the injector is safely fastened and allows the injector to be easily changeable so that different designs may be tested.

Some of the relevant geometry and operating conditions used in calculations that all the injector designs have in common are listed in the following tables:

Orifice D	Orifice L	Injector plate D	Plate contact D
$2mm$	$15mm$	$160mm$	$100mm$

Table 4.1: Geometries that every injector design that is presented have in common. The contact area is the part of the injector that is in contact with the oxidizer. Any orifices must be within this area.

Initial P_1	Final P_1	Operating P_2	Desired average $\dot{m}_{ox}$
6 MPa	4 MPa	3 MPa	$2.6\frac{kg}{s}$*

Table 4.2: Operating conditions that have been used for designing all the injectors that are presented. *2.5 kg/s for injector 1.

Due to deadlines related to Propulse's schedule, the injectors were designed at a point in time when the transient equilibrium model was yet to be made. The model

assuming a linear decline in upstream pressure was used instead, which is why a final P_1 is specified here.

4.1.1 Injector 1 - Showerhead

The first injector designed for Propulse to test was a showerhead injector. It is shown in figure 4.1, and hereby known as injector 1. Although Propulse currently intends to use an impinging design for the 2020 rocket, it is still useful to conduct experiments with a showerhead design that can be designed and manufactured relatively quickly.

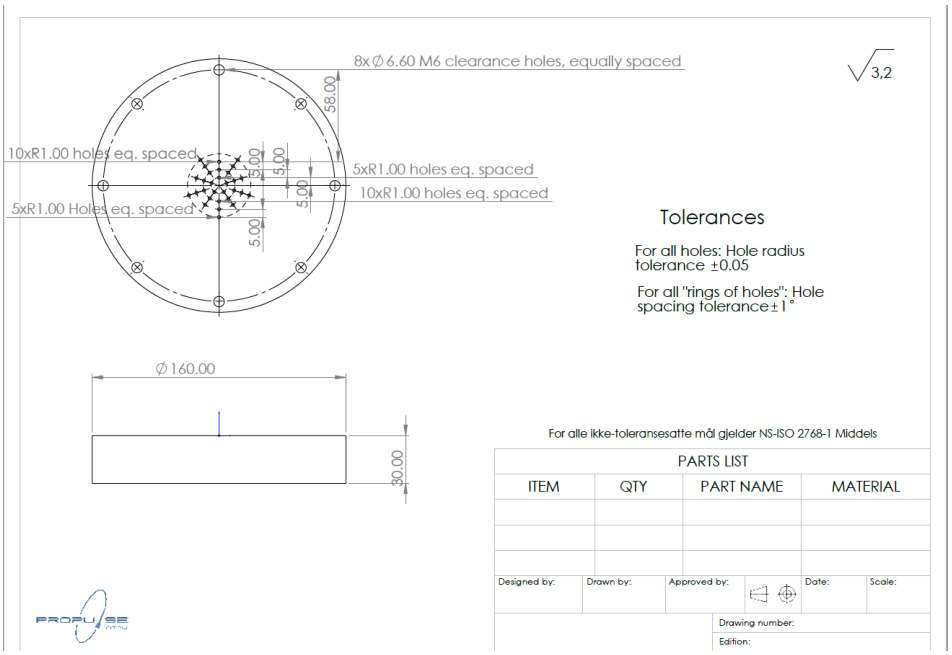
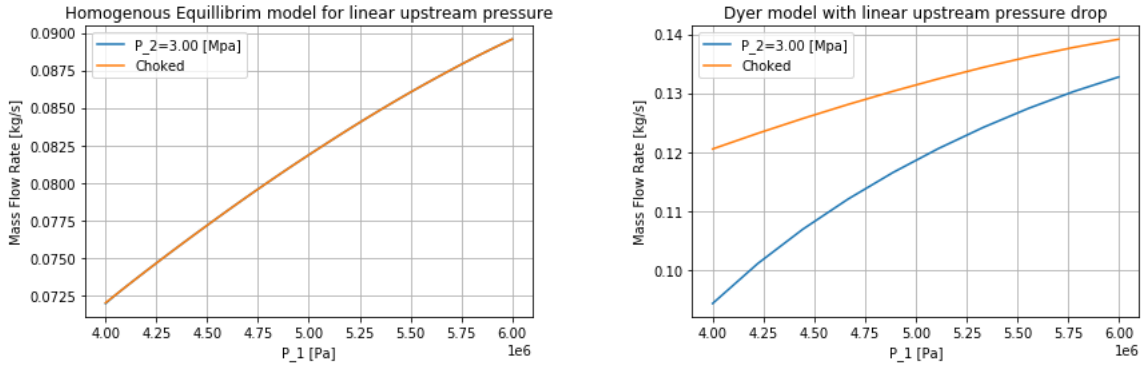


Figure 4.1: Machine drawings of injector 1. All dimensions in mm. Note that the thickness is actually 15mm, not 30mm

As this injector was designed quite early in the process, its design has been based on calculations using the homogeneous equilibrium model with a linear upstream pressure drop. Furthermore, the desired mass flow rate at this point in time was $2.5\frac{kg}{s}$, rather than 2.6. The plots for both HEM and the Dyer model are shown in figure 4.2. It was initially suspected that the Dyer model was inapplicable in saturated conditions due to the issue of it becoming a simple average of SPI and HEM. This concern has been somewhat dispelled after continued research [42], but the injector had by that time already been sent to the manufacturer with 30 orifices specified. Due to the tendency of HEM to under-predict the flow rate, this may result in flow rates that are higher than what is desired. This is particularly evident when the results from the Dyer model are observed, which predicts significantly higher flow rates. This model also does not predict choked flow for any upstream pressure, while HEM predicts choked flow for all P_1 in the range. However, valuable mass flow rate data can still be found while testing with this injector, which can then be compared to different models' predictions.



(a) Average $\dot{m}_{HEM} = 0.0815 \frac{kg}{s}$ (b) Average $\dot{m}_{Dyer} = 0.1167 \frac{kg}{s}$

Figure 4.2: Calculations on a single injector orifice for injector 1 with N_2O.$C_d = 0.9$.

Desired avg. $\dot{m}$	C_d	No. of Orifices	Avg. $\dot{m}_{HEM}$	Avg. $\dot{m}_{Dyer}$
$2.6[\frac{kg}{s}]$	0.9	30	$2.445[\frac{kg}{s}]$	$3.5[\frac{kg}{s}]$

Table 4.3: Table showing total mass flow rates for Injector 1 using the different models

In table 4.3, the relevant calculation results used for designing the injector are presented. In hindsight, the injector should perhaps have been designed with fewer orifices initially when considering the results from the Dyer model. By selecting a high discharge coefficient and 30 orifices, the HEM-calculated flow rate is slightly lower than desired. This was the goal, as during the iterative design of the injectors it is easier to increase the mass flow rate it produces than decrease it. Adding more holes, chamfering the orifice inlets, or increasing their diameter are all methods that can be used to increase the mass flow rate. Decreasing it, on the other hand, would likely require a completely new injector plate. Therefore, it can be wise to assume high discharge coefficients initially and thus predict that fewer orifices are needed for the desired mass flow rate. Due to the abnormally high C_d, the actual flow rate will then be somewhat lower than desired. Then, one can modify the injector as required after testing.

Injector 1 was the only injector that was produced before COVID-19 put a halt to any further activity. In figure 4.3, it is shown attached to the bulkhead.

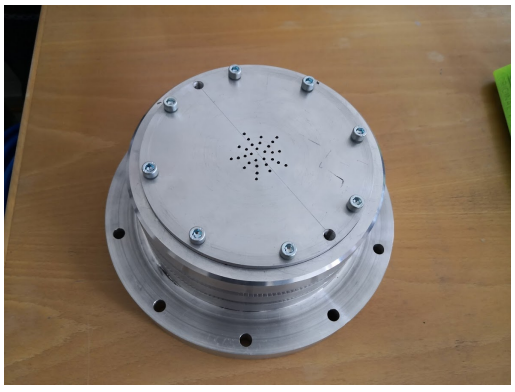


Figure 4.3: Picture of Injector 1

4.1.2 Injector 2 - Impinging

Injector 2 will most likely be the basis for the injector of choice for the final rocket. This is due to the various advantages of impingement that were detailed in chapter 2 alongside its relative ease of manufacturing and design. However, drilling angled holes precisely requires a 5-axis CNC mill, which in practice turned out to be more difficult to get access to than expected. Therefore, this injector did not get produced before the COVID-19 shutdown and has yet to be manufactured.

The machine drawing in figure 4.4 shows that this design uses alternating triplet and doublet impinging orifices.

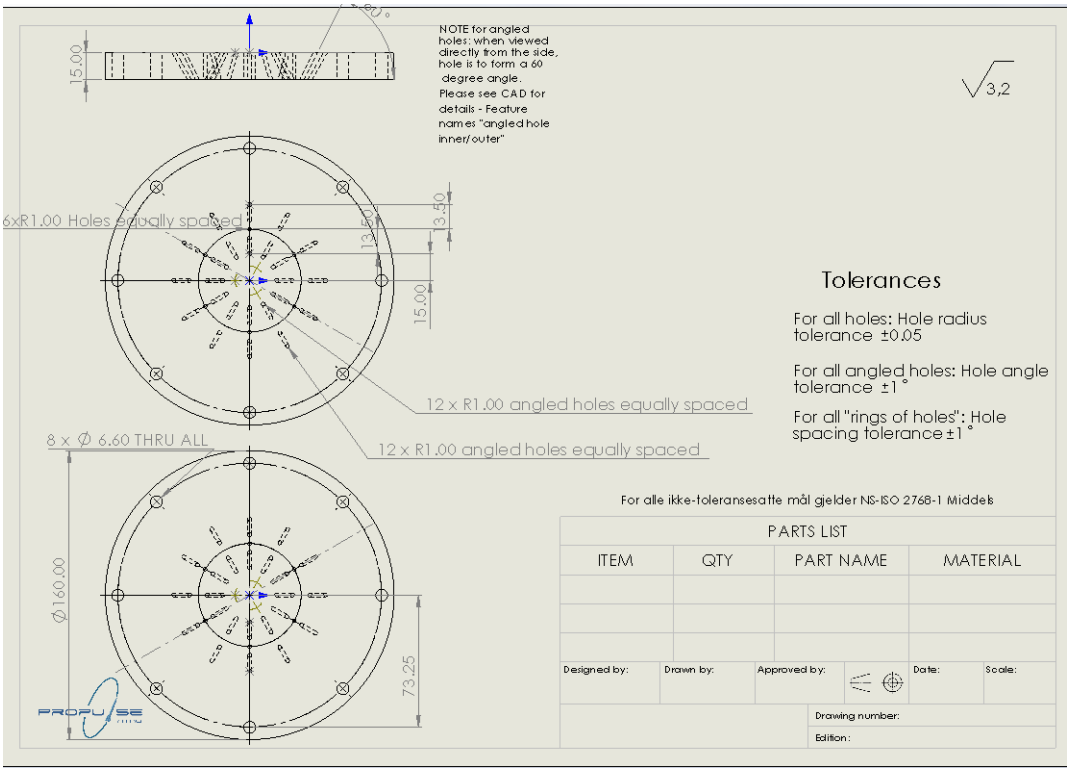


Figure 4.4: Machine drawing of Injector 2

The straight-holed orifices that create triplets are optional, and are shown here as a suggestion for additional holes should the need arise to increase the total flow rate. The doublets form an impingement angle of 60 degrees and are made by angling holes against each other, as illustrated in figure 4.5. The impingement angle was chosen as a middle ground between increased atomization and available space on the injector plate.

This design was made with the Dyer model for the mass flow rate using the linear upstream pressure assumption. However, even the doublet-only configuration with a total of 24 orifices produces a mass flow rate that is slightly too high when using the high C_d calculation from figure 4.2b. That being said, the actual discharge coefficient is likely to be lower than 0.9. In Sutton's Rocket Propulsion Elements [6], it is stated that a good estimate of the single-phase discharge coefficient for straight-holed orifice with sharp edges is approximately 0.65. This is backed up

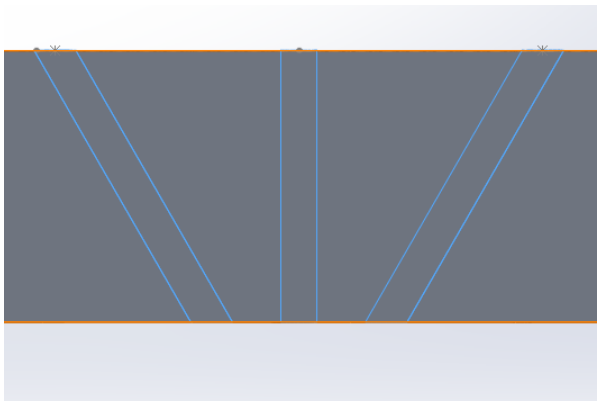


Figure 4.5: CAD cut-out of triplet element in injector 2. Doublets are identical but lack the central straight-holed orifice.

by data on CO_2 discharge coefficients found by Waxman et. al [12] and should be a reasonable estimate. Furthermore, Injector 2 uses angled holes which normally should entail more friction losses, so the discharge coefficient is likely to decrease further. Therefore, it was deemed that the risk of the flow rate being too large was low and thus acceptable to have 24 orifices initially. In table 4.4, mass flow rates from calculations with different C_d are shown to highlight this. As more realistic

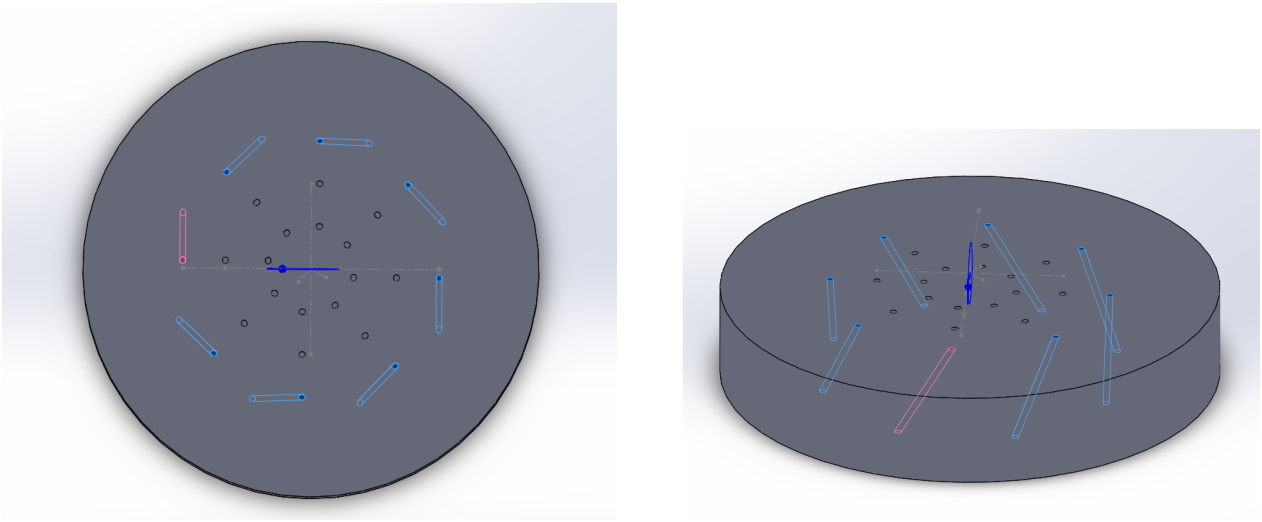
C_d	Avg. $\dot{m}_{Dyer}$	Max. $\dot{m}_{Dyer}$	Min. $\dot{m}_{Dyer}$
0.9	2.808	3.1848	2.256
0.835	2.605	2.955	2.101
0.65	2.027	2.301	1.636

Table 4.4: Table showing N_2O mass flow rates in $\frac{kg}{s}$ for Injector 2 (Doublets only, 24 orifices)

discharge coefficients are used in the calculations, the mass flow rates go below the desired average. Again it is preferable to initially design injectors that give lower mass flow rates than desired than the reverse, as it is an easier task to add a few orifices than to remove unwanted ones. The doublet/triplet design of injector 2 makes it so that any additional orifices that are required can be straight-holed, but still take part in impingement by changing some of the existing doublets into triplets. This eliminates any issues with acquiring access to 5-axis mills during modifications of injector 2.

4.1.3 Injector 3 - Vortex

The third and final injector that was designed for Propulse was a vortex injector. It was thought that this would be easier in terms of manufacturing than the regular swirl injectors, while still possibly being a way to take advantage of swirling flow. A top-down and an isometric view of the injector are presented in 4.6 to show how the orifices are aligned.



(a) Top-down view

(b) Isometric view

Figure 4.6: CAD of Injector 3

This injector would serve as an alternative to the impinging design if higher regression rates were needed after testing. The trade-off would be that the atomization features of impinging injectors would be lost. The orifices are angled 60°on the injector plate, similar to the impinging design, but set up in a pattern such that the oxidizer is injected with a tangential velocity component. Note that the two inner "circles of orifices" that are not highlighted in the figure are set up in a similar pattern to the outer ring.

Injector 3 also has 24 orifices, meaning that the mass flow rate calculations are the same as for injector 2. The two injectors may be different, but due to the similarity of the orifices themselves, C_d should turn out to be very similar to the doublet-only configuration of the impinging design.

4.2 Experimental setup

The most important components of the experimental setup that Propulse NTNU started to build are shown in the piping & instrumentation diagram of figure 4.7. Note that the components are not shown to scale. Additionally, only the experimental setup and procedure from the point at which the tank has been filled will be discussed here. In other words, any part of the setup upstream of the tank, including the relief valve, dump valve, and quick-release system will not be discussed here.

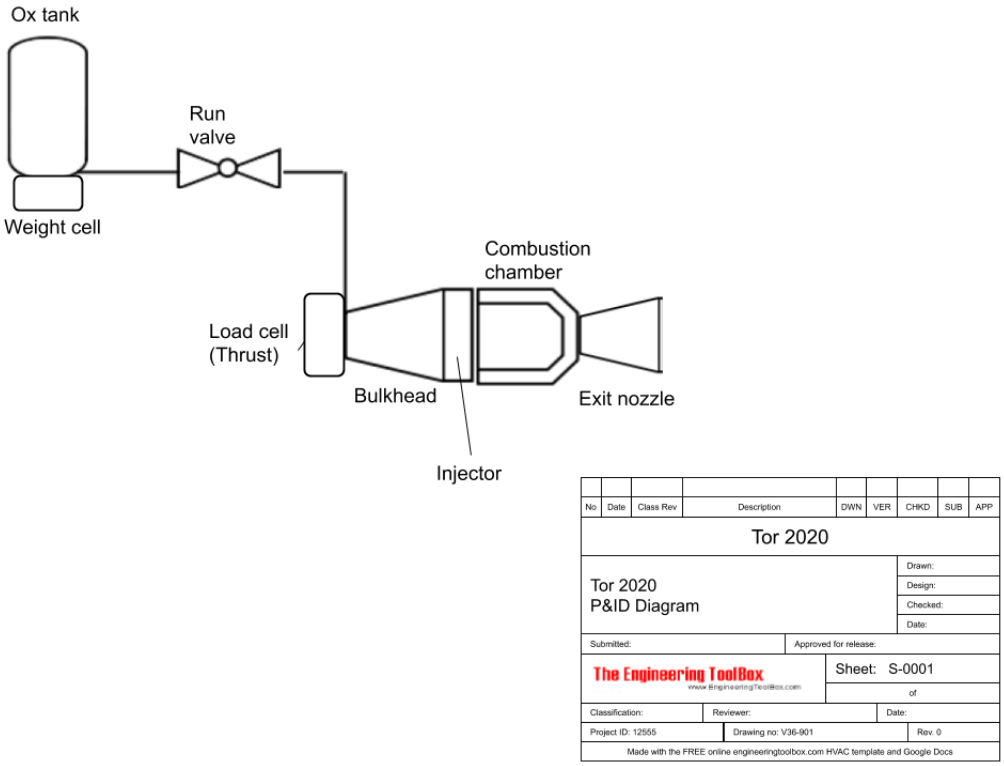


Figure 4.7: Piping and instrumentation diagram of the test bench

When the oxidizer tank has been filled and the pressure has stabilized at the desired level, the test is ready to begin. The tank is equipped with pressure and temperature sensors, and also rests on a weight cell that will continuously monitor the mass of the tank during a test. Venturi flow rate measurements are not always reliable for high vapor pressure propellants, as was mentioned in section 1.4. The weight cell is therefore chosen as the method to measure the mass flow rate. The control valve is also known as the run valve, and is a ball valve. When it opens, the fluid in the tank begins to flow through the system. The run valve also ends the test, by closing when the tank pressure drops to a predesignated value. It was ensured that no parts of the feed system, including the control valve, had a cross-sectional area smaller than the total orifice area in the injector, such that the injector is the flow limiting device.

The front bulkhead that the injector is attached to has been made with a diverging section internally so that the fluid velocity is reduced just upstream of the injector. As discussed in chapter 3, this was done to keep P_1 as close to the tank

pressure as possible. One addition that Propulse should consider for future testing set-ups is to include a pressure sensor in the area just upstream of the injector, which would allow for better validation of the models and could be used to gain insight into the feed system.

The flow then moves through the injector, which marks the point at which the cold-flow test setup deviates from the hot-fire tests. In a cold flow test, the injector is the final component of the system and the fluid is injected into atmospheric conditions. Thus, no ignition or combustion takes place, as "cold flow" implies. Carbon dioxide will be used for most cold-flow testing for the reasons outlined previously in this work. In a hot-fire test, on the other hand, the combustion chamber and exit nozzle are included. The oxidizer, N_2O in Propulse's case, is injected into the combustion chamber, where it is ignited. The hot combustion gases are accelerated through the nozzle, which produces thrust. A load cell is attached to the front of the rocket to measure this force.

Although experiments can provide a means of validating the models, one must keep in mind that experimental data is not always correct either. There are many factors that can affect the validity of measurements. Therefore the uncertainty of measurements should also be taken into account when trying to validate the models. For example, using a weight cell to measure the mass flow rate is not necessarily going to be very accurate. If the measured data is always taken as the absolute truth, this could lead to wrongful assessments. Care should be taken as to how experimental data is processed and to use the best available sensors.

As a final point, the entire test bench set-up was designed such that it does not require anyone to be close to it as a test is ongoing. Rockets are obviously hazardous, and thus all components can be activated remotely. This is part of the rule set of the Spaceport America Cup. Other safety requirements dictated by the SA cup, as well as American and Norwegian law will also be followed.

4.3 Cold-flow experiments

The first kind of experiment that is proposed is cold-flow testing. This means that the combustion chamber removed, and the injector would vent to ambient conditions. This type of testing is useful because it allows for many of the components of the engine to be tested without the additional costs and hazards associated with combustion. In these tests, CO_2 can be used as an analog to N_2O to further reduce the cost and danger. However, one limitation that is important for assessing the injector is that venting to atmospheric conditions may significantly affect the mass flow rate because the downstream pressure is now much lower than the operating P_2 of the combustion chamber. When venting to the atmosphere the flow will almost assuredly choke due to the low atmospheric pressure. If the flow is expected to choke for the combustion chamber conditions as well, the mass flow rate data should be directly transferable from the cold flow test to the actual firing. If this is not the case, the cold flow test can still be used to validate the models by comparing the choked flow rate that the model predicts to the experimental data.

The fact that the discharge coefficient is an empirically determined parameter makes validating the models slightly more difficult. Simply choosing the discharge coefficient that makes the model match the experimental data would not truly allow for the predictive ability of different models to be assessed. A major reason for modeling is to understand how the system functions prior to testing, so if the model is completely dependent on the experiments it will not be as useful. On the other hand, if one uses a guess for the discharge coefficient it could lead to wrongful predictions without necessarily invalidating the model. The discharge coefficient mostly depends on injector geometry, with operating conditions and choice of fluid having a secondary, minor effect. Therefore, a commonly used compromise is to find the single-phase discharge coefficient by running water through the injector and use this as the discharge coefficient for the different models. Table values for discharge coefficients are also often found in this way and can be useful if the injector geometry matches up [12] [6].

A potentially better approach could be to create a downstream chamber for use with cold flow testing similar to the one made by Waxman et. al [12] for their experiments on N_2O and CO_2. This downstream chamber would be equipped with pressure regulators so that as it is filled with pressurizing gas, the pressure will rise to some predetermined value and then stay there for the remainder of the test. Setting the downstream pressure to a high level where even high vapor pressure fluids would operate in the regime of the SPI model would allow the discharge coefficient to be determined more accurately. This could be helpful for validation of the models. Of course, a pressurized downstream chamber would also allow cold flow experiments to be performed at the actual operating pressure of the combustion chamber, eliminating limitations of venting to ambient conditions. Another potential problem with venting to ambient conditions is that this may cause issues while using carbon dioxide for cold-flow testing. As was mentioned in section 1.4, CO_2 has a triple point pressure that is above atmospheric pressure. This means that the CO_2 could potentially be exiting the injector as a solid-vapor mixture, which could cause issues with the mass flow rate. Unfortunately, this does not seem to be mentioned much

throughout the literature and without the opportunity to perform the experiments it remains unclear whether or not this is truly a problem. For example, it could be that the temperature does not reach low enough levels in the injector for sublimation, making the solid phase consideration unimportant. In any case, venting to more realistic chamber pressures would certainly eliminate this issue. However, this kind of chamber has not been made by Propulse yet. It is recommended that Propulse consider using this kind of downstream pressure chamber for cold-flow testing in the future.

Whether or not a downstream chamber is used, the test data can be plotted against the predictions of the models once C_d has been determined. This can then be used to validate the models by assessing their accuracy when compared to the data. Naturally, the mass flow rate prediction is especially important and the prime indicator of how good the model is would be how accurately the model predicts the choked flow rate. Other important parameters to assess would be the tank pressure and temperature data, which can be compared to the Transient Equilibrium model as well as the linear assumption. The remaining tank mass when the test is complete can also be compared to what TEM predicts.

Some deviation from the experimental data is likely due to the many assumptions that the models utilize, even for the best model. Therefore it could be useful to also explore whether it is possible to find an "effective" discharge coefficient that makes the model fit the data. As mentioned earlier, this does not necessarily validate the model but it can be used to aid injector design. Say that the transient equilibrium model seems to somewhat capture the trend, but not quite the correct values of the flow rate when using the standard discharge coefficient. Then, an effective C_d could potentially be found to make the model match the data. Following that, the code could be run again with this effective discharge coefficient. This could potentially be used to assess not only the cold-flow test's ambient downstream conditions but for other downstream pressures as well. As such, it might be possible to obtain a better prediction of the flow rate to the combustion chamber conditions than previously. This could in turn shed light on if the injector should be modified to achieve the desired flow rate before the motor is ever fired.

The final aspect of cold-flow testing is to get an idea of the spray pattern, distribution, and atomization of the oxidizer. This is particularly relevant in the case that utilizes a transparent downstream pressure chamber. When venting to ambient conditions the pressure drop is much larger than it would be in the rocket, which will affect the flow significantly. That being said, venting to ambient conditions can provide a preliminary assessment. The most basic analyzing tool would be a simple video showing the spray - although only the general trends of the spray pattern, distribution, and atomization of the oxidizer can be seen with this method. High-speed cameras can be used for better assessments. Laser technology that can be used to find the actual droplet sizes also exists. This can be used if it is deemed necessary to accurately know the degree of atomization, but this is most likely not the case for Propulse.

4.4 Hot-fire testing

Hot-fire tests are the closest thing to an actual launch that can be done on the ground and allows for the components downstream of the injector to be tested, unlike the cold-flow tests. When hot-firing, the injector and oxidizer system will also be tested at the actual operating conditions of the rocket. Therefore, the data gathered from hot-fire tests is the most important when evaluating whether the targets for thrust, mass flow rate, etc. are met, or if modifications must be made. Hot-fire tests will be used by Propulse to assess the entire system, but here the injector will naturally be in focus.

For validation of the mass flow rate models that have been developed in this work, the general procedure is very similar for both hot-fire testing and cold-flow testing - the data from tests should be compared to the models' prediction. One thing to keep in mind, however, is that unlike the cold-flow case where the downstream conditions are ambient, combustion is now occurring downstream. If the chamber pressure is not relatively constant, this could be an issue for the mass flow rate models that assume a constant downstream pressure, especially if the flow is not choked. On the other hand, having the actual operating conditions downstream will naturally provide a more direct way of assessing whether the mass flow rate is behaving as expected. One important recommendation that the author has for Propulse is to find a way to monitor the combustion chamber pressure during a hot-fire test. This way it could be ensured that the chamber pressure is at the desired value. The validation of the models would also benefit from this, as it could help in identifying the cause of potential deviations from the experimental data.

With hot-fire testing, the regression rate modeling that was done at the end of chapter 3 can now be compared to the data that is gathered. This obviously cannot be done with cold-flow testing and highlights another reason why hot-fire testing is so important. Recall that the a and n parameters in the regression rate equation are empirically determined. As a result, the parameters that are used in any modeling done before hot-firing the motor have been set to table values from different cases than Propulse's. Therefore, after a hot-firing these parameters could be redetermined, and then the model should be run again. The regression rate prediction can then be compared to the data to validate the model more accurately.

Hot-fire testing would also be important for comparing different injector designs. Assessing how impingement improves combustion stability and efficiency when compared to a showerhead injector would be interesting. Due to flash vaporization, there might not be that much to gain from utilizing the more complex impinging design, which could cut down on manufacturing time and difficulty. Similarly, figuring out how much of an effect vortex/swirl injectors have on the regression rate could be very valuable, especially if the targets are not met.

Chapter 5

Summary, Conclusions & Future Work

As was outlined in the introduction, the goal of this thesis was to provide Propulse NTNU with a better understanding of how hybrid rocket engine injectors function, in order to develop a method for how injectors may be designed. To assess how this work has handled the problem it set out to solve, a summary of the work will be presented below.

In the first chapter of the thesis, hybrid rockets were introduced. This background theory highlighted how hybrid rockets function, some of the challenges associated with them, and how important the oxidizer mass flow rate is for the performance of the engine. This provided some motivation as to why injector design is important and served as the foundation for the rest of the thesis. Nitrous oxide was also introduced here, as its two-phase properties significantly impact how the oxidizer flows through the injector in Propulse' case. Some of the most interesting findings were that choking the flow may be a way to achieve a constant oxidizer flow rate if the tank pressure can be kept constant and that choking can reduce instabilities. However, it was also discovered that the tank pressure will most likely fall during the burn, contrary to Propulse's initial assumption.

The second chapter focused on different injector schemes. Specifically, this chapter showed the atomization and performance-enhancing aspects of various injector configurations. The basic theory of atomizing jets was explained. An important finding was that nitrous oxide is likely to be dominated by a flash vaporization mode that results in an aerosol cloud. This means that even showerhead injectors can potentially provide sufficient atomization. It was also shown how impinging injectors can further enhance atomization. Some geometry features of both impinging and swirl injectors that should be directly useful for design were presented. Finally, it was shown that the swirl injectors could have a profound effect on the regression rate, potentially increasing it many times over.

In chapter 3 the mass flow rate was assessed with a variety of models. The most relevant models for Propulse's case are the two-phase models that were shown, but single-phase modeling was also done to serve as a baseline. This highlighted how the two-phase flow significantly affects the flow rate and how choking could be uti-

lized to make the flow rate constant if the upstream pressure remained constant. However, when the tank dynamics were taken into account, it became clear that the upstream pressure and subsequently the flow rate would drop significantly for the duration of the burn, regardless of which mass flow rate model was used. Both the alternatives for modeling the tank dynamics resulted in similar mass flow rate predictions, but the transient equilibrium model provided predictions for several variables that the linear model cannot assess. For Propulse's case the two-phase flow rate models significantly differed in their prediction of choked flow, with HEM predicting choked flow for all P_1 while the Dyer model did not predict choked flow at all. The Dyer model seems to be the most accepted throughout the literature and thus it could mean that Propulse will not have choked flow in their engine which could potentially give rise to combustion instabilities. Finally, as part of the transient equilibrium model, the regression rate, fuel flow rate, and O/F ratio could be modeled as well. The most important finding here was that the regression rate would also drop for the duration of the burn, and was lower than Propulse NTNU desired even at its maximum.

The models have not been compared to experimental data as the experimental portion of this work was stopped due to the COVID-19 pandemic. This was obviously unexpected, and consequently chapter 4 developed quite differently to what was originally intended. The chapter includes a methodology for how injectors may be designed, by showing the design process and reasoning behind three preliminary injector designs. Propulse can use this as a template as they continue to design injectors in the future. This approach, however, is heavily based on mass flow rate models that have yet to be validated. Therefore, the remainder of the chapter was dedicated to how the experimental campaign could have been used for model validation and iterative injector design.

Reflecting on the summary above, the work has covered the most important points pertaining to the problem that was outlined at the start. The thesis has presented the key theory that relates to the injector, and tools have been developed for predicting the mass flow rate through the injector. Armed with these tools, a method for how injectors can be designed was shown. The research has found that the rocket is likely to behave differently to what Propulse had originally anticipated and that they may need to reconsider certain design choices to achieve the desired performance. This comes with the caveat that the experiments that could have confirmed the modeling results were not performed. Instead, guidelines on how the experiments can be used were shown for Propulse to follow at a later stage.

The author has also identified a number of research topics that could be interesting for Propulse NTNU in the future. Some of these are related to the experimental campaign and have been mentioned previously, but there are also other avenues for future work that could be done. A few ideas for the future work are listed below:

- Compare models with experimental data to determine their usefulness
- Continue to develop TEM to include thrust calculations, tank wall heat transfer, etc.
- Adapt models for swirl injectors: can the inlets be used as an analog to standard injector orifices or will the unique geometry change things?
- Find relations that can predict how swirl, vortex and impinging injectors will affect combustion/regression rate through hot-fire testing
- Characterize how angled orifices change the discharge coefficient
- Perform cold-flow tests with both CO_2 and N_2O to ambient conditions to assess if the analog is valid, especially with respect to the potential issue with the triple point of CO_2
- Build a downstream chamber for cold-flow testing
- Find a way to measure combustion chamber pressure during hot-fire testing
- Consider using an external pressurizing gas for maintaining upstream pressure and/or supercharging

By working with these ideas, Propulse could continue to improve their understanding of hybrid rockets and further develop methods of assessing how the injector functions.

In conclusion, this thesis will certainly contribute to Propulse's knowledge base on rocket science and it has provided them with an approach for how they can design injectors. As such, the objective of this thesis has been achieved. The tools and knowledge that this thesis provides should put Propulse at a much better starting point for design and as a final remark, the author wishes Propulse NTNU good luck in their future endeavours.

Appendix A

Python Code

A.1 Saturation Line plot

```
# -*- coding: utf-8 -*-
"""
Created on Sat Apr 25 15:34:42 2020

@author: asus
"""

import numpy as np
import matplotlib.pyplot as plt
import CoolProp
from CoolProp.CoolProp import PropsSI
from CoolProp.Plots import PropertyPlot



T=np.linspace(182.23,309.52,1000) #limits are triple point and
    critical point temperatures
P=np.zeros(1000)
for i in range(len(T)):
    currentP=PropsSI('P','T',T[i],'Q',1,'N2O')
    P[i]=currentP

tripPoint=(T[0],P[0])
critPoint=(T[-1],P[-1])

plt.plot(T,P)
plt.scatter(tripPoint[0],tripPoint[1])
plt.scatter(critPoint[0],critPoint[1])
plt.annotate("Triple Point",(tripPoint[0]-5,tripPoint[1]+600000))
plt.annotate("Critical Point",(critPoint[0]-30,critPoint[1]))
plt.title('Vapor-liquid saturation curve of N2O')
plt.xlabel('T [K]')
plt.ylabel('P [Pa]')
plt.grid()
plt.legend()
#plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
plt.show()
```

Listing A.1: Code for Generating N_2O Saturation line

A.2 Single-Phase Incompressible Model

```
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 11 11:51:51 2019

@author: Jonas

Single-Phase Incompressible Model
"""
import numpy as np
import matplotlib.pyplot as plt
from CoolProp.CoolProp import PropsSI

#Input example:
Cd=0.75 #discharge coeff.
D_2=2e-3 #diameter of orifice [m]
P_1=np.linspace(6e6,4e6,100) #Upstream pressure, each new value provides a new graph
Fluid='N2O' #Must be such that CoolProp understands
steps=100 #number of iterations used to generate each graph, i.e. size of delta P
Operating_P2=3e6 #downstream pressure of interest (needs to match P_1?)

#Function starts here
def SPI_Model(Fluid,P_1,Cd,D_2,steps,Operating_P2):
    A_2=(np.pi/4)*D_2**2 #calculate area of orifice
    number_graphs=len(P_1) #number of graphs, new graph for each value of P_1

    #initialize arrays for rho, T_1, DeltaP, mSPI, Operating values
    rho=np.zeros(number_graphs)
    T_1=np.zeros(number_graphs)
    mSPI=np.zeros((steps,number_graphs))
    DeltaP=np.zeros((steps, number_graphs))
    Operating_DeltaP=np.zeros(number_graphs)
    Operating_FlowRate=np.zeros(number_graphs)
    #P_2=np.zeros((steps,number_graphs))

    #calculate values for different P_1 cases
    for j in range(number_graphs):
        #P_2=np.linspace(P_1[j],0,steps)
        DeltaP[:,j]=np.linspace(0,P_1[j],steps)
        #DeltaP[:,j]=P_1[j]-P_2[:,j]
        rho[j]=PropsSI('D','P',P_1[j],'Q',0,Fluid) #Saturated liquid density
        T_1[j]=PropsSI('T','P',P_1[j],'Q',0,Fluid) #Saturated liquid temperature (not used)
        #print(T_1)

        for i in range(steps):
            mSPI[i,j]=Cd*A_2*np.sqrt(2*rho[j]*DeltaP[i,j])

        Operating_DeltaP[j]=P_1[j]-Operating_P2
        Operating_FlowRate[j]=np.interp(Operating_DeltaP[j],DeltaP[:,j],mSPI[:,j])

```

```
50
51          plt.plot(DeltaP[:,j],mSPI[:,j],label='P1=%.2f [MPa]'%(P_1[j
    ]/(1e6)))
52
53      Average_FlowRate=np.average(Operating_FlowRate)
54
55      plt.title('Single-Phase Incompressible Model')
56      plt.xlabel('P1-P2[Pa]')
57      plt.ylabel('Mass Flow Rate [kg/s]')
58      plt.grid()
59      plt.legend()
60      plt.show()
61
62      plt.figure()
63      plt.plot(P_1,Operating_FlowRate,label='P2=%.2f [MPa]'%(
    Operating_P2/(1e6)))
64      plt.title('Single-Phase Incompressible Model')
65      plt.xlabel('P1[Pa]')
66      plt.ylabel('Mass Flow Rate [kg/s]')
67      plt.grid()
68      plt.legend()
69      plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
70      plt.show()
71
72      print('The average flow rate is:')
73      print(Average_FlowRate)
74
75      return mSPI,Operating_FlowRate, Operating_DeltaP
```

Listing A.2: Code for Single-Phase Incompressible Model

A.2.1 Simplified SPI for use with Dyer model

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Dec  9 15:30:26 2019
4  SPI for Dyer
5  Changed to only calculate for one upstream pressure.
6  Also changed so that the minimum P_2 (given by deltaP below) is 1
       bar, to match HEM.
7  NOTE FOR CO2: minimum P_2=5.2bar
8  Start from highest deltaP to match HEM as well.
9
10 Furthermore, DeltaP cannot be 0 as this will cause a division by 0
       in
11 the Dyer model's calculation of k, so has been set to stop at 1.
12 @author: asus
13 """
14 import numpy as np
15 import matplotlib.pyplot as plt
16 from CoolProp.CoolProp import PropsSI
17
18 #Input example:
19 Cd=0.75
20 D_2=2e-3 #diameter of orifice [m]
21 P_1=5.05e6 #Upstream pressure, each new value provides a new graph
22 Fluid='N2O' #Must be such that CoolProp understands
23 steps=1000 #number of iterations used to generate each graph, i.e.
       size of delta P
```

```
24
25 #Function starts here
26 def SPI4Dyer_Model(Fluid,P_1,Cd,D_2,steps):
27     A_2=(np.pi/4)*D_2**2 #calculate area of orifice
28
29     mSPI=np.zeros(steps)
30     if Fluid=='N2O':#See introduction text above for explaination
31         DeltaP=np.linspace(P_1-1e5,1,steps)
32     else:
33         DeltaP=np.linspace(P_1-5.2e5,1,steps)
34     rho=PropsSI('D','P',P_1,'Q',0,Fluid) #Saturated liquid density
35
36     for i in range(steps):
37             mSPI[i]=Cd*A_2*np.sqrt(2*rho*DeltaP[i])
38
39     """
40     plot here if desired
41
42     plt.plot(DeltaP,mSPI,label='P1=%.2f [MPa]'%(P_1/(1e6)))
43     plt.title('Single-Phase Incompressible Model')
44     plt.xlabel('P1-P2[Pa]')
45     plt.ylabel('Mass Flow Rate [kg/s]')
46     plt.grid()
47     plt.legend()
48     plt.show()
49     """
50     return mSPI
```

Listing A.3: Adjusted SPI model for use with Dyer Model

A.3 Perfect Gas Model

```
1  # -*- coding: utf-8 -*-
2  """
3  @author: Jonas
4
5  Perfect gas model
6
7  Calculates mass flow rates using the Perfect Gas model
8  """
9
10
11 import numpy as np
12 import matplotlib.pyplot as plt
13 from CoolProp.CoolProp import PropsSI
14
15
16 #Input Example:
17 Cd=0.75 #Discharge Coefficient
18 D_2=0.0015 #[m]
19 P_1=np.array([5.05e6, 4.6e6,3e6]) #[Pascal] NOTE: MAX 6 plots can
       be displayed!
20 T_1=293 #[Kelvin] Care: pick such that fluid is in gaseous region
       for your pressure, see phase diagram.
21 Fluid='N2O' #Must be such that CoolProp understands!
22 steps=100 #iteration steps in mass flow calc
23 #Function starts here#
24 def PerfectGasModel(Fluid,T_1,P_1,D_2,Cd):
```

```
    number_graphs=len(P_1)
    P_ratio=np.linspace(0,1,steps)
    size=len(P_ratio)

    #initialize mass flow, critical values and property arrays:
    mModel=np.zeros((size,number_graphs)) #will be shown in dashed
line, follows PG mass flow model
    mPG=np.zeros((size,number_graphs)) #"actual" mass flow using PG
    model but taking choked flow into account
    P_ratio_crit=np.zeros(number_graphs) #critical pressure ratio
    mCrit=np.zeros(number_graphs) #critical mass flow ratio
    rho_1=np.zeros(number_graphs) #density upstream

    #heat capacities and ratio:
    Cp=np.zeros(number_graphs)
    Cv=np.zeros(number_graphs)
    k=np.zeros(number_graphs)

    A_2=(np.pi/4)*D_2**2 #Calculate orifice Area

    #begin first for loop - each iteration gives mass flow for a
new P_1
    for j in range(number_graphs):
        #Calculate remaining thermodynamic properties, enforcing
gaseous phase.
        rho_1[j]=PropsSI('D','T',T_1, 'P|gas',P_1[j],Fluid)
        Cp[j]=PropsSI('C','T',T_1, 'P|gas',P_1[j],Fluid)
        Cv[j]=PropsSI('O','T',T_1, 'P|gas',P_1[j],Fluid)
        k[j]=Cp[j]/Cv[j]


    #find critical values
        P_ratio_crit[j]=(2/(k[j]+1))**(k[j]/(k[j]-1))
        mCrit[j]=Cd*A_2*rho_1[j]*np.sqrt(2*Cp[j]*T_1*((P_ratio_crit
[j])**(2/k[j])-(P_ratio_crit[j])**((k[j]+1)/k[j])))

    #begin for loop for each P_ratio to calculate mass flow rate
        for i in range(size):
            if P_ratio[i]>P_ratio_crit[j]:
                mPG[i,j]=Cd*A_2*rho_1[j]*np.sqrt(2*Cp[j]*T_1*((
P_ratio[i])**(2/k[j])-(P_ratio[i])**((k[j]+1)/k[j])))
            else:
                mPG[i,j]=mCrit[j]
            mModel[i,j]=Cd*A_2*rho_1[j]*np.sqrt(2*Cp[j]*T_1*((
P_ratio[i])**(2/k[j])-(P_ratio[i])**((k[j]+1)/k[j])))

        #Begin plot
        colors=['r','b','g','m','y','c']
        dashcolors=['r--','b--','g--','m--','y--','c--']
        plt.plot(P_ratio,mPG[:,j],colors[j], label='P1=%.2f [MPa]'
%(P_1[j]/(1e6)))
        plt.plot(P_ratio, mModel[:,j],dashcolors[j])
    #finish plot
    plt.title('Perfect Gas Model')
    plt.xlabel('P2/P1')
    plt.ylabel('Mass Flow Rate [kg/s]')
    plt.grid()
    plt.legend()
```

```
75      plt.show()
```

Listing A.4: Code for Perfect gas model

A.4 Homogeneous Equilibrium Model

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Tue Feb 11 17:36:06 2020
4
5  @author: asus
6  """
7
8  # -*- coding: utf-8 -*-
9  """
10 Created on Tue Dec  3 18:12:56 2019
11
12 Homogenous Equillibrium Model
13
14 subscript 1 denotes upstream of injector, assumed to be equal to
       tank
15 subscript 2 denotes downstream of injector
16
17 Note that for P_2, the lower bound has been set to (slightly above)
       the triple point pressure as very low values caused issues with
       the entropy calcs
18 Also, P_2 is set to go up to P1-1 as we otherwise get a division by
       0 in the Dyer model
19
20 @author: Jonas
21 """
22
23 import numpy as np
24 import matplotlib.pyplot as plt
25 from CoolProp.CoolProp import PropsSI
26
27 #input example:
28 Cd=0.75 #Discharge Coefficient
29 D_2=1e-3 #Injector Orfice Diameter
30 Fluid='N2O' #nitrous oxide
31 steps=100 #Number of steps in iterations in mass flow calc for each
       P_1
32 Operating_P2=3e6 #Operating pressure in burnchamber, the value we
       are interested in.
33 P_1=np.linspace(6e6,4e6,100)  #linear upstream pressure vector
34
35 #begin function
36 def HEM(Fluid,P_1,Cd,D_2,steps, Operating_P2):
37
38     #initilaze parameters:
39     number_graphs=len(P_1)
40     rho_2=np.zeros((number_graphs,steps)) #density
41     h_2=np.zeros((number_graphs,steps)) #enthalpy
42     mHEM=np.zeros((number_graphs,steps)) #Mass flow rate
43     mHEMchoked=np.zeros((number_graphs,steps)) #Mass flow rate when
       adjusting for choked flow
44     CriticalFlowRate=np.zeros(number_graphs) #Track value of
       critical flow rate for each graph
```

```
45      ActualFlowRate=np.zeros(number_graphs) #Track value of flow
    rate at Operating_P2 for each graph
46      A_2=(np.pi/4)*D_2**2 #injector area
47
48
49      for j in range(number_graphs):
50          #Initial calculations:
51          s_1=PropsSI('S','P',P_1[j],'Q',0,Fluid) #Using saturated
    liquid upstream (Q=0 in coolProp)
52          s_2=s_1     #follow line of constant entropy
53          h_1=PropsSI('H','P',P_1[j],'Q',0,Fluid)
54          #P_2=np.linspace(100000,P_1[j]-1,steps)
55          if Fluid=='CO2': #See text on top of code for explaination
    of if statement
56              P_2=np.linspace(5.2e5,P_1[j]-1,steps)
57          else:
58              P_2=np.linspace(1e5,P_1[j]-1,steps)
59
60          #for loop for graph that follows model exactly
61          for i in range(steps):
62              rho_2[j,i]=PropsSI('D','P',P_2[i],'S',s_2,Fluid) #use
    downstream pressure&entropy to find density
63              h_2[j,i]=PropsSI('H','P',P_2[i],'S',s_2,Fluid) #use
    downstream pressure&entropy to find enthalpy
64              mHEM[j,i]=Cd*A_2*rho_2[j,i]*np.sqrt(2*(h_1-h_2[j,i])) #
    mass flow rate
65
66          flow_tracker=mHEM[j,:]
67          criticalIndex=np.where(flow_tracker == np.max(flow_tracker)
    ) #find index location of max flow rate
68
69          #for loop for graph where the choked flow will show
70          for i in range(steps):
71              if i > criticalIndex[0]:
72                  mHEMchoked[j,i]=mHEM[j,i]
73              else:
74                  mHEMchoked[j,i]=np.max(flow_tracker)
75
76          #plot results: (can also plot flow rate vs p_2 if desired)
77          plt.plot(P_1[j]-P_2,mHEMchoked[j,:],'b', label='P1=%.2f [
    MPa]'%(P_1[j]/(1e6)))
78          plt.plot(P_1[j]-P_2, mHEM[j,:],'b--')
79          plt.title('Homogenous Equillibrium Model')
80          plt.xlabel('P1-P2 [Pa]')
81          plt.ylabel('Mass Flow Rate [kg/s]')
82          plt.grid()
83          plt.legend()
84          plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0)
    )
85          plt.show()
86
87          ActualFlowRate[j]=np.interp(Operating_P2,P_2,mHEMchoked[j
    ,:])
88          CriticalFlowRate[j]=np.max(mHEM[j,:])
89          if ActualFlowRate[j]!=CriticalFlowRate[j]:
90              print('Flow not choked')
91
92      #plot operating & critical flow rate vs p1 after all iterations
```

```
    are complete
93    plt.plot(P_1,ActualFlowRate, label='P_2=%.2f [Mpa]'%(
    Operating_P2/(1e6)))
94    plt.plot(P_1,CriticalFlowRate,'r--', label='Choked, P_2=%.2f [
    Mpa]'%(Operating_P2/(1e6)))
95    plt.title('Homogenous Equillibrim model for linear upstream
    pressure')
96    plt.xlabel('P_1 [Pa]')
97    plt.ylabel('Mass Flow Rate [kg/s]')
98    plt.grid()
99    plt.legend()
100   plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
101   plt.show()
102
103   AverageFlowRate=np.average(ActualFlowRate)
104   return CriticalFlowRate, ActualFlowRate, AverageFlowRate
```

Listing A.5: Code for Homogeneous Equilibrium Model

A.4.1 Simplified HEM for use with Dyer model

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Tue Dec  3 18:12:56 2019
4
5  Homogenous Equillibrium Model
6
7  Only one case at a time here for simplicity
8
9  ------------------------------------------------------------
10 Note that for P_2, the lower value has been set to 1 & 5.2 bar (for
       N2O and CO2 respectively) as very low values
11 caused issues with the entropy calcs.
12
13 "It should be noted that CO2, due to it's high triple point
     pressure cannot exist in
14 liquid phase at pressures below that of it's triple point pressure
     of 517.95 kPa. At any
15 pressure below this value, the CO2
16 ow would consist of solid-vapor mixture. It is for this
17 reason that CO2 exists as a "Dry Ice" at atmospheric pressure and
     temperature."
18
19 Presumably similar story for N2O. (triplepoint at 87.85kPa)
20 ------------------------------------------------------------------------
21
22 P_2 is set to go up to P1-1 as we otherwise get a division by 0
     when implemented in the Dyer model
23
24 @author: Jonas
25 """
26
27 import numpy as np
28 import matplotlib.pyplot as plt
29 from CoolProp.CoolProp import PropsSI
30
31 #input example:
32 Cd=1 #Discharge Coefficient
```

```
D_2=2e-3 #Injector Orfice Diameter
P_1=5.03e6 #Upstream pressure
Fluid='CO2'#Fluid
steps=1000 #Number of steps in iteration below


#begin function
def simpleHEM(Fluid,P_1,Cd,D_2,steps):
    #initilaze parameters:
    rho_2=np.zeros(steps) #downstream density
    h_2=np.zeros(steps) #downstream enthalpy
    if Fluid=='CO2': #See green text on top for explaination of if statement
        P_2=np.linspace(5.2e5,P_1-1,steps)
    else:
        P_2=np.linspace(1e5,P_1-1,steps)

    mHEM=np.zeros(steps)
    mHEMchoked=np.zeros(steps)

    #Initial calculations:
    A_2=(np.pi/4)*D_2**2
    s_1=PropsSI('S','P',P_1,'Q',0,Fluid) #Using saturated liquid upstream (Q=0 in coolProp)
    s_2=s_1    #follow line of constant entropy
    h_1=PropsSI('H','P',P_1,'Q',0,Fluid)

    #for loop for graph that follows model exactly
    for i in range(steps):
        rho_2[i]=PropsSI('D','P',P_2[i],'S',s_2,Fluid)
        h_2[i]=PropsSI('H','P',P_2[i],'S',s_2,Fluid)
        mHEM[i]=Cd*A_2*rho_2[i]*np.sqrt(2*(h_1-h_2[i]))

    criticalIndex=np.where(mHEM == np.max(mHEM)) #find index location of max flow rate

    #for loop for graph where the choked flow will show
    for i in range(steps):
        if i > criticalIndex[0]:
            mHEMchoked[i]=mHEM[i]
        else:
            mHEMchoked[i]=np.max(mHEM)
    return mHEM

    #plot results:
    plt.plot(P_2,mHEMchoked,'b', label='P1=%.2f [MPa]'%(P_1/(1e6)))
    plt.plot(P_2, mHEM,'b--')
    plt.title('Homogenous Equillibrium Model')
    plt.xlabel('P2 [Pa]')
    plt.ylabel('Mass Flow Rate [kg/s]')
    plt.grid()
    plt.legend()
    plt.show()
    print(np.max(mHEM))
```

Listing A.6: Code for Perfect gas model

A.5 Dyer Model

```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec  9 15:24:27 2019
Dyer Model

Made for use with N2O or CO2

This basic version is meant to be used for saturated liquid upstream, no supercharge.
To get supercharged, some changes would have to be made to SPI and HEM models, but this is not currently relevant for Propulse

Note that P_2 ranges from 1/5.2 bar. this is because the CoolProp fails in HEM model for very small values of P2.
This has to do with the triple point of the fluid, see HEM for details.

Alse note that P_2 ends at P_1 - 1 instead of P_1. This is because we otherwise divide by 0 in the calculation of k.


@author: Jonas
"""
import numpy as np
import matplotlib.pyplot as plt
from CoolProp.CoolProp import PropsSI
from SPI4Dyer_Model import SPI4Dyer_Model
from SimpleHEM import simpleHEM
from SPI_Model import SPI_Model
from HEM import HEM

#Example Input:
Cd=0.75 #Discharge Coefficient
D_2=2e-3 #Injector Orfice Diameter (should be 1-2mm)
Fluid='N2O'#Fluid - both N2O and CO2 can be used.
steps=100 #Number of steps in each iteration of mHEM & mSPI
P1=np.linspace(6e6,4e6,10) #linear upstream pressure drop from 60 bar to 40 bar
Operating_P2=3e6 #operating pressure in burnchamber (assumed constant)
# ----------------------------------------------------------------------------



FlowTracker=np.zeros(len(P1)) # tracks of flowrate at P2=Operating P2 for different P1 iterations
ChokedValue=np.zeros(len(P1)) #tracks value of critical flow for different P1 iterations

for j in range(len(P1)):
    P_1=P1[j] #Upstream pressure for this iteration
    #Get mSPI
    mSPI=SPI4Dyer_Model(Fluid,P_1,Cd,D_2,steps)
    #Get mHEM
    mHEM=simpleHEM(Fluid,P_1,Cd,D_2,steps)
```

```
46
47
48      #Initialize parameters
49      P_v=P_1 #saturated case
50      P_2=np.linspace(100000,P_1-1,steps) #matches what we used in
        HEM
51
52
53      if Fluid=='N2O':#See introduction text above for explaination
54          DeltaP=np.linspace(P_1-1e5,1,steps)
55          P_2=np.linspace(1e5,P_1-1,steps)
56      else:
57          DeltaP=np.linspace(P_1-5.2e5,1,steps)
58          P_2=np.linspace(5.2e5,P_1-1,steps)
59      mDyer=np.zeros(steps)
60      k=0
61      mDyerChoked=np.zeros(steps)
62
63      #for loop for Dyer model
64      for i in range(steps):
65          k=np.sqrt((P_1-P_2[i])/(P_v-P_2[i])) #In saturated case
        this will always be 1...
66          mDyer[i]=(((k*mSPI[i])/(1+k))+(mHEM[i]/(1+k)))
67
68      criticalIndex=np.where(mDyer== np.max(mDyer)) #index where
        critical value occurs.
69
70      ChokedValue[j]=np.max(mDyer)
71      #for loop for choked flow
72      for i in range(steps):
73              if i > criticalIndex[0]:
74                  mDyerChoked[i]=mDyer[i]
75              else:
76                  mDyerChoked[i]=np.max(mDyer)
77      """
78      plot against P2 if desired:
79
80      plt.plot(P_2,mDyerChoked,'b',label='P1=%.2f [MPa]'%(P_1/(1e6)))
81      plt.plot(P_2,mDyer,'b--')
82      plt.title('Dyer Model')
83      plt.xlabel('P2[Pa]')
84      plt.ylabel('Mass Flow Rate [kg/s]')
85      plt.grid()
86      plt.legend()
87      plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
88      plt.show()
89      print(np.max(mDyer))
90      """
91
92      plt.plot(DeltaP,mDyerChoked,'b',label='P1=%.2f [MPa]'%(P_1/(1e6
        )))
93      plt.plot(DeltaP,mDyer,'b--')
94      plt.title('Dyer Model')
95      plt.xlabel('P1-P2 [Pa]')
96      plt.ylabel('Mass Flow Rate [kg/s]')
97      plt.grid()
98      plt.legend()
99      plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
```

```
    plt.show()

    FlowTracker[j]=np.interp(Operating_P2,P_2,mDyerChoked)



plt.plot(P1,FlowTracker, label='P_2=%.2f [Mpa]'%(Operating_P2/(1e6)))
plt.plot(P1,ChokedValue, label='Choked, P_2=%.2f [Mpa]'%(Operating_P2/(1e6)))
plt.title('Dyer model with linear upstream pressure drop')
plt.xlabel('P_1 [Pa]')
plt.ylabel('Mass Flow Rate [kg/s]')
plt.grid()
plt.legend()
plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
plt.show()
AverageFlowRate=np.average(FlowTracker)
print(ChokedValue)
print(FlowTracker)
print(AverageFlowRate)
```

Listing A.7: Code for Dyer Model

A.6 Transient Equilibrium Model

```
# -*- coding: utf-8 -*-
"""
Created on Tue May  5 15:59:24 2020
Transient Equilibrium with regression rate modeling
@author: asus
"""

# -*- coding: utf-8 -*-
"""
Created on Tue Mar 31 18:05:53 2020


@author: asus
"""
import numpy as np
import matplotlib.pyplot as plt
from CoolProp.CoolProp import PropsSI
from SPI4Dyer_Model import SPI4Dyer_Model
from SimpleHEM import simpleHEM

#input:
Cd=0.75 #discharge coefficient
Fluid='N2O'
D_2=0.002 #injector orifice diameter [m]
NumOrifices=28 #number of orifices on injector plate (Few orifices will cause long run times)
steps=100 #iteration steps for mDyer (negligable difference from 100 to 1000)
timestep=0.010 #seconds for each iteration of outer while loop
V=0.03255 # tank volume [m^3]
M=20 #mass of tank contents, [kg]
T1=300.86 #Initial Upstream Temperature [K]
```

```
Operating_P2=3e6 #Operating P2
r=50e-3 #radius of fuel grain port [m]
L=0.48 #fuel grain length [m]
a=15.5e-5 #regression rate constant n2o/paraffin (from Waxman et al., adjusted so regression rate is given in m/s by multiplication with 10^-3)
n=0.5 #regression rate exponent n2o/paraffin (from Waxman et al.)
rho_f=900 #fuel density (paraffin) [kg/m^3]
#--------------------------

#initial calcs:
rho=M/V  #overall density (vapor and liquid)
u=PropsSI('U','T',T1,'D',rho,Fluid) #specific Internal energy (vapor and liquid mixture)
U=u*M #Internal energy (vapor and liquid)
#A_2=(np.pi/4)*D_2**2 #injector orifice cross-sectional area
x=(U/M - PropsSI('U','T',T1,'Q',0,Fluid))/(PropsSI('U','T',T1,'Q',1,Fluid) -PropsSI('U','T',T1,'Q',0,Fluid)) #initial vapor quality


Time=0 #start at time=0

j=0 #initialize iteration counter j for while loop
ChokedValue=np.array([]) #initialize choked flow rate tracker
FlowTracker=np.array([]) #initialize operating flow rate tracker (flow rate with Operating P_2)
TempVector=np.array([]) #initialize upstream temperature tracker
PressureVector=np.array([]) #initialize upstream pressure tracker
MassVector=np.array([]) #initialize tank mass tracker
TimeVector=np.array([]) #initialize Time tracker

#initialize regression rate, fuel flow rate vectors:
regRateVector=np.array([])
mdot_fuel_Vector=np.array([])
while x<0.95: #while vapor quality is less than 0.95, i.e. so long there is liquid in tank
    TempVector=np.insert(TempVector,j,T1) #record temperature in TempVector
    TimeVector=np.insert(TimeVector,j,Time) #record Time in TimeVector
    MassVector=np.insert(MassVector,j,M) #record mass in MassVector

    P_1=PropsSI('P','T',T1,'Q',0,Fluid)  #saturated liquid pressure
    PressureVector=np.insert(PressureVector,j,P_1) #record pressure in PressureVector

    #start solving for mDyer by finding mSPI and mHEM:
    mSPI=SPI4Dyer_Model(Fluid,P_1,Cd,D_2,steps) #SPI mass flow rate
    mHEM=simpleHEM(Fluid,P_1,Cd,D_2,steps)  #HEM mass flow rate

    #Initialize parameters
    P_2=np.linspace(100000,P_1-1,steps) #matches what we used in HEM
    DeltaP=np.linspace(P_1-100000,1,steps)
    mDyer=np.zeros(steps)
    k=1 #Always = 1 in saturated case
    mDyerChoked=np.zeros(steps)
```

```
78
79      #for loop for Dyer model
80      for i in range(steps):
81
82          mDyer[i]=(((k*mSPI[i])/(1+k))+(mHEM[i]/(1+k)))
83
84      criticalIndex=np.where(mDyer== np.max(mDyer)) #index where
    critical value occurs.
85      ChokedValue=np.insert(ChokedValue,j,np.max(mDyer)) #add the
    choked value for this j iteration to ChokedValue
86
87      #for loop to account for choked flow
88      for i in range(steps):
89              if i > criticalIndex[0]:
90                  mDyerChoked[i]=mDyer[i]
91              else:
92                  mDyerChoked[i]=np.max(mDyer)
93
94      #add the operating flow rate for this j iteration to
    FlowTracker
95      FlowTracker=np.insert(FlowTracker,j,np.interp(Operating_P2,P_2,
    mDyerChoked))
96
97
98      #Get the new values of U and M:
99      mDyerTotal=NumOrifices*FlowTracker[j]
100     h1=PropsSI('H','P',P_1,'Q',0,Fluid) #calculate enthalpy
101     U=U-mDyerTotal*h1*timestep #update internal energy (neglecting
    wall heat transfer etc.)
102     M=M-mDyerTotal*timestep #update mass
103
104     #regression rate calcs:
105     A_p=np.pi*r**2 #fuel grain port cross-sectional area
106     A_d=2*np.pi*r*L #fuel grain port surface area (area available
    to heat transfer with flame)
107     G_ox=mDyerTotal/A_p #oxidizer mass flux through port
108     regRate=a*(G_ox**(n)) #regression rate equation
109     regRateVector=np.insert(regRateVector,j,regRate) #record
    regression rate in vector
110     mdot_fuel=rho_f*A_d*regRate #calculate fuel mass flow rate
111     mdot_fuel_Vector=np.insert(mdot_fuel_Vector,j,mdot_fuel) #
    record fuel flow rate
112     r=r+regRate*timestep #update radius
113
114
115     #Start iteration for new T1, "First round" must be outside
    while loop:
116     Tguess=T1 #use previous T as an initial guess for calculating
    new T
117     u_l=PropsSI('U','T',Tguess,'Q',0,Fluid) #liquid specific
    internal energy
118     u_v=PropsSI('U','T',Tguess,'Q',1,Fluid) #vapor specific
    internal energy
119     rho_l=PropsSI('D','T',Tguess,'Q',0,Fluid) #liquid density
120     rho_v=PropsSI('D','T',Tguess,'Q',1,Fluid) #vapor density
121     x=((U/M)-u_l)/(u_v-u_l)  #vapor quality
122     Vguess=M*(((1-x)/rho_l)+x/rho_v) #volume constraint
123
```

```
124     error=V-Vguess
125     oldError=1000 #chose some random large value, to force into
    oldError>error case for first iteration below....
126     #...as we expect T1 to drop
127 
128 
129     while error>0.000001: #Set small enough so that no issues of
    T_1 not updating
130         if oldError >error:
131             oldError=error #prepare the current error to be next
    iteration's oldError
132             Tguess=Tguess-0.01 #- update Tguess.
133             u_l=PropsSI('U','T',Tguess,'Q',0,Fluid) #liquid
    specific internal energy
134             u_v=PropsSI('U','T',Tguess,'Q',1,Fluid) #vapor specific
     internal energy
135             rho_l=PropsSI('D','T',Tguess,'Q',0,Fluid) #liquid
    density
136             rho_v=PropsSI('D','T',Tguess,'Q',1,Fluid) #vapor
    density
137             x=((U/M)-u_l)/(u_v-u_l)  #vapor quality
138             Vguess=M*(((1-x)/rho_l)+x/rho_v) #volume constraint
139 
140             error=V-Vguess#update error
141 
142         else:
143             oldError=error #prepare the current error to be next
    iteration's oldError
144             Tguess=Tguess+0.004 #update Tguess the other way, as
    our error has increased .
145             u_l=PropsSI('U','T',Tguess,'Q',0,Fluid) #liquid
    specific internal energy
146             u_v=PropsSI('U','T',Tguess,'Q',1,Fluid) #vapor specific
     internal energy
147             rho_l=PropsSI('D','T',Tguess,'Q',0,Fluid) #liquid
    density
148             rho_v=PropsSI('D','T',Tguess,'Q',1,Fluid) #vapor
    density
149             x=((U/M)-u_l)/(u_v-u_l)  #vapor quality
150             Vguess=M*(((1-x)/rho_l)+x/rho_v) #volume constraint
151 
152             error=V-Vguess#update error
153 
154     T1=Tguess #update T1 for next iteration
155 
156     #update vapor quality:
157     u_l=PropsSI('U','T',T1,'Q',0,Fluid) #liquid specific internal
    energy
158     u_v=PropsSI('U','T',T1,'Q',1,Fluid) #vapor specific internal
    energy
159     rho_l=PropsSI('D','T',T1,'Q',0,Fluid) #liquid density
160     rho_v=PropsSI('D','T',T1,'Q',1,Fluid) #vapor density
161     x=((U/M)-u_l)/(u_v-u_l)  #vapor quality te vapor quality
162 
163     Time=Time+timestep #update Time
164     j=j+1 #update iteration counter j
165 
166 #Remember to include last iteration into our tracking vectors:
```

```
167 TempVector=np.insert(TempVector,j,T1) #record final temperature in
    TempVector
168 TimeVector=np.insert(TimeVector,j,Time) #record final Time in
    TimeVector
169 MassVector=np.insert(MassVector,j,M) #record final mass in
    MassVector
170 P_1=PropsSI('P','T',T1,'Q',0,Fluid) #record final pressure in
    PressureVector:
171 PressureVector=np.insert(PressureVector,j,P_1)
172
173 OFratioVector=(NumOrifices*FlowTracker)/mdot_fuel_Vector
174
175
176 #Plot flow rate vs time, all orifices combined:
177 plt.plot(TimeVector[:-1],NumOrifices*FlowTracker, label='P_2=%.2f [
    Mpa]'%(Operating_P2/(1e6)))
178 plt.plot(TimeVector[:-1],NumOrifices*ChokedValue, label='Choked,
    P_2=%.2f [Mpa]'%(Operating_P2/(1e6)))
179 plt.title('Dyer mass flow rate time history for Transient
    Equilibrium Model')
180 plt.xlabel('Time [s]')
181 plt.ylabel('Mass Flow Rate [kg/s]')
182 plt.grid()
183 plt.legend()
184 plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
185 plt.show()
186
187 #Plot flow rate vs upstream pressure (for comparisons etc.)
188 plt.figure
189 plt.plot(PressureVector[:-1],FlowTracker, label='P_2=%.2f [Mpa]'%(
    Operating_P2/(1e6)))
190 plt.plot(PressureVector[:-1],ChokedValue, label='Choked, P_2=%.2f [
    Mpa]'%(Operating_P2/(1e6)))
191 plt.title('Dyer mass flow rate plotted against P1 for Transient
    Equilibrium Model')
192 plt.xlabel('P_1 [Pa]')
193 plt.ylabel('Mass Flow Rate [kg/s]')
194 plt.grid()
195 plt.legend()
196 plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
197 plt.show()
198
199 #plot Pressure vs time
200 plt.figure
201 plt.plot(TimeVector,PressureVector, label='P_2=%.2f [Mpa]'%(
    Operating_P2/(1e6)))
202 plt.title('Pressure time history for Transient equillibrium model')
203 plt.ylabel('P_1 [Pa]')
204 plt.xlabel('Time [s]')
205 plt.grid()
206 plt.legend()
207 plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
208 plt.show()
209
210 #plot Temperature vs time
211 plt.figure
212 plt.plot(TimeVector,TempVector, label='P_2=%.2f [Mpa]'%(
    Operating_P2/(1e6)))
```

```
plt.title('Temperature time history for Transient equillibrium model')
plt.ylabel('T_1 [K]')
plt.xlabel('Time [s]')
plt.grid()
plt.legend()
plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
plt.show()
#plot mass vs time
plt.figure
plt.plot(TimeVector,MassVector, label='P_2=%.2f [Mpa]'%(Operating_P2/(1e6)))
plt.title('Mass plotted against time using Transient equillibrium model')
plt.xlabel('Time [s]')
plt.ylabel('Mass [kg]')
plt.grid()
plt.legend()
plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
plt.show()

#plot regression rate vs time:
plt.figure
plt.plot(TimeVector[:-1],regRateVector, label='P_2=%.2f [Mpa]'%(Operating_P2/(1e6)))
plt.title('Regression rate plotted against time using Transient equillibrium model')
plt.xlabel('Time [s]')
plt.ylabel('Regression rate [m/s]')
plt.grid()
plt.legend()
plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
plt.show()

#plot fuel flow rate vs time:
plt.figure
plt.plot(TimeVector[:-1],mdot_fuel_Vector, label='P_2=%.2f [Mpa]'%(Operating_P2/(1e6)))
plt.title('Fuel mass flow rate plotted against time using Transient equillibrium model')
plt.xlabel('Time [s]')
plt.ylabel('Fuel mass flow rate [kg/s]')
plt.grid()
plt.legend()
plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
plt.show()

#plot O/F ratio vs time:
plt.figure
plt.plot(TimeVector[:-1],OFratioVector, label='P_2=%.2f [Mpa]'%(Operating_P2/(1e6)))
plt.title('O/F ratio plotted against time using Transient equillibrium model')
plt.xlabel('Time [s]')
plt.ylabel('O/F ratio')
plt.grid()
plt.legend()
plt.ticklabel_format(axis="x", style="sci", scilimits=(0,0))
```

```
plt.show()

#Calculate and print average flow rate , regression rate
AverageFlowRate=np.average(FlowTracker) #single orifice
TotalAverageFlowRate=NumOrifices*AverageFlowRate #all orifices
    combined
AvgFuelFlowrate=np.average(mdot_fuel_Vector) #fuel flow rate
AvgRegrate=np.average(regRateVector) #average regression rate
print('the average oxidizer flow rate is [per orifice, total]:')
print(AverageFlowRate,TotalAverageFlowRate)
print('The average fuel mass flow rate is:')
print(AvgFuelFlowrate)
print('the average regression rate is:')
print(AvgRegrate)
```

Listing A.8: Code for Transient Equilibrium Model

Bibliography

[1] Brian J. Cantwell. *Aircraft and Rocket Propulsion*. 2010.

[2] Hammad Shah et al. "Design and Prototyping of a Solid Fuel/Gaseous Oxidizer Hybrid Rocket Engine". PhD thesis. Dec. 2016. DOI: 10.13140/RG.2.2.12066.22728.

[3] James L. Cannon. "Liquid Propulsion: Propellant Feed System Design". In: *Encyclopedia of Aerospace Engineering* 2 (Jan. 2010). DOI: 10.1002/9780470686652.eae110. URL: https://ntrs.nasa.gov/search.jsp?R=20100035254.

[4] Ralph Ewig. *Vapor Pressurization (VaPak) Systems History, Concepts, and Applications*. 2009.

[5] Iain Waugh et al. *Testing of a novel nitrous-oxide and ethanol fuel blend*. May 2018.

[6] G.P. Sutton and O. Biblarz. *Rocket Propulsion Elements*. John Wiley & Sons, 2010. ISBN: 9780470080245. URL: https://books.google.no/books?id=1Sf6eV6CgtEC.

[7] Arif M. Karabeyoglu and Brian J. Evans. "Effect of "O/F Shift" on Combustion Efficiency". In: *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. DOI: 10.2514/6.2014-3851. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2014-3851. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2014-3851.

[8] B. J. Cantwell. "Similarity solution of fuel mass transfer, port mass flux coupling in hybrid propulsion". In: *Journal of Engineering Mathematics* 84.1 (2013). DOI: 10.1007/s10665-013-9624-y. URL: http://link.springer.com/content/pdf/10.1007/s10665-013-9624-y.pdf#page-1.

[9] G. Marxman and M. Gilbert. "Turbulent boundary layer combustion in the hybrid rocket". In: *Symposium (International) on Combustion* 9.1 (1963), pp. 371–383. ISSN: 0082-0784. DOI: https://doi.org/10.1016/S0082-0784(63)80046-6. URL: http://www.sciencedirect.com/science/article/pii/S0082078463800466.

[10] Gregory Zilliac and M. Karabeyoglu. "Hybrid Rocket Fuel Regression Rate Data and Modeling". In: *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference Exhibit* (Sept. 2006). DOI: 10.2514/6.2006-4504.

[11] Arif Karabeyoglu, Brian Cantwell, and Greg Zilliac. "Development of Scalable Space-Time Averaged Regression Rate Expressions for Hybrid Rockets". In: *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference Exhibit* (Oct. 2005). DOI: 10.2514/6.2005-3544.

[12] Benjamin S. Waxman et al. "An investigation of injectors for use with high vapor pressure propellants with applications to hybrid rockets". PhD thesis. 2014. URL: http://purl.stanford.edu/ng346xh6244.

[13] Mohammad Mahdi Heydari and Nooredin Ghadiri Massoom. "Experimental Study of the Swirling Oxidizer Flow in HTPB/N2O Hybrid Rocket Motor". In: *International Journal of Aerospace Engineering* 2017 (2017), pp. 1–10. DOI: 10.1155/2017/3174140. URL: https://www.hindawi.com/journals/ijae/2017/3174140/cta/.

[14] Byeonguk Ahn et al. "Design of Multiport Grain with Hydrogen Peroxide Hybrid Rocket". In: *Journal of Propulsion and Power* 34.5 (2018), pp. 1189–1197. DOI: 10.2514/1.b36949.

[15] Dario Pastrone. "Approaches to Low Fuel Regression Rate in Hybrid Rocket Engines". In: *International Journal of Aerospace Engineering* 2012 (2012), pp. 1–12. DOI: 10.1155/2012/649753.

[16] Brian Cantwell, Arif Karabeyoglu, and David Altman. "Recent Advances In Hybrid Propulsion". In: *International Journal of Energetic Materials and Chemical Propulsion* 9.4 (2010), pp. 305–326. DOI: 10.1615/intjenergeticmaterialschempr v9.i4.20.

[17] M. A. Karabeyoglu, D. Altman, and B. J. Cantwell. "Combustion of Liquefying Hybrid Propellants: Part 1, General Theory". In: *Journal of Propulsion and Power* 18.3 (2002), pp. 610–620. DOI: 10.2514/2.5975.

[18] Jonah Zimmerman, Brian Cantwell, and Gregory Zilliac. "Initial Experimental Investigations of Self-Pressurizing Propellant Dynamics". In: *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit 2012* (July 2012). DOI: 10.2514/6.2012-4198.

[19] Air Liquide Group. *Physical properties of N_2O*. 2020. URL: https://encyclopedia.airliquide.com/nitrous-oxide.

[20] Douglas E. Raynie. "Warning concerning the use of nitrous oxide in supercritical fluid extractions". In: *Analytical Chemistry* 65.21 (1993), pp. 3127–3128. DOI: 10.1021/ac00069a028.

[21] Arif Karabeyoglu et al. "Modeling of N2O Decomposition Events". In: *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference Exhibit* (2008). DOI: 10.2514/6.2008-4933.

[22] John Campbell, Frank Macklin, and Zachary Thicksten. "Handling Considerations of Nitrous Oxide in Hybrid Rocket Motor Testing". In: *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference Exhibit* (2008). DOI: 10.2514/6.2008-4830.

[23] Fiona K. Leverone et al. "Performance sensitivity study on a blowdown nitrous oxide paraffin wax hybrid sounding rocket". In: *Acta Astronautica* 160 (Apr. 2019). DOI: 10.1016/j.actaastro.2019.04.043.

[24] Marco Invigorito, Gianpaolo Elia, and Mario Panelli. "Numerical Modeling of Self-Pressurizing Oxidizers for Hybrid Rocket Injection". In: *6TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)* (June 2015).

[25] Neuterium.net. *Choked Flow*. Apr. 2015. URL: https://neutrium.net/fluid_flow/choked-flow/.

[26] E. Gamper and R. Hink. "Design and Test of Nitrous Oxide Injectors for a Hybrid Rocket Engine". In: *Deutscher Luft- und Raumfahrtkongress* (2013).

[27] Benjamin Waxman, Brian Cantwell, and Greg Zilliac. "Effects of Injector Design and Impingement Techniques on the Atomization of Self-Pressurizing Oxidizers". In: *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference Exhibit* (2012). DOI: 10.2514/6.2012-3906.

[28] Pasquale M. Sforza. "Chapter 11 - Liquid Rockets". In: *Theory of Aerospace Propulsion*. Butterworth-Heinemann, 2012, pp. 439–482. ISBN: 978-1-85617-912-6. DOI: https://doi.org/10.1016/B978-1-85617-912-6.00011-6. URL: http://www.sciencedirect.com/science/article/pii/B9781856179126000116.

[29] Mehdi Mojtabi, Graham Wigley, and Jerome Helie. "The Effect Of Flash Boiling On The Atomization Performance Of Gasoline Direct Injection Multistream Injectors". In: *Atomization and Sprays* 24.6 (2014), pp. 467–493. DOI: 10.1615/atomizspr.2014008296.

[30] M. Bouziane et al. "Performance comparison of oxidizer injectors in a 1-kN paraffin-fueled hybrid rocket motor". In: *Aerospace Science and Technology* 89 (2019), pp. 392–406. DOI: 10.1016/j.ast.2019.04.009.

[31] Anlong Yang et al. "Periodic atomization characteristics of an impinging jet injector element modulated by Klystron effect". In: *Chinese Journal of Aeronautics* 31.10 (2018), pp. 1973–1984. DOI: 10.1016/j.cja.2018.07.011. URL: https://www.sciencedirect.com/science/article/pii/S1000936118302590.

[32] Kevin W. Brinckman, Gregory Feldman, and Ashvin Hosangadi. "Impinging Fuel Injector Atomization and Combustion Modeling". In: *51st AIAA/SAE/ASEE Joint Propulsion Conference* (2015). DOI: 10.2514/6.2015-3763.

[33] G. Bailardi, M. Negri, and H.K. Ciezki. "Several Aspects of the Atomization Behavior of Various Newtonian Fluids with a like-on-like Impinging Jet Injector". In: *ILASS – Europe 2010, 23rd Annual Conference on Liquid Atomization and Spray Systems* (Sept. 2010).

[34] Marcus F. Heidmann and Hampton H. Foster. "Effect of Impingement Angle on Drop-Size Distribution and Spray Pattern of Two Impinging Water Jets". In: *NASA Technical Note D-872* (July 1961).

[35] Ned P. Hannum et al. "SOME INJECTOR ELEMENT DETAIL EFFECTS ON SCREECH IN HYDROGEN-OXYGEN ROCKETS". In: *NASA Technical Memorandum X-2982*. Lewis Research Center, Feb. 1974.

[36] Clement Indiana et al. "Combustion of sprays from triplet injector with green propellants: ethyl alcohol and hydrogen peroxide". In: *7TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)* (2015). DOI: 10.13009/EUCASS2017-233.

[37] K. K. Kuo et al. "DENSE SPRAY AND MIXING OF LIQUID JETS EMANATING FROM DOUBLET INJECTORS". In: *NASA-CR-191222*. 1992.

[38] Mohammed Bouziane et al. "Design and Experimental Evaluation of Liquid Oxidizer Injection System for Hybrid Rocket Motors". In: *7TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)* (2017). DOI: 10.13009/EUCASS2017-133.

[39] Susane Ribeiro Gomes, Leopoldo Rocco Junior, and José Atílio Fritz Fidel Rocco. "Swirl Injection Effects on Hybrid Rocket Motors". In: *Journal of Aerospace Technology and Management* 7.4 (Aug. 2015), pp. 418–424. DOI: 10.5028/jatm.v7i4.368.

[40] Matthew H. Summers, Jacob D. Dennis, and James K. Villarreal. "Small-Scale Hybrid Rocket Test Stand Characterization of Swirl Injectors". In: *49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference* (Dec. 2013). DOI: 10.2514/6.2013-3831.

[41] Zhongtao Kang et al. "Review on pressure swirl injector in liquid rocket engine". In: *Acta Astronautica* 145 (2018), pp. 174–198. DOI: 10.1016/j.actaastro.2017.12.038.

[42] Jonah E. Zimmerman et al. "Review and Evaluation of Models for Self-Pressurizing Propellant Tank Dynamics". In: *49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference* (Dec. 2013). DOI: 10.2514/6.2013-4045.

[43] Ian H. Bell et al. "Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp". In: *Industrial & Engineering Chemistry Research* 53.6 (2014), pp. 2498–2508. DOI: 10.1021/ie4033999. eprint: http://pubs.acs.org/doi/pdf/10.1021/ie4033999. URL: http://pubs.acs.org/doi/abs/10.1021/ie4033999.

[44] Frank M. White. *Fluid mechanics*. 6th ed. McGraw-Hill, 2008. ISBN: 978-0072938449.

[45] Jonny Dyer et al. "Modeling Feed System Flow Physics for Self-Pressurizing Propellants". In: *43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference Exhibit* (2007).

[46] Brian J. Solomon. "Engineering Model to Calculate Mass Flow Rate of a Two-Phase Saturated Fluid Through An Injector Orifice". In: *All Graduate Plan B and other Reports. 110.* (2011). URL: https://digitalcommons.usu.edu/gradreports/110.

[47] *Engineering ToolBox, (2018). Metric Bolts - Minimum Ultimate Tensile and Proof Loads. [online].* URL: https://www.engineeringtoolbox.com/metric-bolts-minimum-ultimate-tensile-proof-loads-d_2026.html.

Jonas Sømod Ahmed

An Investigation into Hybrid Rocket Injectors

NTNU
Norwegian University of
Science and Technology

PROPULSE NTNU