Design of a Thrust-Vectoring Electric Propulsion Thruster

a project presented to
The Faculty of the Department of Aerospace Engineering
San Jose State University

In partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by

Devin J. Jordao

December 2024

approved by
Dr. Yawo Ezunkpe
Faculty Advisor

San José State
UNIVERSITY

ABSTRACT

Design of a Thrust-Vectoring Electric Propulsion Thruster

Devin J. Jordao

With the ambitious goal of landing humans on Mars within the next few decades, electric propulsion emerges as a crucial technology, but requiring further development to become a viable solution for interplanetary travel. This report will focus on designing the mechanism for a thrust-vectoring spacecraft thruster, along with the development of the electric propulsion thruster itself. A comprehensive comparison study will be conducted to identify the most suitable type of electric propulsion for the mission. Once selected, the thruster will be designed to optimize performance based on key parameters such as electric and magnetic field interactions, thrust generation, specific impulse, efficiency, and power requirements.

Acknowledgements

First and foremost, I extend my deepest appreciation to my advisor, Dr. Yawo Ezunkpe, for guiding me through this project and by having an open-door policy. I would also like to thank my parents, sisters, and girlfriend, for pushing me to keep pursuing my dream of becoming an aerospace engineer. Finally, I would like to thank my friends that I made in the department that motivated me and helped me to keep it all together.

Table of Contents

List of Figures

List of Tables

Symbols

Symbol	Definition	Units
M	Mass	kg
m_s	Mass of Spacecraft	kg
m_p	Mass of Fuel	kg
T	Thrust	N
v_e	Exhaust Velocity	m/s
t	Time	s
Δv	Delta-v	m/s
I_{sp}	Specific Impulse	s
g	Gravitational Acceleration	m/s^2
q	Charge	C (Coulomb)
V	Voltage	V (Volts)
σ	Surface Charge Density	C/m^2
ε_0	Permittivity of Free Space	Farad/m
E	Electric Field	V/m
F	Force	N
F'	Force Per Unit Area	N/m^2
B	Magnetic Field	T (Tesla)
I_b	Beam Current	A (Amperes)
n	Number of Particles	—

v_i	Ion Velocity	m/s
∇p	Pressure Gradient	N/m^3
e	Elementary Charge	C
P_{ie}	Ion and Electron Friction Force	N/m^3
ρ	Density	kg/m^3
γ	Proportionality Constant	unitless
J	Current Density	A/m^2
V_b	Beam Voltage	V
$\dot{m}_p$	Particle Mass Flow Rate	kg/s
α	Correction Factor	—
F_t	Beam Divergence	—
θ_d	Divergence Angle	deg
T_m	Sum of Thrust of All Species	N
T'	Thrust in One Direction	N
η_m	Mass Utilization Efficiency	—
η_e	Electrical Efficiency	—
η_T	Total Efficiency	—
P_T	Total Input Power	Watts
P_b	Beam Power	Watts
P_o	Other Input Power	Watts

P_d	Power Needed to Create Ions	W
P_{in}	Input Power Needed for All Components	W
$P_{dissipated}$	Dissipated Power	W
Q	Particle Flow Rate for Propellant	particles/s
B_r	Remanent Magnetic Field	T

Chapter 1: Introduction

1.1 Motivation

Electric Propulsion is a type of propulsion system for spacecraft that differs from chemical propulsion systems. Electric propulsion is believed to be significantly more efficient due to its higher specific impulse. Some electric propulsion systems can range between 2000 seconds of specific impulse and around 19,000 seconds and their ability to generate more thrust per unit of propellant [1][2]. Despite its efficiency, electric propulsion produces a minimal amount of thrust in comparison to its chemical counterparts. For example, electric propulsion systems typically yield approximately 0.0024 N of thrust, while another might reach 2.5 N of thrust [1][2]. To compare this to chemical propulsion, the SpaceX Merlin rocket engine can produce 845 kN of thrust at sea level [3]. That is 383,000 times that amount of thrust being produced compared to a high thrust electric propulsion system. Therefore, electric propulsion lacks the power to launch rockets into Earth's orbit, but it is suitable for propelling objects that have already departed the Earth's atmosphere.

Various electric propulsion systems have distinct advantages and disadvantages, and understanding them is crucial when one considers their use for a specific mission. While one option may be better altogether, the current technological limitations may prevent it from practical implementations that are not available to make it the best option currently. In such cases, it is worth exploring the development of prerequisite technologies. Nonetheless, if the technology is not there then maybe there is the option to explore specifically how to get to that technology. It is also possible that exploring these three types of propulsion systems may lead to the discovery or development of an entirely new propulsion method.

1.2 Literature Review

1.2.1 Gridded Ion Thrusters

Electrostatic propulsion is a type of electric spacecraft propulsion that uses electromagnetic fields to propel ions from a gas to high velocities through the back of a spacecraft to produce thrust [4]. Gridded Ion thrusters are the simplest type of electrostatic propulsion system, where the ions are accelerated by an electromagnetic field and then expelled from the back, through a grid. After they get pushed out of the spacecraft, they are then neutralized to prevent the ions from counteracting the thrust they produce, otherwise they would be attracted back towards the spacecraft [5]. In Figure 1, there is a simple diagram of the necessary components of an ion thruster. The ion thruster consists of some type of ion source, typically this source is Xenon gas because of its high density. This gas is split into ions by having electrons strike the gas, making them positively charged. These ions are then accelerated towards an accelerating electrode to speed them up. Finally, once the ions pass the accelerating electrode, they are then neutralized by having electrons shot into the beam, resulting in the ions containing a neutral charge [5]. While ion thrusters are on the simpler side, and not too expensive to create, they are some of the weakest electrical propulsion systems. They produce orders of magnitude less thrust than some of the better electrical propulsion options. For example, an ion engine typically creates between 1-10 mN of thrust, which can cause a spacecraft to take days to reach highway speeds of a car [1]. While this is the case, if someone has the time for a spacecraft to get to high speeds, then this is one of the best options for propulsion as they are some of the most efficient systems. The exit velocity of this type of propulsion system can be defined, as seen in Eq. (1.1). v_e is the exit velocity, $\frac{q}{M}$ is the charge-to-mass ratio, and V is the applied voltage.

$$v_e = \left[\frac{2 \cdot q \cdot V}{M}\right]^{1/2} \tag{1.1}$$

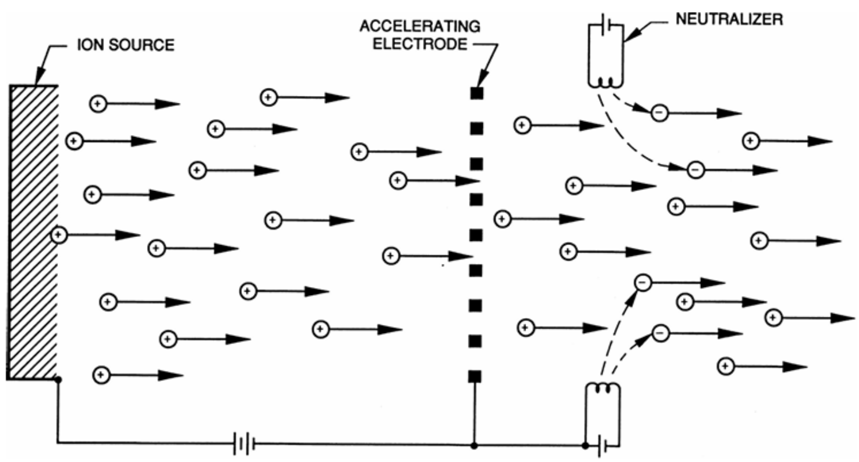

Figure 1.1 - Schematic of an ion thruster [5]

1.2.2 Hall-Effect Thrusters

Hall-effect thrusters are another type of electrostatic propulsion, but this type is different compared to gridded ion thrusters because, instead of having an electrostatic grid, Hall-effect thrusters use the Hall effect to accelerate electrons [5]. In Figure 1.2, a diagram of a Hall-effect thruster can be seen. The thruster consists of magnets at the back, near the exit, so when electrons are ejected out of the cathode neutralizer, some of these electrons get trapped inside of the magnetic field created by these magnets. These electrons that are trapped start spinning around at extremely high speeds. During this time, the anodes release gas, such as Xenon (Xe), which hits the trapped electrons, forcing electrons from the Xenon to become knocked loose. This converts the Xenon into positively charged ions and propels the ions out of the exhaust at high speeds. Finally, like a gridded ion thruster, these ions need to become neutralized, and this happens from the ions interacting with the electrons being released from the cathode neutralizer [5]. Most hall-effect thrusters produce a slightly larger amount of thrust, compared to gridded ion thrusters. The amount of thrust these thrusters typically produce is 10-50 mN of thrust [1]. This amount of thrust is seen in small satellites, such as CubeSats. While this is a significant improvement, it is still a small amount of thrust; however, researchers at the University of Michigan, along with NASA and Air Force, have designed a Hall-effect thruster capable of 5.4 Newtons of thrust. This is the largest amount of thrust ever produced by a plasma thruster [6]. This is 108 times the amount of thrust previously mentioned for these types of thrusters. The cause for the significant increase is, instead of a small form factor for small satellites that can usually be carried by one person, the Hall-effect thruster produced at the University of Michigan is 500 pounds and is nearly a full meter in diameter [6]. While it is large in terms of other versions, this version would be used on spacecraft to reach Mars, not just satellites in Earth's orbit.

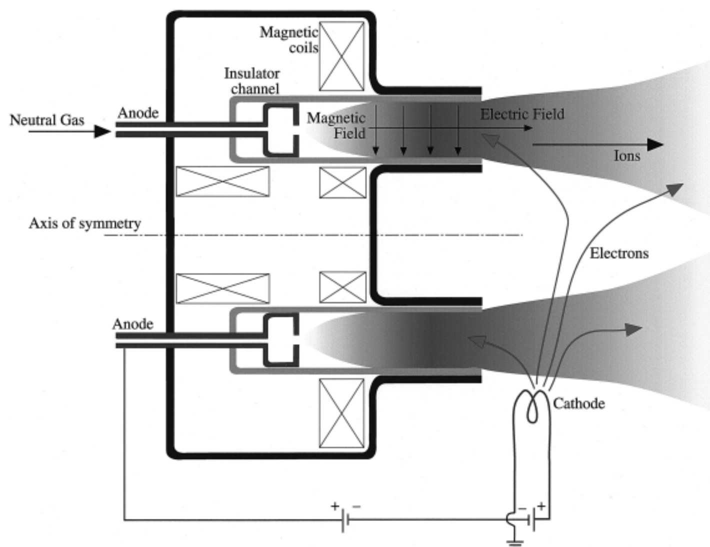

Figure 1.2 - Schematic of a Hall-effect thruster [5]

1.2.3 Magnetoplasmadynamic Thrusters

The final type of electric thruster that will be researched is the magnetoplasmadynamic (MPD) thruster. Unlike the other two thrusters, which were electrostatic thrusters, the magnetoplasmadynamic thruster is a type of electromagnetic propulsion. By looking at Figure 1.3, a better understanding can be achieved for what is needed to create a magnetoplasmadynamic thruster. The main two bodies of this thruster are the cathode and anode. The cathode is a rod in the middle of the anode, a cylindrical body that encapsulates it. There is an insulator that keeps the cathode and anode from physically touching, and it also contains small holes to allow for the propellant to be ejected through, into the circuit [5][7]. Magnetoplasmadynamic thrusters have an advantage over the other two thrusters, and that is the amount of thrust they can produce. Some of the theoretical statistics of an MPD thruster is that it can produce between 2.5 and 25 Newtons of thrust, have an exit velocity that can go up to 60 kilometers per second, and have a high efficiency that can range from 40 to 60 percent [7]. This is also the performance characteristic of a small MPD thruster, so the thrust can be increased as the size of the system gets larger. There is a large drawback to these thrusters that makes them obsolete, and that is the amount of energy they require to operate. The required energy needed can only be made possible by a nuclear reactor, as of right now. NASA, along with the Department of Energy and the reactor program of the Navy, were given the task to create such an energy source back in 2003, called Project Prometheus [8]. Due to foreseen budget constraints, the program was canceled two years after it began by NASA. Luckily, there is a program, called the Demonstration Rocket for Agile Cislunar Operations (DRACO), which is a propulsion system that uses nuclear thermal propulsion. The requirement for this program is to create a nuclear reactor capable of heating hydrogen propellant, and then release the heated gas, creating thrust [9]. While this is a different type of propulsion compared to electromagnetic propulsion that an MPD is based on, the reactor that is created could be a contender for the power system onboard a spacecraft with a magnetoplasmadynamic thruster.

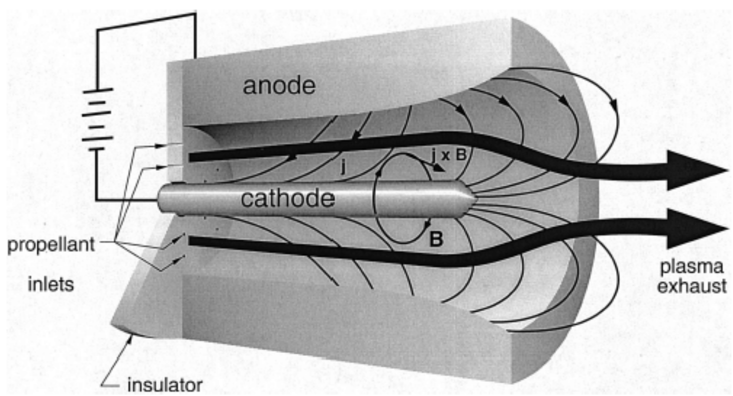

Figure 1.3 - Schematic of a magnetoplasmadynamic thruster [5]

1.2.4 Thrust Vectoring

1.2.4.1 Thrust Vectoring Background

Thrust vectoring is the concept of gimbaling the thrust of an aircraft or spacecraft, allowing for the vehicle to alter its attitude without the need for control surfaces. This is especially useful when the option for control surfaces is not even an option, like a ballistic missile that accesses high altitudes where there is low air density, or spacecrafts that spend most of their operational life outside of the atmosphere. Thrust vectoring is nothing new in the technological world, as aircraft and spacecrafts have been using them for decades, such as the F-18 fighter jet and the Space Shuttles. The F-18 needed them for improved maneuverability, but because the engine of an aircraft runs along its airframe, only the nozzle was able to be moved; For the Space Shuttles, the RS-25 engines gimbaled as a full assembly and was needed for the launch because of the weight distribution and aerodynamic drag induced by the asymmetrical shape [10][11]. An example of a thrust vectoring nozzle mechanism can be seen below, in Figure 1.4.

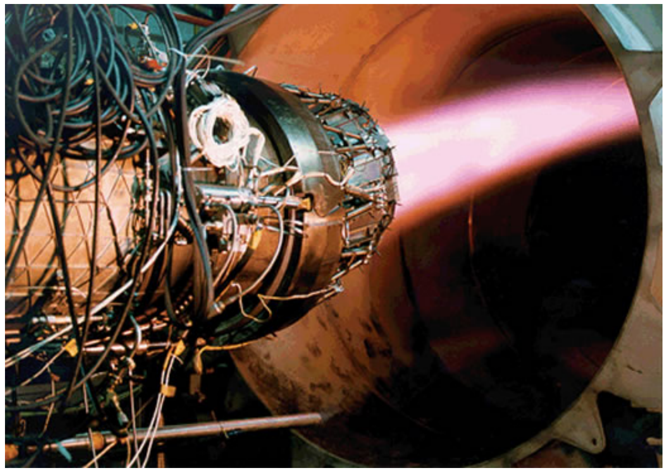

Figure 1.4 - Thrust vectoring nozzle [18]

Within the last decade, SpaceX has shifted the public eye on what a reusable rocket should be, and they have done so with their self-landing boosters and rockets, which is only possible because of thrust vectoring. Their most successful rocket, which is the most reused rocket overall, the Falcon 9, has had a total of 345 launches, 302 landings, and 276 re-flights since its first launch in 2010 [3]. SpaceX's most ambitious vehicle, the Starship, is their method of choice for transporting humans and resources to Mars. The Starship as well as its booster, the Super Heavy, are both capable of reusability, and both feature thrust vectoring paired with control surfaces to accomplish this goal [12]. As of writing this report, the Starship had its fourth and most successful flight, where the Starship was nearly fully recoverable. While there is almost no doubt that the Starship will be the main Mars transportation vehicle, there is certainly room for improvement when it comes to decreasing the cost per launch; This comes in the form of electric propulsion.

There have already been a few space missions involving thrust vectoring electric propulsion systems. An example of a mission like this was the New Millennium Program, with the Deep Space 1 spacecraft, which can be seen in Figure 1.5 [13]. The goal of the spacecraft was to test a dozen technologies to be used in future space missions, with one of the main focuses being on the electric propulsion system. The spacecraft had two types of attitude thrusters onboard, one being thrust vectoring control (TVC) for the main ion propulsion system, and the other being an attitude control system (ACS) that involved twelve thrusters. During its operation, it found that the ACS system was better for rapid changes in attitude, while the TVC was better for a stable flight. The stableness of the TVC also made it easier on the optical navigation system that was on the spacecraft [13].

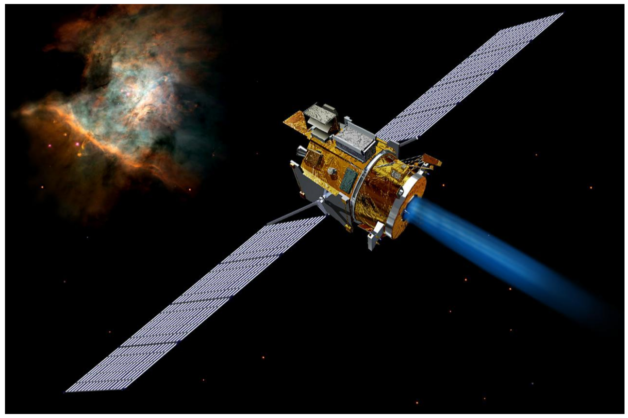

Figure 1.5 - Deep Space 1 spacecraft [19]

1.2.4.2 Thrust Vectoring Control Systems

Control systems in general are a combination of hardware and software that is used to control some type of system. There are open-loop systems, which don't have feedback, and there are closed-loop systems, that have a feedback loop. For the system in this report, a closed-loop control system will be used. A simple example of a closed-loop control system for an aircraft can be seen in Figure 1.6. The purpose of a control system for an aircraft is to dampen the pilot input, allow for autopilot systems, and to increase the performance of the aircraft [14]. A more complex controller might include gains for difference sensors, saturation limits of the actuators, a second feedback loop to analyze position, and a second transfer for the actuators used on the aircraft. When it comes to a closed-loop TVC system, the controller does not look too much different, and usually has only a few different components, such as two outputs, and a different type of controller block. Another option would be to have an adaptive controller. The benefits of this have been seen to be a decrease in settling time, as well as lower overshoot of the system, compared to a typical controller [15]. NASA conducted their own experiments for thrust vectoring. One of those focuses for this specific research was incorporating a control for the throat area and exit area. The results showed that shifting the throat gave the most efficient thrust; however, larger thrust vector angles were achieved with shock vector control [16].

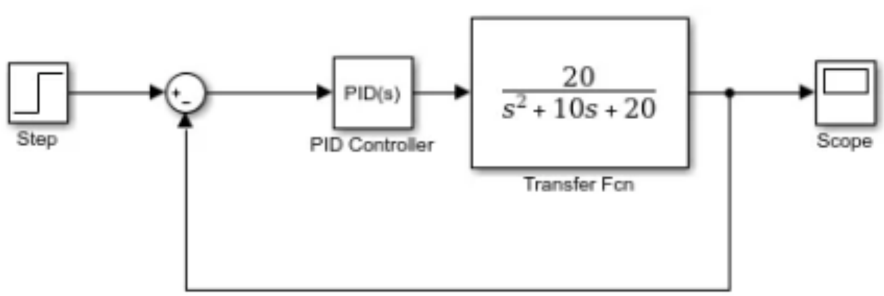

Figure 1.6 - Simple PID controller [20]

1.3 Proposal

The objective of this report is to design a type of electric propulsion that can be implemented with thrust vectoring. The electric propulsion type will be chosen based on the mission, and the thrust vector control system will be chosen based on the electric propulsion type. Once each of these specifics are decided, a model will be created for the engine and a TVC system will be designed to accommodate this engine. Finally, the model will go through extensive electrostatic analysis in order to analyze the performance and find areas of improvement.

The primary application for a specific mission could be for a small satellite, such as a CubeSat, which is a small, cube-shaped satellite. Considering that this satellite could have an ambitious mission of changing its orbit, or to eventually reach Mars, an option could be to include an electric propulsion system on board to help propel it where it needs to go. Due to the limited space on the CubeSat, instead of reaction wheels, the electric propulsion system could also gimbal its thrusters to change the direction of the satellite and aim itself accordingly. A possible continuation of the project could be to design a control system for the gimbaling mechanism onboard.

1.4 Methodology

This report will have two major phases. The first phase is to do a preliminary design. This involves a comparative study between the three electric propulsion systems discussed in the literature review. A legacy system for each of the propulsion systems will be chosen and analyzed. One for a gridded ion thruster, one for a hall-effect thruster, and one for a magnetoplasmadynamic thruster. After a mission design has been determined, one of the three thruster types will be chosen based on the mission.

The second phase is the design and analysis phase. During this phase, a design of an electric propulsion system will be completed, along with a potential TVC system that works best for the chosen engine. Physical hardware implementation is not a requirement for this project. The next step will be obtaining results for the design. The results will go over a cost breakdown, thrust-to-weight ratio, and efficiency.

Chapter 2: Comparison Study

In order to understand the best type of electric propulsion system to use for the system, a comparison is needed. The mission is to create an efficient form of propulsion that can travel from Earth to Mars, without hard time constraints. What this means is that the engine does not need to produce a large amount of thrust, but to create a reasonable amount of thrust while having a large specific impulse and high delta-v.

2.1 General Performance

When it comes to electric propulsion, the same principles that are used in chemical rockets are also used in electric thrusters. However, due to charged particles being the source of fuel for electric propulsion, the performance parameters are calculated differently for things, such as the efficiency and specific impulse. As previously mentioned, one of the advantages of electric propulsion is that it can produce a higher exhaust velocity and a higher specific impulse compared to traditional chemical propulsion. The downside is that the mass of the propellant is very small, thus the acceleration is not as aggressive and the thrust is a fraction of the thrust seen in chemical engines. The different performance values for different types of electric propulsion methods can be seen in Table 2.1.

Table 2.1 - Performance values for different thrusters (legacy systems) [17]

Thruster	**Specific Impulse (s)**	**Thrust**	**Input Power (kW)**	**Efficient Range (%)**	**Propellant**
Cold Gas	25-75	0.1-100N	-	-	Various
Solid Chemical	250-304	10^7N	-	-	Various
Liquid Chemical (Monopropellant)	150-235	1-500N	-	-	$N_2H_4H_2O_2$
Liquid Chemical (Bipropellant)	247-467	10^7N	-	-	Various
Ressistorjet	100-300	0.5-6000mN	0.1-1	65-90	N_2H_4 Monopropellant
Arcjet	130-600	50-6800mN	0.9-2.2	25-45	N_2H_4 Monopropellant
Ion Thruster	2500-4000	0.1-750mN	0.4-5.4	40-83	Xe, Kr
Hall Thruster	1500-3000	0.1-2000mN	1.5-4.5	35-70	Xe, Kr
PPT	850-1200	0.05-10mN	< 0.2	3-30	Teflon
MPD	200-3200	0.1-2000mN	< 1	20-60	H_2, Li, Ar, Xe

The rocket equation is derived from the thrust, T, of the spacecraft being equal to the negative value of exhaust velocity of the propellant, v_e, multiplied by the change in mass as the spacecraft loses mass due to expending fuel, $\frac{dm_p}{dt}$.

$$T = -v_e \cdot \frac{dm_p}{dt} \tag{2.1}$$

The total mass of the spacecraft is the sum of the mass of the dry spacecraft (no fuel) and the mass of the propellant. The change in mass is only influenced by the mass of the propellant, because the mass of the spacecraft itself is not changing. Thus, the change in velocity, dv, is influenced by the negative exhaust velocity times the change in mass over the mass, $\frac{dM}{M}$.

$$dv = -v_e \cdot \frac{dM}{M} \tag{2.2}$$

The integral of Eq. (2.2) can be taken, with the left side being integrated from the starting velocity to the end velocity (also known as the delta-v), while the right side can be integrated from the mass of the spacecraft plus the mass of the fuel, $(m_s + m_p)$, to the mass of the spacecraft only, m_s.

$$\Delta v = v_e \cdot ln\left(\frac{m_s}{m_s + m_p}\right) \tag{2.3}$$

Eq. (2.3) can be rewritten to incorporate the specific impulse, I_{sp}, and the acceleration due to gravity, g, which is directly equal to the exhaust velocity. This is the final form for the rocket equation [17].

$$\Delta v = I_{sp} \cdot g \cdot ln\left(\frac{m_s}{m_s + m_p}\right) \tag{2.4}$$

2.2 Ion Thruster Force Transfer

In order to accelerate ionized propellant, electric fields and magnetic fields are used for electric propulsion. The method is different depending on the type of electric propulsion system. For ion thrusters, ions are electrostatically accelerated through oppositely charged grids: the screen grid and the acceleration grid. Using Poisson's Equation and integrating it from the screen grid to the acceleration grid, the equation below is obtained.

$$\sigma = \varepsilon_0 \cdot E_{screen} \tag{2.5}$$

σ is the density of the surface charge, ε_0 is the permittivity of free space, and E_{screen} is the electric field at the screen. This equation assumes that the screening grid is an ideal conductor. The screen grid and acceleration grid both experience a force per unit area due to the charge of the ions in the gap between the two grids. These two forces are equivalent to the σ multiplied by the average of the field. The force on the acceleration grid is the same as Eqs. (2.6) and (2.7), but is negative.

$$F' = \sigma \cdot \frac{(E+0)}{2} \tag{2.6}$$

F' is the force per unit area. This term can be calculated for both the screen grid and acceleration grid. Eq. (2.5) can be plugged into Eq. (2.6) to obtain the equation below.

$$F' = \varepsilon_0 \cdot E \cdot \frac{(E+0)}{2} = \frac{\varepsilon_0 \cdot E}{2} \tag{2.7}$$

The final thrust can be described as the sum of the two forces times the area, A [17].

$$T = \frac{\varepsilon_0}{2} \cdot \left(E_{screen}^2 - E_{acceleration}^2\right) \cdot A \tag{2.8}$$

2.3 Hall Thruster Force Transfer

Hall thrusters generate ions through the use of a discharge channel and accelerating the ions in a plasma electric field. The magnetic field raises the axial resistivity, which in turn increases the velocity of the ions. This is what causes the Hall current in the thruster. As the electrons are in the plasma, they experience a force known as the $\vec{E} \times \vec{B}$ force and have a velocity that can be expressed as follows.

$$\vec{v}_d = \frac{\vec{E} \times \vec{B}}{B^2} \tag{2.9}$$

v_d is the drift velocity of the electrons, B is the magnetic field. The force that is exerted on the ions comes from quasi-neutrality and the Hall current density.

$$\vec{F} = 2\pi \int\int \vec{J} \times \vec{B}\, r dr dz \tag{2.10}$$

F is the force on the ions, J is the current density. After integrating, a final value for the forces on the ions is obtained, and that is equal and opposite to the thrust [17].

$$\vec{T} = \vec{J} \times \vec{B} \tag{2.11}$$

2.4 Electromagnetic Thruster Force Transfer

When it comes to electromagnetic thrusters, they use a combination of pressure gradients and the Lorentz force to create thrust. By using the momentum equation paired with assuming the plasma consists of only single-charged ions and electrons, a momentum equation for a fluid for each component can be seen below.

$$mn\frac{d\vec{v}}{dt} = qn(\vec{E} + \vec{v}_i \times \vec{B}) - \nabla p - mn\nu(\vec{v} - \vec{v}_0) \tag{2.12}$$

m is the mass of the particle, n is the number density of the particles, $\frac{dv}{dt}$ is the acceleration, $v \times B$ is the magnetic force, ∇p is the pressure gradient. The $mn\nu(v - v_0)$ term represents the drag force, This is equal to the collision variable P_{ie} which is equal to $-P_{ei}$. To simplify Eq. (2.12) further, it can be assumed as 1D.

$$Mn_i \frac{d\vec{v}_i}{dt} = en_i(\vec{E} + \vec{v}_i \times \vec{B}) - \nabla p_i - \vec{P}_{ie} \tag{2.13}$$

M is the ion mass, e is the elementary charge. The final step to change the fluid equation of motion is to define the ion density as Mn, the pressure gradient as $\nabla p_i + \nabla p_e$, and J equal to $en(v_i - v_e)$.

$$\rho \frac{d\vec{v}_i}{dt} = \vec{J} \times \vec{B} - \nabla p_i \tag{2.14}$$

ρ is the ion density [17].

2.5 Thrust, Specific Impulse, and Efficiency

The main performance parameters of the thrusters are the thrust, specific impulse, and the efficiency of each respective engine. These parameters were already shown in Figure 2.1, but it is important to understand how to design each of these thrusters through the use of equations. For simplicity of the equations, Xenon will be used as the fuel source as a baseline.

When it comes to the total thrust with Xenon as the propellant, the equation can be written as follows.

$$T = 1.65 \cdot \gamma \cdot I_b \cdot \sqrt{V_b} \tag{2.15}$$

T is the thrust, γ is the correction factor, I_b is the beam current, V_b is the beam voltage. The correction factor can be calculated with Eq. (2.16).

$$\gamma = \alpha F_t \tag{2.16}$$

α is the thrust correction factor, F_t is the beam divergence. If a thruster has a uniform divergence once leaving the thruster, the beam divergence can be simplified.

$$F_t = cos\left(\theta_d\right) \tag{2.17}$$

θ in this equation is the beam's divergence angle divided by 2. The smaller the divergence angle, the more efficient the thruster is since more of the thrust is exiting parallel to the spacecraft's direction. In order to determine the thrust correction factor, Eq. (2.18) can be used.

$$\alpha = \frac{T_m}{T'} \tag{2.18}$$

T_m is the thrust from all species present summed up, T' is the thrust for single-charged fuel. To calculate the sum of all species, the equation below can be used.

$$T_m = I^+ \sqrt{\frac{2MV_b}{e}} \left(1 + \frac{1}{\sqrt{2}} \cdot \frac{I^{++}}{I^+}\right) \tag{2.19}$$

I^+ is the ion current for a single-charged ion, I^{++} is the ion current for a double-charged ion.

$$T' = \sqrt{\frac{2M}{e}} \cdot I_b \cdot \sqrt{V_b} \tag{2.20}$$

e is the elementary charge, which is the same as q. For Xenon as the propellant, which has $\sqrt{\frac{2M}{e}}$ being equal to 1.65E-3, the equation becomes

$$T' = 1.65E\text{-}3 \cdot I_b \cdot \sqrt{V_b} \tag{2.21}$$

The specific impulse equation when using Xenon can be seen below.

$$I_{sp} = 123.6 \cdot \gamma \cdot \eta_m \cdot \sqrt{V_b} \tag{2.22}$$

η_m is the mass utilization efficiency for the system. Eq. (2.22) comes from the fact that the specific impulse for constant flow rate of the propellant and a constant thrust net the equation below.

$$I_{sp} = \frac{T}{\dot{m}_p \cdot g} \tag{2.23}$$

g is the gravity effect, usually known as 9.81 m/s^2 on Earth. By plugging in Eq. (2.24) into Eq. (2.23), the equation for specific impulse for any thruster is given.

$$T = \dot{m}_p \cdot v_e \tag{2.24}$$

$$I_{sp} = \frac{v_e}{g} = \frac{v_i}{g} \cdot \frac{M}{\dot{m}_p} \tag{2.25}$$

v_i is the exit velocity for an ion propellant that is unidirectional. Pairing Eq. (2.25) with Eq. (2.26), the equation below is obtained.

$$T = \gamma \cdot \sqrt{\frac{2M}{e}} \cdot I_b \cdot \sqrt{V_b} \tag{2.26}$$

$\sqrt{\frac{2M}{e}}$ is equal to 1.65E-3 for Xenon, causing the equation to simplify to the final form of the equation, Eq. (2.27).

$$T = (1.65 \cdot 10^{-3}) \cdot \gamma \cdot I_b \cdot \sqrt{V_b} \tag{2.27}$$

When it comes to the total efficiency equation, three equations are needed.

$$\eta_m = \frac{I_b}{e} \cdot \frac{M}{\dot{m}_p} \tag{2.28}$$

$$\eta_e = \frac{P_b}{P_T} = \frac{I_b V_b}{I_b V_b + P_o} \tag{2.29}$$

$$\eta_T = \gamma^2 \cdot \eta_e \cdot \eta_m \tag{2.30}$$

Eq. (2.28) finds the efficiency of the thruster mass utilization, η_m, Eq. (2.29) is the electrical efficiency, η_e, and Eq. 2.30 is for the total efficiency, η_T. M is the ion mass, $\dot{m}_p$ is the flow rate of the propellant, P_T is the total input power, P_b is the beam power, and P_o is the remaining input power necessary to power the thruster. The flow rate of the propellant can be found in Eq. (2.31), which can be plugged back into Eq. (2.28).

$$\dot{m}_p = Q \cdot M \tag{2.31}$$

Q is the flow rate of the particles of the propellant. The discharge loss, η_d, can also be calculated.

$$\eta_d = \frac{P_d}{I_b} \tag{2.32}$$

P_d is the power needed to create the ions, while I_b is the beam current. Since Eq. (2.29) calculates a power loss, this needs to be as low as possible. One of the last things needed to know is the dissipation factor for the power, $P_{dissipated}$. This is the power that does not get used by the system.

$$P_{dissipated} = P_{in} \cdot \left(1 - \eta_e\right) \tag{2.33}$$

P_{in} is the power needed for all the components of the thruster. There are instances where the efficiency of the electrical system of the thruster is not known accurately enough, where other methods are needed to calculate the dissipated power [17].

Chapter 3: Hall Thruster Engine Design

The main component of the project is the engine. After doing the comparative study, the electric propulsion that this report will go over and design is a Hall-effect thruster. The literature review discusses the advantages of each type of thruster and how they work, while the comparative study briefly described the mathematics for each of the thrusters. This chapter will go over the design and analysis of the Hall-effect thruster, starting with the components.

3.1 Key Components and Material Selection

One of the main components that will be discussed are the components for creating the magnetic field inside of the acceleration chamber. The magnetic field location and strength are critical in the design. The vector of the magnetic field needs to act radially, from the center magnet to the outer magnets, and vice versa. The orientation of the system will be to have 4 coils at each corner of the base plate, and a fifth inverted coil at the center. The purpose of the inverted coil is to make the path of the magnetic field into a loop rather than a figure eight. as seen in Figure 3.1. The configuration can be seen in Figure 3.1. The material chosen is a permanent magnet to reduce power requirements, as well as to add simplicity. The material chosen will be Neodymium Iron Boron, specifically NdFe30 due to it being available in the Analysis tool that was used, along with it being a common magnet.

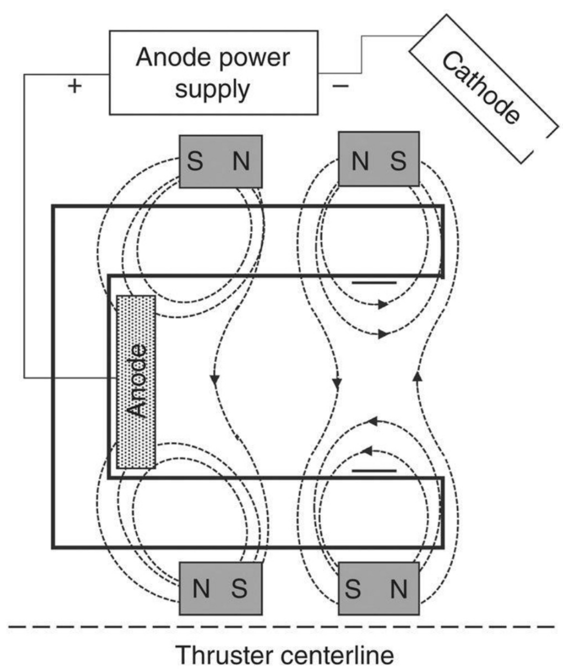

Figure 3.1 - Two-peak magnetic field Hall thruster schematic [17]

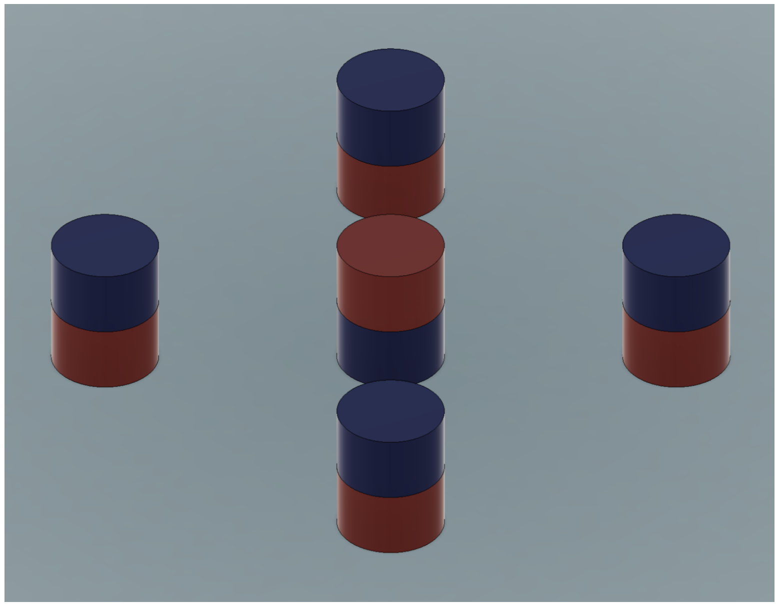

Figure 3.2 - Magnetic field placement in configuration

The second main design parameter of hall thrusters is the electric field that is created from the anode and the cathode. Starting with the anode, which can be seen in Figure 3.2, this component is typically placed at the back of the acceleration chamber, while also being positively charged. The anode has holes that allow for the neutral Xenon atoms to enter the acceleration chamber before they are ionized. The anode is usually created out of a corrosive resistance material, such as tungsten.

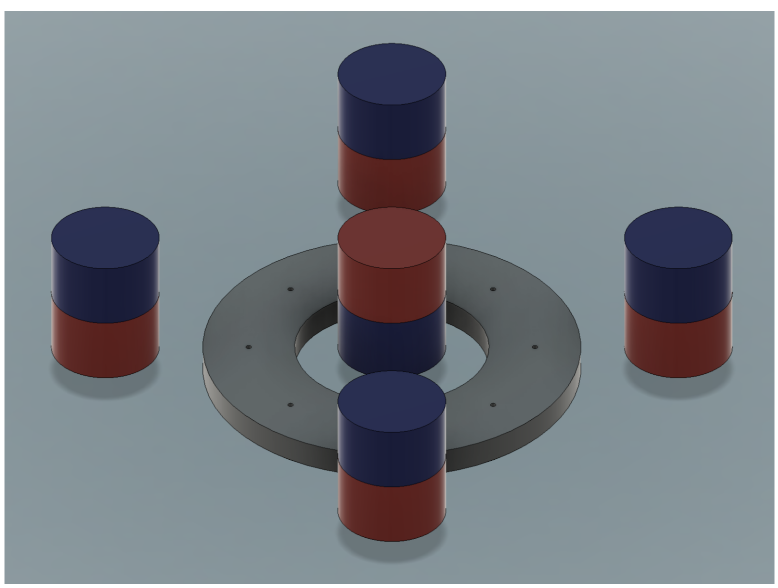

Figure 3.3 - Anode placement in configuration

The cathode is the second component that is needed in order to complete the circuit for the electric field, and is placed at the exit of the acceleration chamber, as seen in Figure 3.3. The cathode is the negatively charged component, while also being the initial source for the electrons that will be used to ionize the propellant; that being the Xenon gas. The other purpose of the cathode is to neutralize the ions. As mentioned before, this prevents the ions from being pulled back towards the spacecraft that would cancel out the thrust being produced. For the cathode, the material chosen will also be tungsten as it has a large electron emissivity, while also being able to handle high temperatures. Having a high electron emissivity means that electrons can easily escape from a material when that material is heated up. This makes it a suitable candidate for the specific task. The cathode design will change later in the final design. It will be placed above the center magnet to create a uniform electric field, and to minimize the size of the overall Hall thruster.

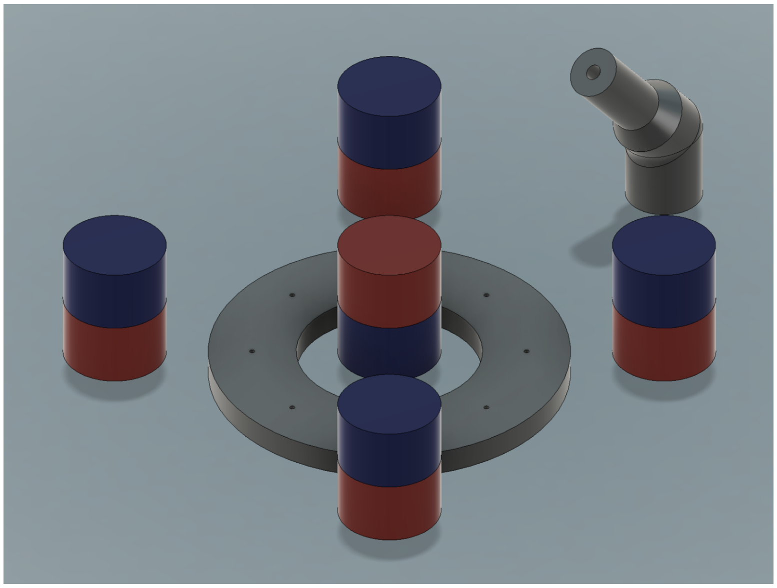

Figure 3.4 - Cathode placement in configuration

The next component, which has already been mentioned, is the acceleration chamber. This component is the location where the Xenon atoms will get released into, come into contact with the electrons that are trapped inside of the magnetic field, and accelerate out of the back, hence the name of the component. This component is usually annular in shape and positioned in a way that the magnetic field is created inside of this chamber. As for the material of the acceleration chamber, boron nitride will be chosen for its thermal and erosion resistance, as well as its low electrical conductivity which will help with it not interfering with the electric field being created by the anode and cathode. Boron nitride is also non-magnetic, so it will not interfere with the magnetic field that passes through it. Finally, this material will also not react with the ionized Xenon atoms due to it being chemically compatible with the environment. The acceleration chamber's design and placement can be seen below, in Figure 3.4.

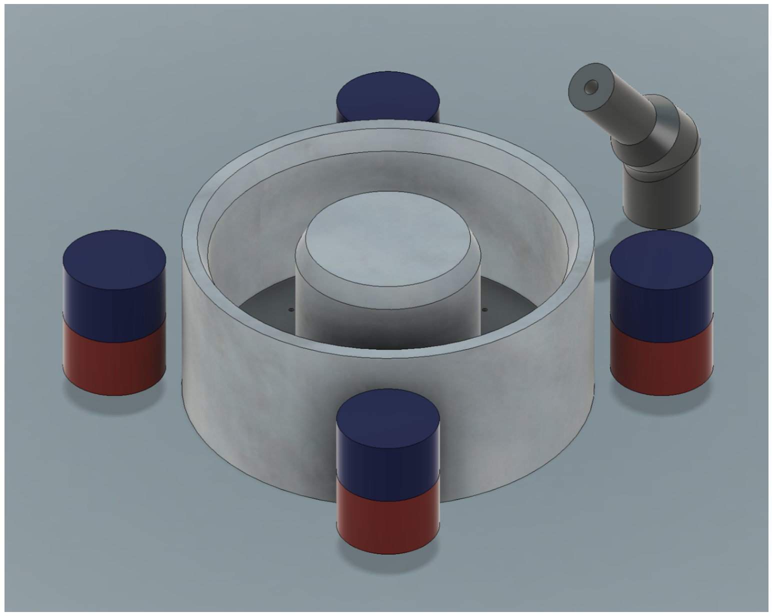

Figure 3.5 - Acceleration chamber placement in configuration

The final components for the Hall thruster are the base plate and exit plate that hold everything together and act as the structural component. An important aspect of these plates is that they are non-magnetic, and less conductive than the anode/cathode. A good structural material that would not affect the fields a concerning amount is stainless steel that is not magnetic. While stainless steel is conductive, it is not a strong conducive metal like the tungsten cathode and anode, meaning it would be a good choice for the application. The full configuration of the Hall-effect thruster can be seen in Figure 3.5.

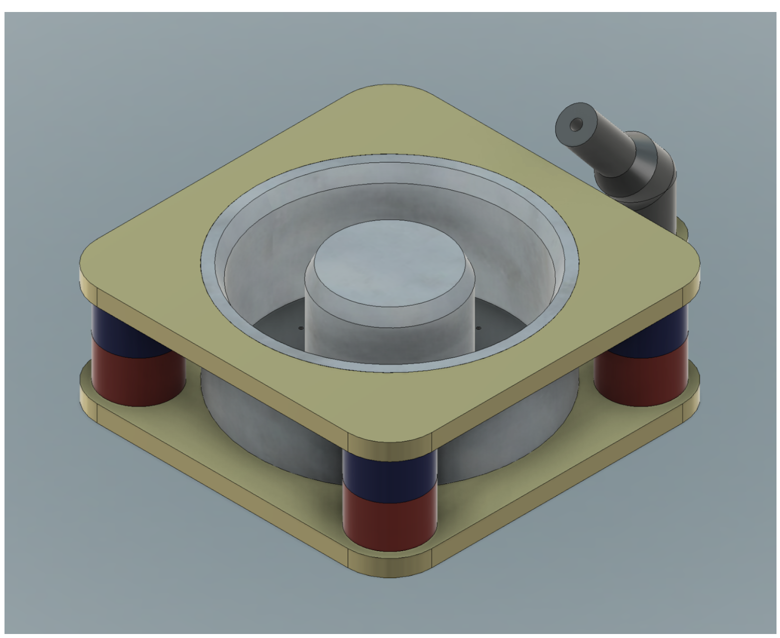

Figure 3.6 - Base plate and exit plate placement in configuration

Chapter 4: Ansys Maxwell Validation

The software that will be used for the design and optimization of the Hall thruster will be Ansys Maxwell 3D. The benefit of this software is that it can show the magnitude and vector direction of the magnetic and electric fields, which allows for the engineer to adjust size, placement, material, and so on, to create the fields that are necessary for the design parameters. Before taking the results of the software for what they are, the results for a simple model will be compared to hand calculations in order to validate the results.

4.1 Magnetic Field Validation

4.1.1 Magnetic Field Ansys Maxwell Results

For the validation of this software, a two permanent magnet system will be analyzed. Both of the magnets will be cylindrical, with one of the magnets being placed at the origin, and the other being placed at a value in the X-direction. The centerlines of each magnet will run through the Z direction, with one magnet having its North pole going positively in the Z-direction and the other having its direction facing the negative Z-direction. This configuration is similar to Figure 3.1, as well as the configuration of the Hall thruster being designed for this report for the center magnet and one of the corner magnets. The final configuration for the validation can be seen in Figure 4.1.

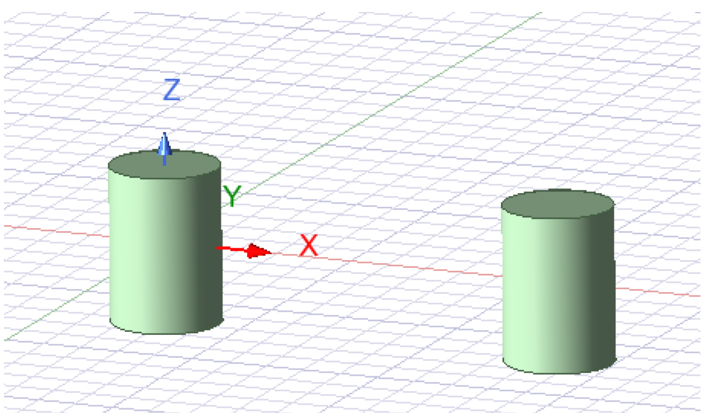

Figure 4.1 - Simple permanent magnet configuration for validation

The magnets are cylindrical to mimic Hall thrusters that are widely used. In order to have an optimal magnetic field, the magnetic field should have a value between 10 and 30 mTesla. This amount is what is needed in order to trap electrons in the magnetic field while also not being too strong to limit electron mobility. Having high electron mobility allows for a better ionization process. Keeping this range in mind, the permanent magnets were placed 53 mm away from each other, which is what was the case for the initial design that was seen in Chapter 3. The magnets have a radius of 7 mm and a height of 20 mm. The last piece that was needed was the material, which has already been chosen to be NdFe30. For choosing the analysis setup in Ansys Maxwell, the allowed percent error selected was a 0.5 percent to ensure accurate measurements while also not hitting the max limit of the mesh size that is allowed for the educational version of Ansys Maxwell. Finally, the results were analyzed along the XZ plane to visualize the magnetic field and ensure that the magnitude and direction of the magnetic field was what was expected. Taking a look at Figure 4.2, the magnetic field lines are traveling from the North poles of the permanent magnets to the South poles, which is what is expected.

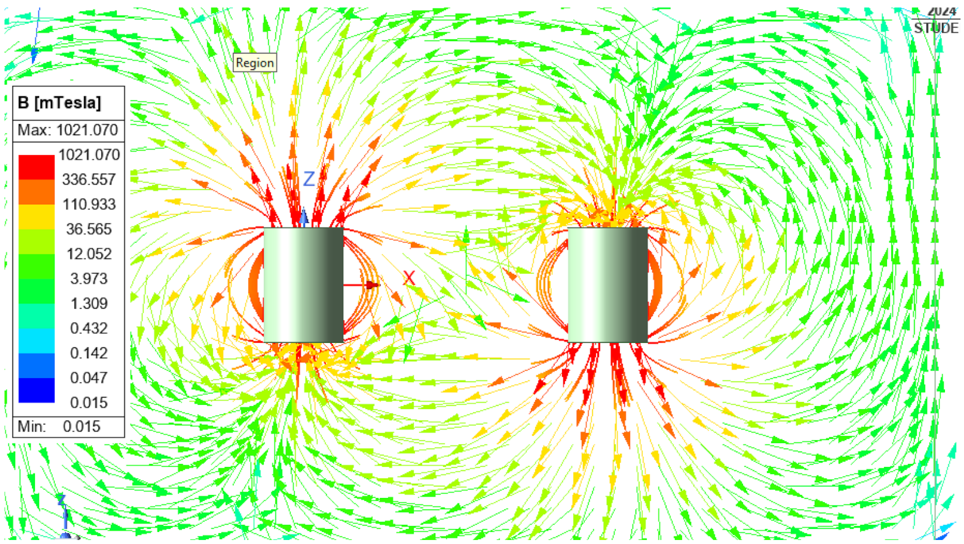

Figure 4.2 - Magnetic field vector direction for Ansys validation

The specific location of the magnetic field that will be validated will be at the coordinates (-53/2, 0, 10). The purpose for this is that it is directly in the middle of both magnets, but also, at z = 10 mm is one of the main locations where the electrons will be trapped because of the height of the magnets. At this location, it can be seen in Figure 4.2 that the magnetic field vector travels in the positive X-direction. Now that the magnetic field direction is known, the magnitude needs to be known. Taking a look at Figure 4.3, the magnitude of the entire domain can be seen. While this figure does show the magnitude, the scale makes it difficult to determine the exact magnitude at z = 10 mm. Adjusting the scale, Figure 4.4 is obtained. This figure shows the magnitude of the magnetic field between the two magnets at z = 10 mm to be between 20 and 23 mTeslas, which is in the range for the magnetic field that is needed.

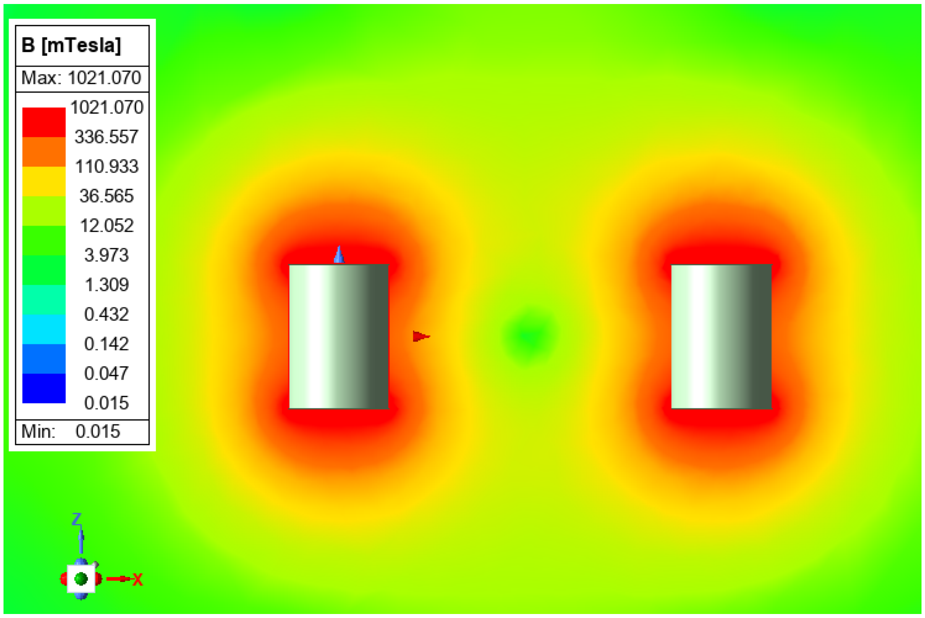

Figure 4.3 - Magnetic field magnitude for Ansys validation

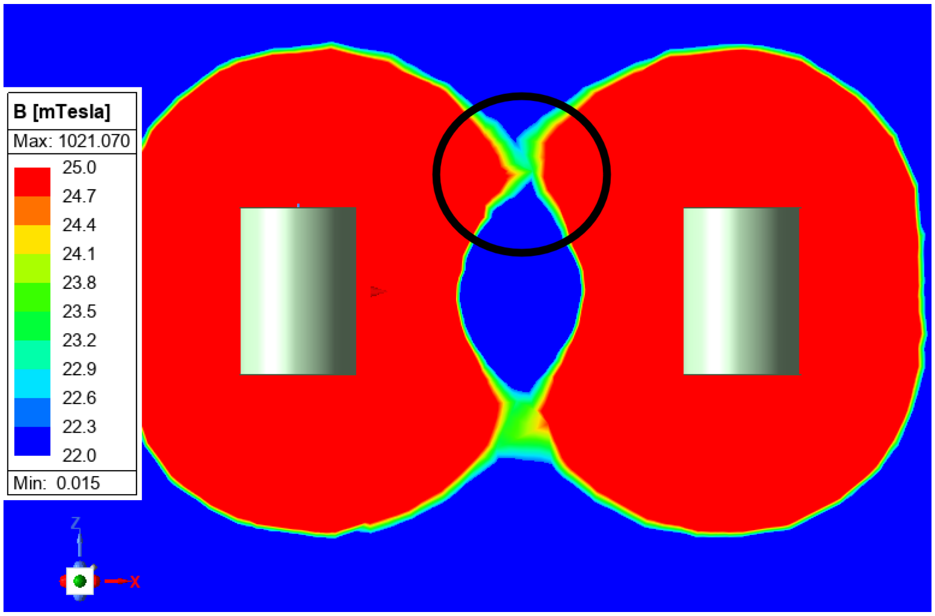

Figure 4.4 - Scaled magnetic field magnitude for Ansys validation

4.1.2 - Magnetic Field Hand Calculation Results

In order to validate the results that Ansys Maxwell determined, hand calculations can be used. The method that will be used is the dipole approximation. The dipole approximation assumes a magnetization that is uniform through the volume of the magnet. This in turn allows for the simplification of using the dipole moment, m.

$$m = M \cdot V \tag{4.1}$$

The dipole moment is described by the magnetization, M, times the magnet's volume, V. M is found through Eq. (4.2). For NdFe30, B_r is equal to 1.1 Tesla. μ_0 is the permeability of free space, which is equal to $4\pi \times 10^{-7}$ T·m/A.

$$M = \frac{B_r}{\mu_0} = 875,352 \text{ A/m} \tag{4.2}$$

For the volume of the magnet, this depends on the shape of the magnet. In the case of this report, the volume needs to be calculated for a cylinder. The dipole moment can be found with the calculated magnetization and the volume, as seen in Eq. (4.3).

$$\vec{m} = (0,\ 0,875352) \cdot \pi(0.007)^2 \cdot (0.02) = (0,\ 0,\ 2.695) \text{ Am}^2 \tag{4.3}$$

The point that the magnetic field needs to be evaluated will be referred to as the evaluation point. The position vector of the evaluation point then gets subtracted from the position vector of each magnet separately to find the relative position vector, $\vec{r}$. This vector is then inserted into the dipole field equation, Eq. (4.3).

$$\vec{B} = \frac{\mu_0}{4\pi} \cdot \left[\frac{3(\vec{m}\cdot\vec{r})\vec{r} - \vec{m}}{|\vec{r}|^3} \right] \tag{4.4}$$

The dipole field equation needs to be calculated for both magnets, and then summed together to find the net magnetic field at that point. Doing all of this gets a magnitude value of 23.506 mT, with a vector component of (23.506, 0, 0).

4.2 Electric Field Validation

4.2.1 Electric Field Ansys Maxwell Results

To validate the Maxwell model for predicting the electric field, the configuration was redesigned to place the cathode at the back of the Hall thruster, with the centerline of the cathode running through the centerline of the thruster. This also creates a uniform electric field, instead of the electric field being stronger on one side of the thruster, and weaker on the opposite side. For the Maxwell model, an annular ring is modeled for the anode, with an outer radius of 35 mm and an inner radius of 18 mm, with a thickness of 5 mm. It is also placed on the XY plane, with the center of the ring at 0.0 mm in the Z-direction. For the cathode, it had a radius of 12 mm and a thickness of 5 mm and was placed in the Z-direction at 30 mm. This causes the center-to-center distances of the cathode and anode to be 30 mm apart. The final geometry configuration can be seen in Figure 4.5.

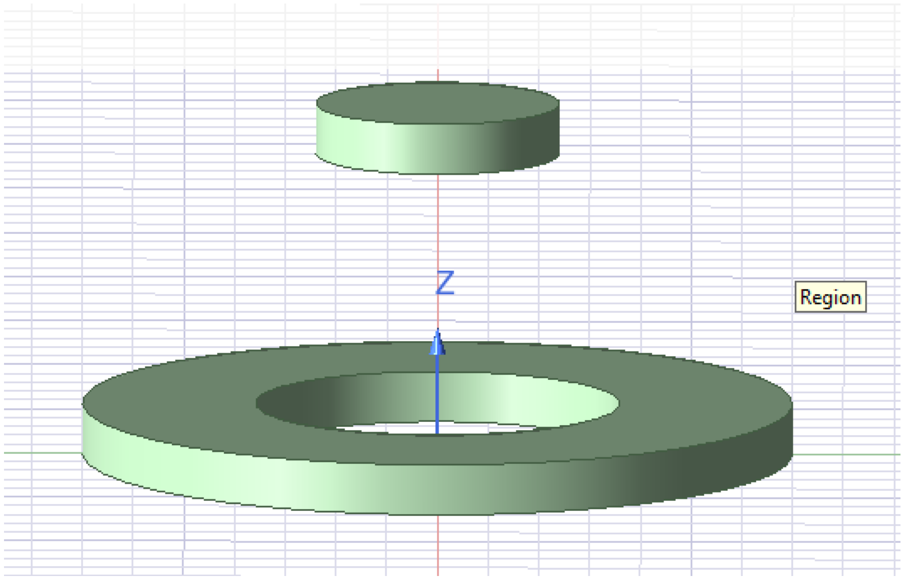

Figure 4.5 - Simple anode and cathode configuration for validation

The anode and cathode were given the material properties of tungsten, as that was originally chosen in Chapter 3 for being a good conductor of electricity and high electron emissivity. The next part for setting up the model was to add an excitation of 500 V to the anode and an excitation of 0.0 V to the cathode, and the model was chosen to reach a percent error that is less than 0.5 percent. In order to view the electric field's orientation, the YZ-plane was chosen for the viewing plane. The first field overlay can be seen in Figure 4.6. This field overlay consists of the vectors for the electric field. Comparing this to the fully configured Hall thruster, it can be seen that the location of the acceleration chamber will have an electric field that is parallel to the centerline of the Hall thruster, which is what is wanted.

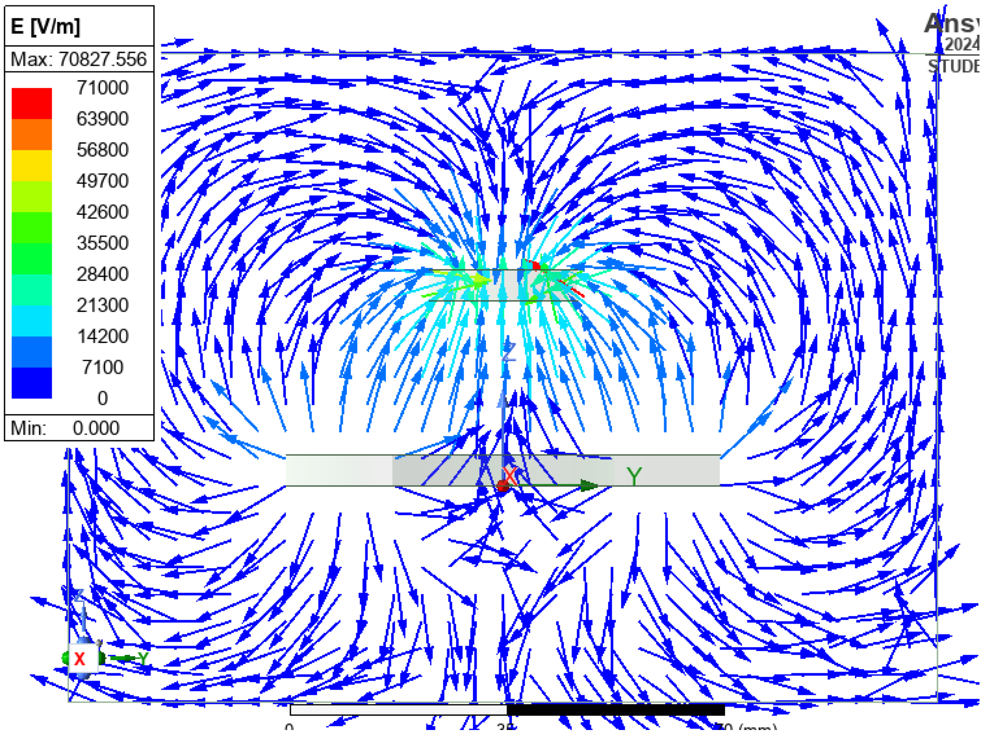

Figure 4.6 - Vector field vector direction for Ansys validation

The magnitude of the electric field ranges from 0.0 V/m along the outer sections of the domain, and scales up to 71,000 V/m at the edges of the cathode. Around the location of the channel for the acceleration chamber, the electric field ranges from 2,500 V/m to 25,000 V/m.

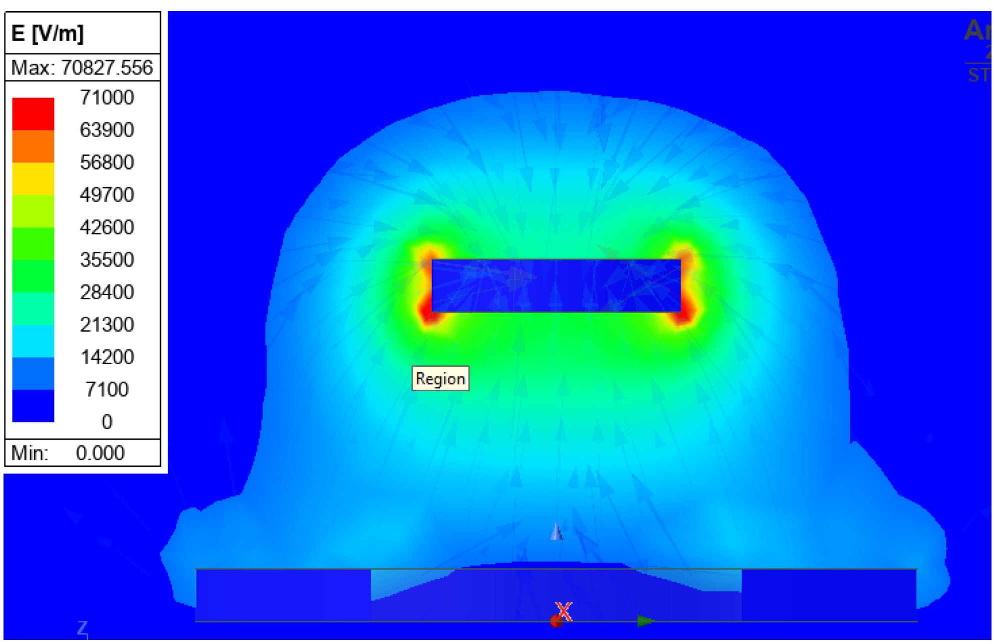

Figure 4.7 - Electric field magnitude for Ansys validation

The strength of the location for the electric field will be simplified by focusing on the mid-point between the anode and cathode. The reason for this is because the method for computing the electric field with hand calculations properly is difficult, and assumptions to make it more simple to compute involves assuming a uniform electric field. The uniform electric field that is found through the hand calculations is equivalent to the center of the electric field found in a proper electric field simulator, such as Ansys Maxwell. For Ansys Maxwell, if the scale of the color map is altered, the gradient can be made to focus on the center of the electric field, as seen in Figure 4.8, the exact center between the anode and cathode yields a value around 16,500 V/m.

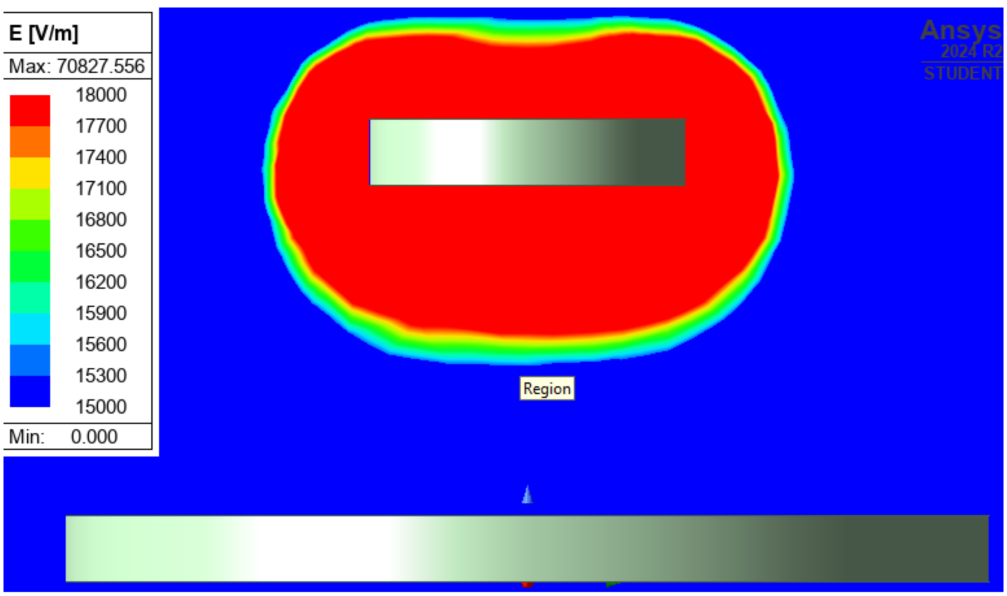

Figure 4.8 - Scaled electric field magnitude for Ansys validation

4.2.2 Electric Field Hand Calculation Results

To calculate the value for the electric field, only two inputs are needed. The first input is the voltage that is being supplied to the anode. The second input is the distance between the anode and cathode. This gives a value for the electric field, E, assuming a uniform electric field between the components.

$$E = \frac{V}{D} \tag{4.5}$$

By using the same input values as for Ansys Maxwell, V is equal to 500 V, and D is equal to 0.03 m, the value for the electric field would be 16,667 V/m.

4.3 Comparison Cases For the Fields

4.3.1 Results and Analysis

In order to validate the software, and ensure that the test cases previously chosen were not a coincidence, three test cases will be compared. For the first test case, the values previously chosen in section 4.1 and 4.2 will be compared more accurately. The other two cases will have randomly selected inputs

Table 4.1 - Test Case 1 Inputs

Magnetic Field Parameters	Values
Radius	0.007 m
Height	0.02 m
Distance	0.053
B_r	1.1 T
Evaluation point, r	[0.0265, 0, 0.01] m
Electric Field Parameters	Values
Voltage	500 V
Distance	0.03 m

Table 4.2 - Test Case 1 Comparison

Magnetic Field Ansys Maxwell	21.5 mT
Magnetic Field Hand Calculation	23.5 mT
Magnetic Field Percent Error	14.63%
Electric Field Ansys Maxwell	16700 V/m
Electric Field Hand Calculation	16667 V/m
Electric Field Percent Error	0.20%

Table 4.3 - Test Case 2 Inputs

Magnetic Field Parameters	Values
Radius	0.05 m
Height	0.05 m
Distance	0.3
B_r	1.1 T
Evaluation point, r	[0.15, 0, 0.025] m
Electric Field Parameters	**Values**
Voltage	300 V
Distance	0.1 m

Table 4.4 - Test Case 2 Comparison

Magnetic Field Ansys Maxwell	12.5 mT
Magnetic Field Hand Calculation	9.51 mT
Magnetic Field Percent Error	23.92%
Electric Field Ansys Maxwell	3000 V/m
Electric Field Hand Calculation	2800 V/m
Electric Field Percent Error	6.67%

Table 4.5 - Test Case 3 Inputs

Magnetic Field Parameters	Values
Radius	0.02 m
Height	0.03 m
Distance	0.1
B_r	1.1 T
Evaluation point, r	[0.05, 0, 0.015] m
Electric Field Parameters	Values
Voltage	10000 V
Distance	0.5 m

Table 4.6 - Test Case 3 Comparison

Magnetic Field Ansys Maxwell	40 mT
Magnetic Field Hand Calculation	38.31 mT
Magnetic Field Percent Error	4.225%
Electric Field Ansys Maxwell	15200 V/m
Electric Field Hand Calculation	20000 V/m
Electric Field Percent Error	31.58%

4.3.2 Discussion

When comparing the results between the different test cases, multiple conclusions can be made. When it comes to the magnetic field, it is noticeable that the dipole approximation is less accurate at larger distances between the two magnets, paired with a smaller radius for the magnets size. For the first test case, the distance between the two magnets from the surfaces is 39 mm, which yields a magnetic field percent error of 9.30%. For test case 2, the distance from surface to surface is 200 mm, with a percent error of 23.92%. For the third case, 60 mm is the distance between the surfaces of the two magnets, with a percent error of 4.225%. While the distance is playing a part, it is not the main factor or else test case 1 would have the smallest percent error. While the dipole approximation is a good approximation for specific configurations, it still has its limitations.

Comparing the electric fields together, where the only factors are the voltage and distance between the anode and cathode, there are configurations that make the results very volatile in terms of the percent error. For test case 1, with a voltage of 500 volts and a distance of 30 mm, the percent error becomes 0.20%. The next case slightly decreased the voltage to 300 volts and significantly increased the distance to 100 mm. This caused the percent error to increase to 6.67%. For the final case, both the voltage and distance were largely increased to 10,000 volts and 500 mm, respectively, yielding a percent error of 31.58%. This reveals that increasing the distance between the anode and cathode also increases the percent error between the hand calculation model and Ansys Maxwell. While this could be problematic, test case 1 is close to the final configuration for the electric field, which is the test case where both calculations agree favorably.

Chapter 5: Final Hall Thruster Design and Configuration

Before finalizing a design for the Hall-effect thruster, a parameter needs to be set to ensure the proper Electric Field and Magnetic Field relationship. That relationship refers to Eq. (2.9), which is the velocity drift equation. This velocity needs to be within the range of 10,000 m/s and 50,000 m/s [17]. It is in this range that the electrons will move at a high enough speed to ionize the Xenon gas, without having too much energy loss. After many iterations, the best value for the electric field is between 300 V/m to 2100 V/m. In order to reach these values, the configuration in Figure 5.1 was the solution. The configuration consists of an anode ring with an outer diameter of 40 mm, an inner diameter of 20 mm, and a thickness of 3 mm, paired with a cathode placed 50 mm above with a radius of 10 mm and a thickness of 3 mm. The anode was given a voltage of 100 volts.

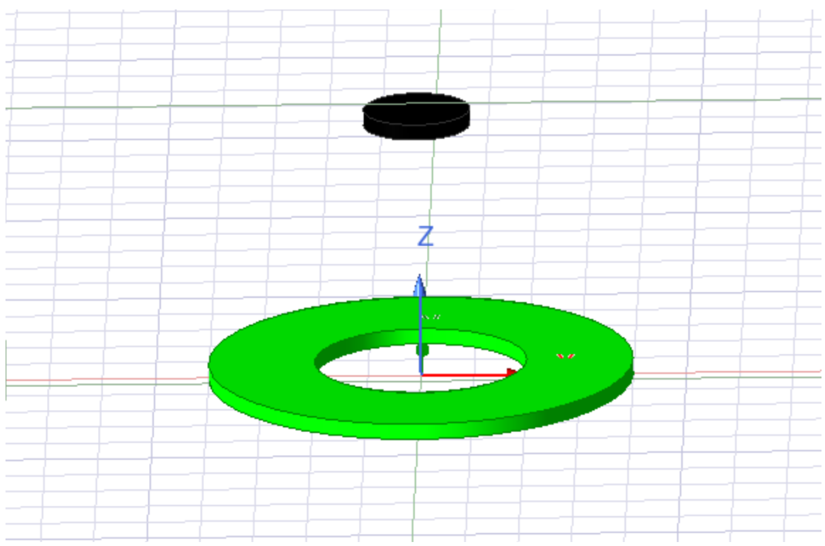

Figure 5.1 - Configuration of the electric field components

When analyzing the results of the electric field in Figure 5.2, it can be seen that the range in the acceleration chamber will be around the range needed; however, one thing to point out is that the direction of the electric field vectors nearly point in the x-direction exclusively near the top of the channel. This can greatly affect the velocity drift and cause the velocity vector to go in the wrong direction. To fix this, when designing the magnetic field, it is important to have the magnetic field lower in the acceleration chamber, where the electric field magnitude and vectors are more appropriate for the design. The best location for this is around 40 mm above the anode.

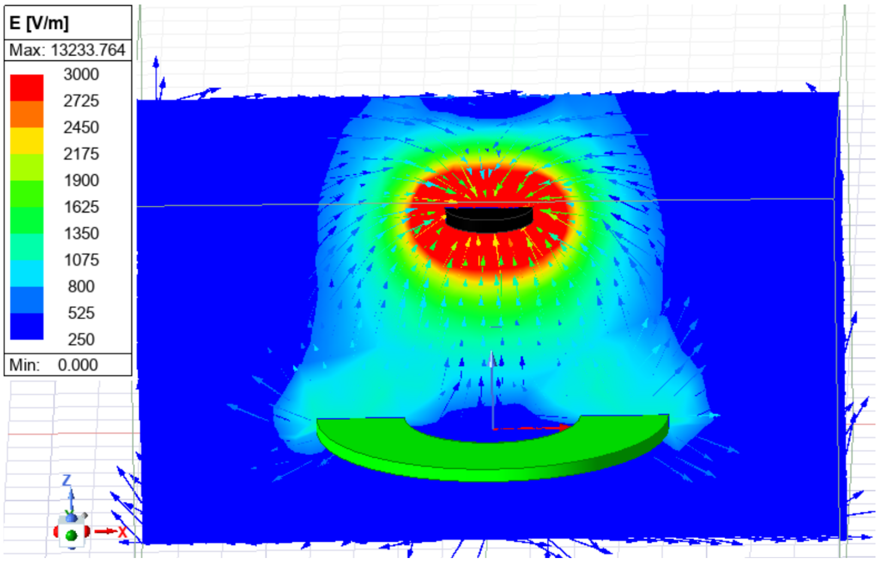

Figure 5.2 - Magnitude and direction of the electric field

Now that the location for the magnetic field is known, along with the range of 10-30 mTeslas, the magnetic field can be finalized. There is a centralized magnet with the north pole facing up, and 4 additional magnets placed along the perimeter with the north pole facing down, creating an X shape. Each magnet has a radius of 6 mm and a height of 40 mm. The perimeter magnets are placed symmetrically 40 mm in the X-direction and 40 mm in the Y-direction. This results in each of these magnets being $40\sqrt{2}$ mm from the center magnet, or approximately 56.6 mm. The electric field components are placed to give a sense of scale for the whole configuration, which can be seen in Figure 5.3. The material of the magnets are set to be NdFe30.

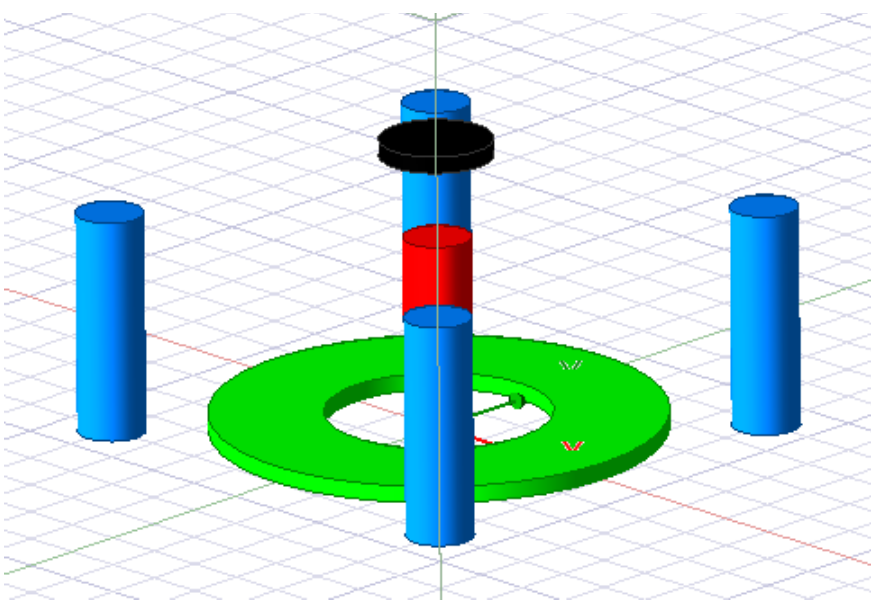

Figure 5.3 - Configuration of the magnetic field components

Taking a look at the results for the magnetic field in Figure 5.4, the magnetic field that will be inside of the acceleration chamber is around 20 mTeslas, which is between the necessary 10-30 mTeslas, as previously stated. The inside of the acceleration chamber will run along the outside of the anode, which means that there will be part of the magnetic field that will be more than 30 mTeslas inside of the acceleration chamber. While this is the case, this should not cause too much of an issue as the majority of the magnetic field is below that. As the electrons are running along the magnetic field path, they have a high probability of coming into contact with the Xenon atoms under the 30 mTeslas sections. Decreasing the magnetic field would cause the field to be too weak to trap the electrons.

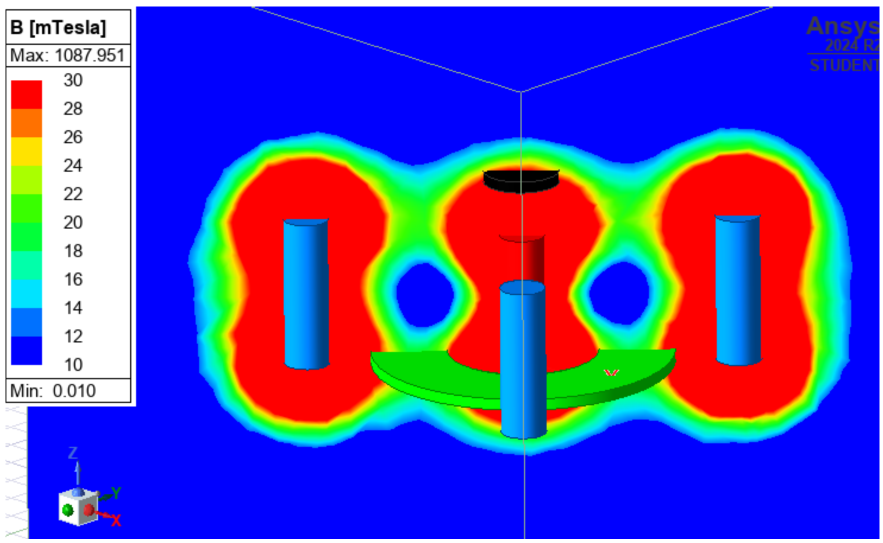

Figure 5.4 - Magnitude and direction of the magnetic field

Now that the electric and magnetic fields are properly optimized for the Hall-effect thruster, more parameters of the thruster can be analyzed with the equations from Section 2.5. To determine the thrust of the thruster, a few constants have to be set, along with assumptions that have to be made. These variables can be found in Table 5.1 and Table 5.2.

Table 5.1 - List of constants for calculations

Variable	Value
q	1.602E-19 C
M_{Xenon}	2.18E-25 kg
g	9.81 m/s^2

Table 5.2 - List of assumptions for calculations

Variable	Value
V_b	$0.95 \cdot V_d = 0.95 \cdot 100\,V = 95\,V$
I_b	1.1 A
I^+	$0.96 \cdot I_b = 0.96 \cdot 1.1 = 1.056\,A$
I^{++}	$I_b - I^+ = 1.1 - 1.056 = 0.044\,A$
θ	10°
P_0	$12\,V \cdot 1.1\,A = 13.2\,W$
Q	8E18 Particles/sec

After inserting all of these variables into the equations from Section 2.5, the thrust, specific impulse, efficiencies, and power dissipated can be calculated. The results are found in Table 5.3.

Table 5.3 - Calculated parameters for the Hall-effect thruster

Variable	Value
Thrust	16.34 mN
Specific Impulse	955.02 sec
Mass Utilization Efficiency	0.9236
Electrical Efficiency	0.8583
Total Efficiency	0.8878
Power Dissipated	13.82 W

After calculating all of the parameters for the Hall thruster, the final iteration can be made, which can be seen below, in Figure 5.5 and Figure 5.6. Two section analysis views can be seen in Figure 5.7 and Figure 5.8. The first section analysis view is along the XZ-plane to show the internal path for the cathode's power source. The second section analysis view is along a plane that is rotated 45 degrees from the original XZ-plane to show how the magnets will be held in place.

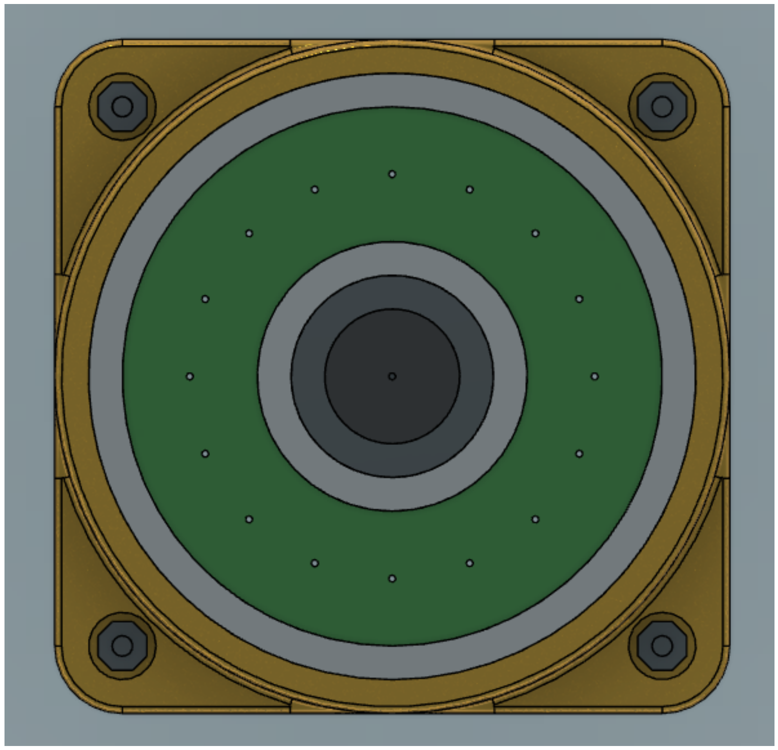
Figure 5.5 - Top-down view of Hall-effect thruster

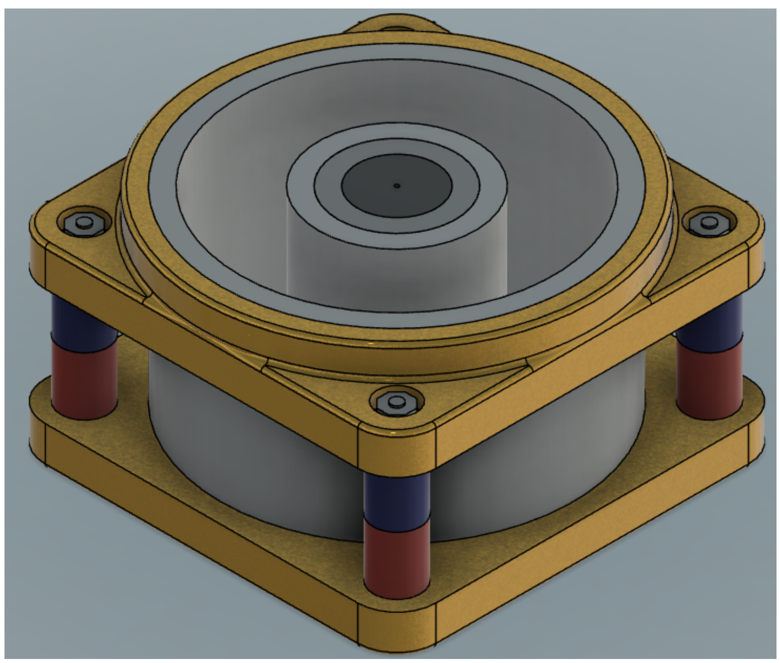
Figure 5.6 - Isometric view of Hall-effect thruster

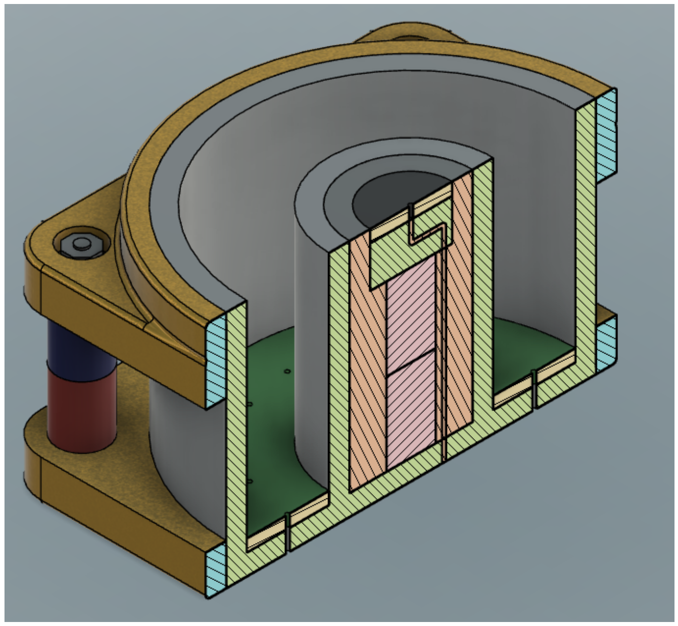

Figure 5.7 - Hall-effect thruster section analysis view (XZ-plane)

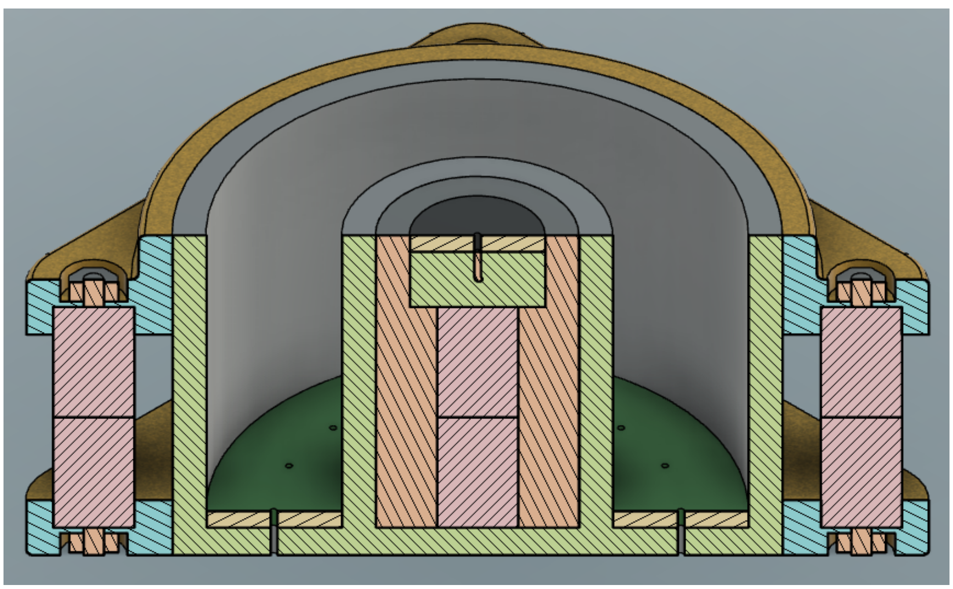

Figure 5.8 - Hall-effect thruster section analysis view (45 degrees from XZ-plane)

Chapter 6: Thrust Vectoring CAD Model

The second main component of the module is the thrust vectoring hardware. This hardware will allow for the necessary pivoting of the thruster.

6.1 First Drafted Model

The initial model, seen in Figures 6.1-6.4, was created in order to understand the components necessary for the model to behave as one would expect. The gold component is a hollow cylinder, meant to act as the outer dimensions of the engine. There are two gray components, with one being the holder of the engine, and the other component being the “ring” that contains all of the pivot points for the thrust vectoring. There are also two blue and white components, which are the servos that allow for controlling the motion of the thrust vectoring. The black components are the connecting rods, connecting the servos to the components that pivot for each of the two directions. Finally, the green component is just to act as an anchor point for the mechanism, and can be treated as the main body of the satellite.

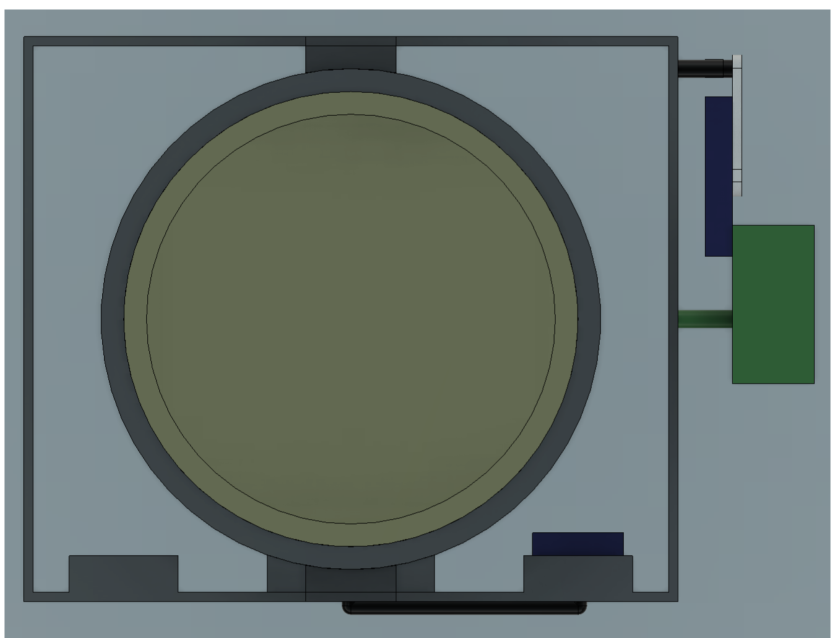

Figure 6.1 - Top down view of the thrust vectoring mechanism (Version 1)

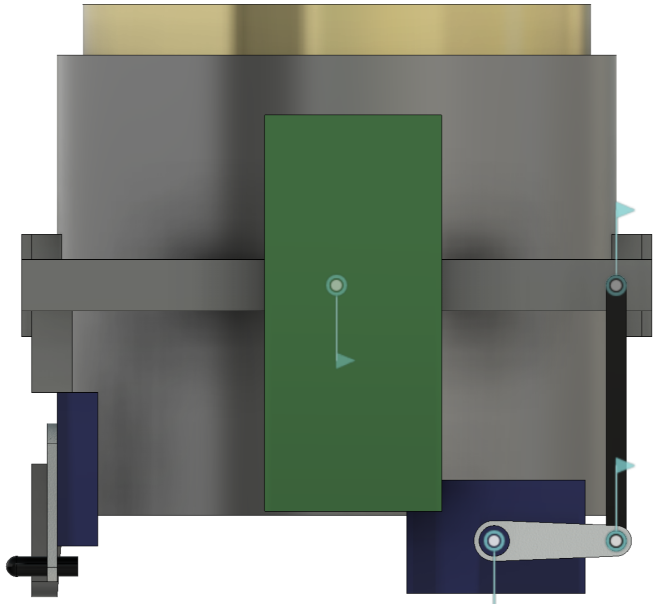

Figure 6.2 - Side view of the thrust vectoring mechanism (Version 1)

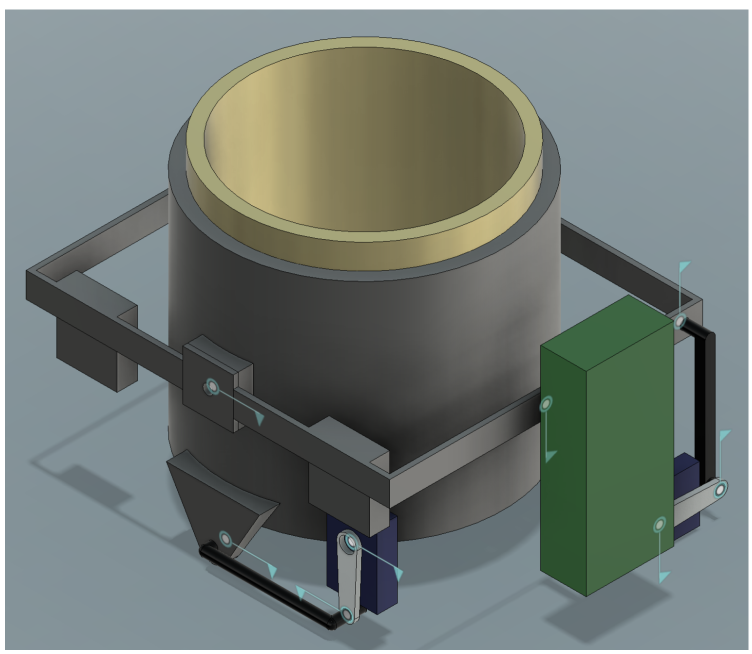

Figure 6.3 - Isometric view of the thrust vectoring mechanism (Version 1)

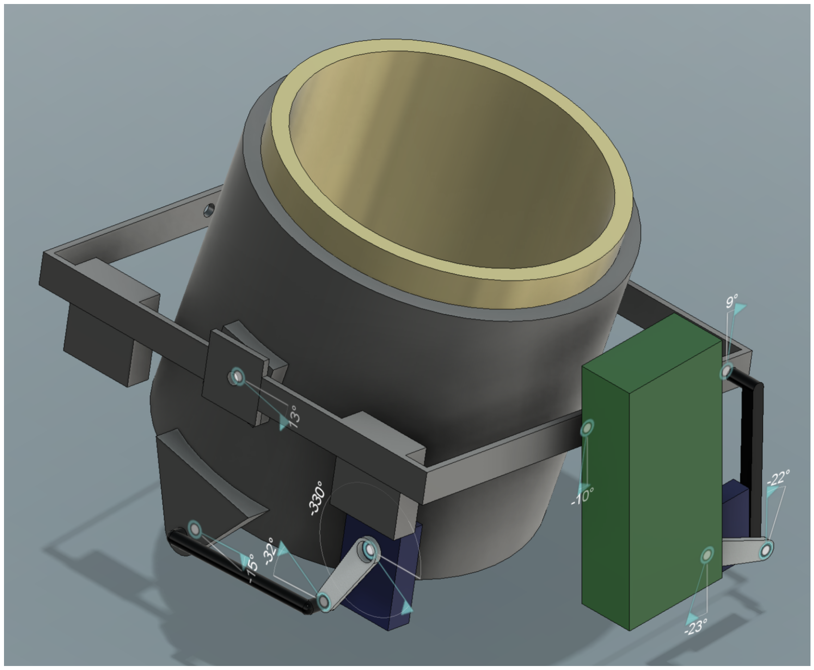

Figure 6.4 - Depiction of how servos affect motion (Version 1)

6.2 Second Drafted Model

The second model took the understanding of how the first model worked, mechanically, and focused on decreasing the footprint of the model itself. The second model can be seen in Figures 6.5-6.8. The main difference between the two models can be seen with the ring that surrounds the engine cover. While the ring itself is much closer to the engine cover, it still allows for the engine to pivot 10 degrees in all directions without touching any of the other components.

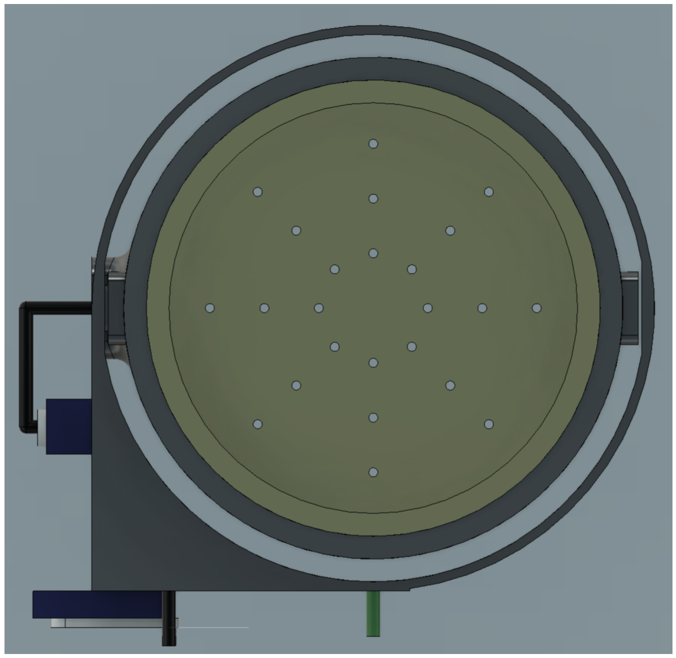

Figure 6.5 - Top down view of the thrust vectoring mechanism (Version 2)

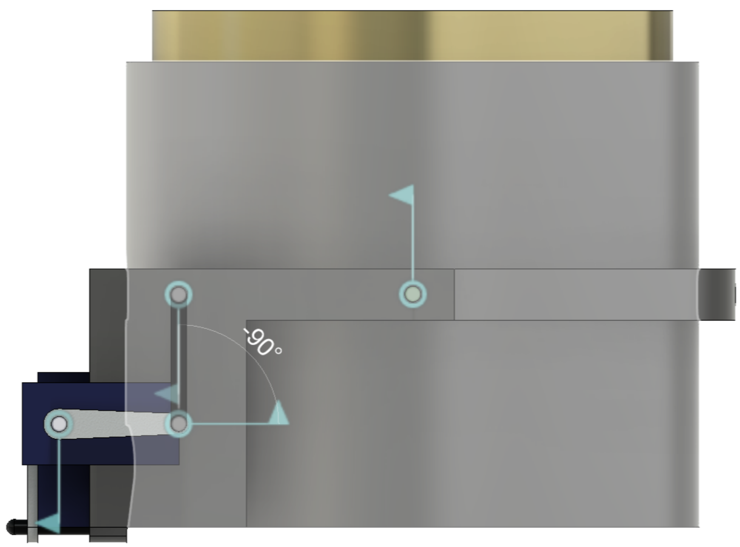

Figure 6.6 - Side view of the thrust vectoring mechanism (Version 2)

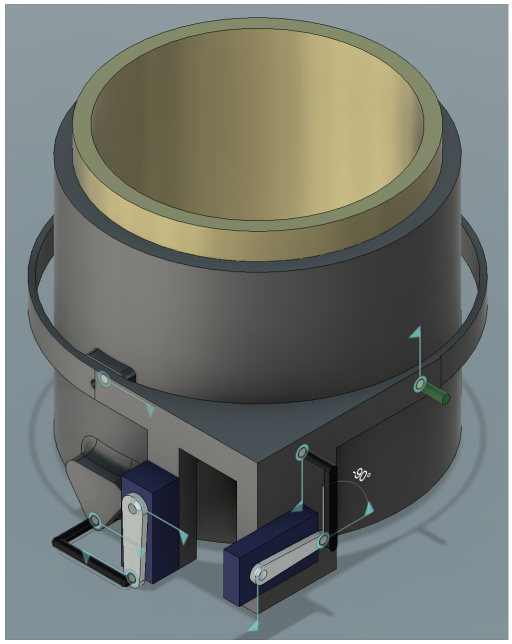

Figure 6.7 - Isometric view of the thrust vectoring mechanism (Version 2)

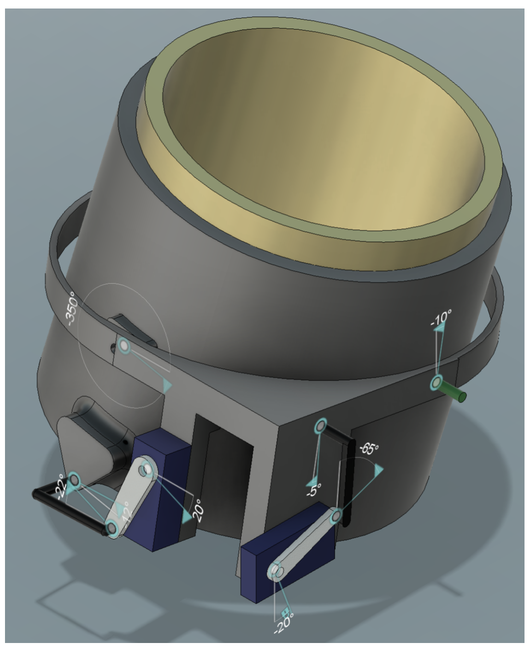

Figure 6.8 - Depiction of how servos affect motion (Version 2)

6.3 Final Drafted Model

For the final model, the dimensions of the actual engine and servo motors were taken into account. Due to the shape and placement of the two plates, the pivot points and connection points for the rods had very little room for adjustment. While the shape of the ring returned to a square shape, similar to the first model, this final model is more detailed and has a proper mounting for the first servo motor. The second servo motor did not have a mount built for it as it would need to be connected to the actual spacecraft to achieve the necessary motion. The final model can be seen in Figure 6.9, while Figure 6.10 and Figure 6.11 show the motion of the servo motors.

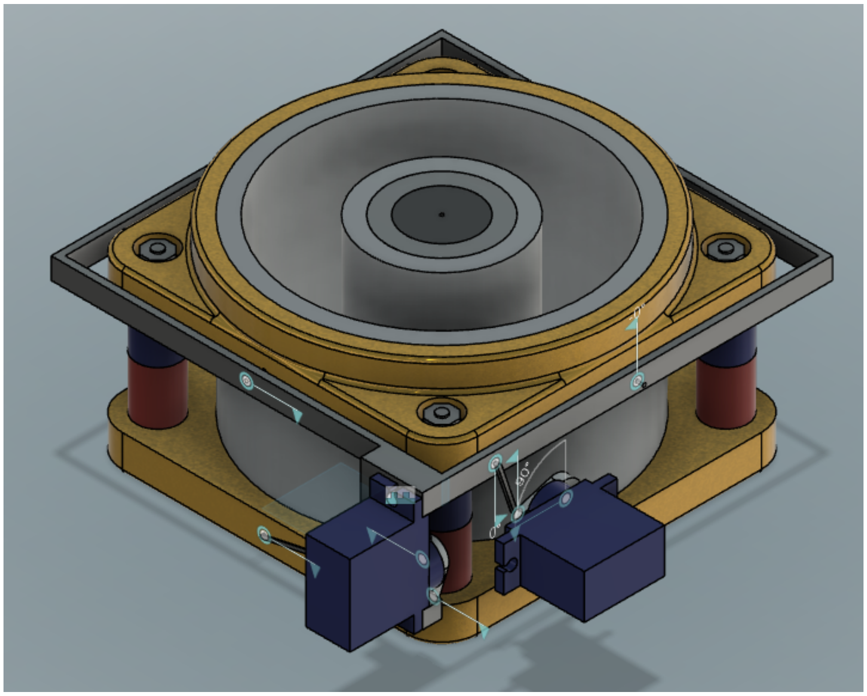

Figure 6.9 - Isometric view of Hall-effect thruster (Final version)

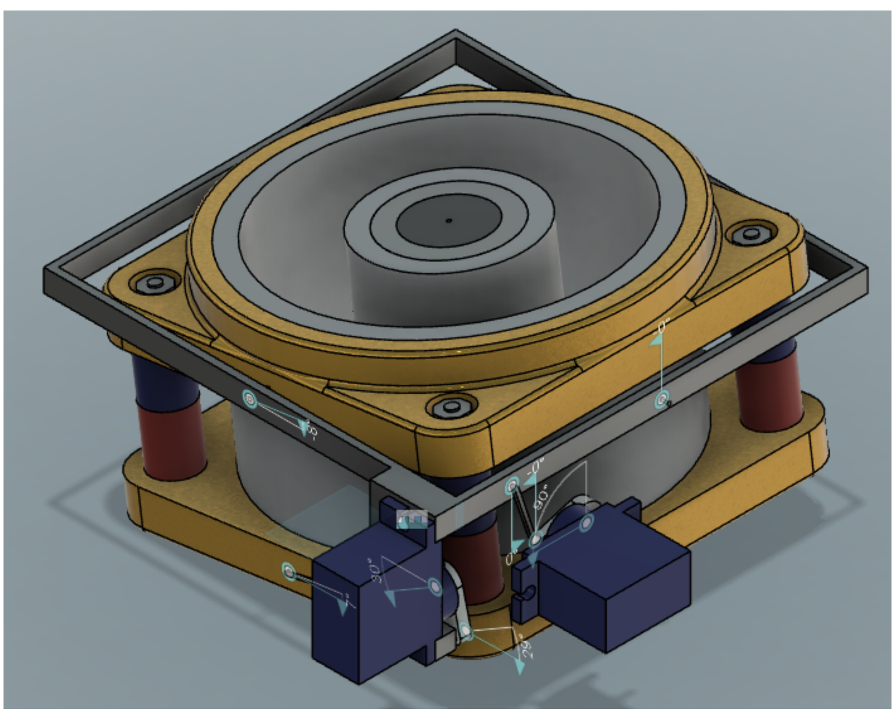

Figure 6.10 - Depiction of motion for first axis (Final version)

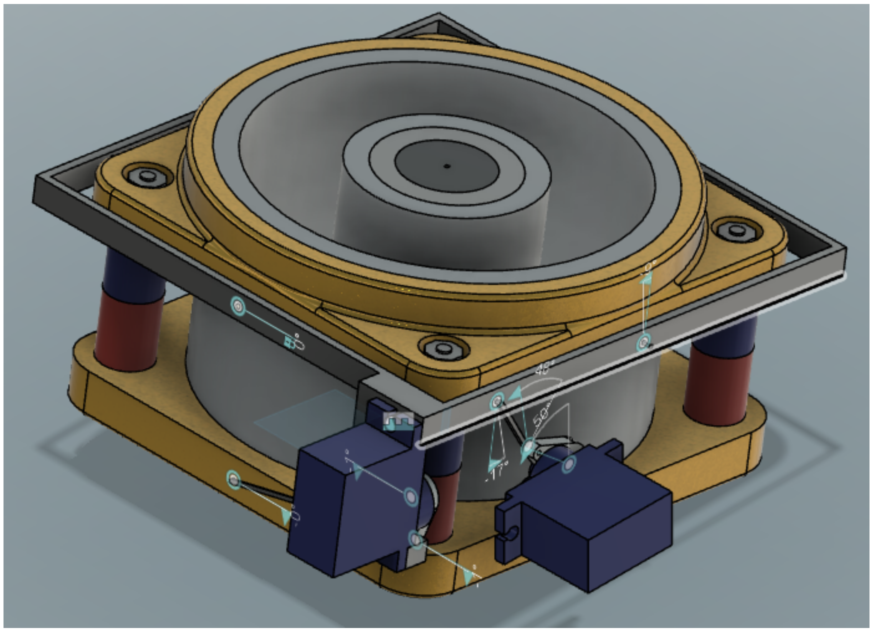

Figure 6.11 - Depiction of motion for second axis (Final version)

Conclusion

The goal of this project was to design a thrust-vectoring Hall-effect thruster as a baseline, within reasonable results. The work that was completed in this report was the first step for creating an electric propulsion system for spacecraft. The results and analysis prove that the Hall-effect thruster that was designed had acceptable data based on the calculations and assumptions. These results were compared to the range of design parameters taken from Goebel, Katz, and Mikellides book. Using the ANSYS Maxwell solver proved to be an efficient tool for analyzing the electric and magnetic fields. Lastly, the

The next steps for this project would be to compare the results obtained with other sources and reciprocate the data that was collected with Ansys Maxwell to further build understanding in the more complex design parameters of Hall-effect thrusters. Another step would be to create the control system for the thrust-vectoring mechanism, along with designing the wire harness to be able to build a complete thrust-vectoring Hall-effect thruster module that can be attached to small-scale spacecraft that want to use the technology for specific missions.

References

[1] Parker, K. I. and Folta, D. C., “Propulsion System,” Cubesat Handbook, Academic Press, 2021, pp. 283-301.
doi: 10.1016/B978-0-12-817884-3.00015-1

[2] Bramanti, C., Walker, R., Sutherland, O., Boswell, R., Charles, C., Fearn, D., Amo, J.G.D. and Orlandi, M., “The Innovative Dual-stage 4-Grid Ion Thruster Concept - Theory and Experimental Results,” 57th International Astronautical Congress, pp. C3-4.
doi: 10.2514/6.IAC-06-C4.4.07

[3] SpaceX, “Falcon 9.” *SpaceX*, https://www.spacex.com/vehicles/falcon-9/.

[4] Choueiri, E. Y., “The Efficient Future of Deep-Space Travel: Electric Rockets,” *Scientific American,* Vol. 299, No. 3, 2008.

[5] Jahn, Robert G., and Choueiri, Edgar Y., “Electric Propulsion,” *Encyclopedia of Physical Science and Technology*, 3rd ed., Vol.5, Academic Press, San Diego, CA, 2002, pp.125-141.
doi: 10.1016/B0-12-227410-5/00201-5

[6] Pultarova, T., “Ion Thruster Prototype Breaks Records in Tests, Could Send Humans to Mars,” *Space.com*, October 13, 2017.
https://www.space.com/38444-mars-thruster-design-breaks-records.html.

[7] Choueiri, Edgar Y., “New Dawn for Electric Rockets,” *Scientific American,* Vol. 300, No. 2, 2009, pp.58-65.
doi: 10.1038/scientificamerican0209-58

[8] Brown, D. W., “NASA’s ‘Nuclear Option’ May Be Crucial for Getting Humans to Mars,” *Scientific American,* January 27, 2022.

[9] Lockheed Martin, “Lockheed Martin Selected to Develop Nuclear-Powered Spacecraft,” *Lockheed Martin Corporation.*
https://news.lockheedmartin.com/2023-07-26-Lockheed-Martin-Selected-to-Develop-Nuclear-Powered-Spacecraft.

[10] NASA, “F-18 High Alpha Research Vehicle (HARV),” NASA Dryden Flight Research Center, Edwards, CA, August 14, 2009, https://www.nasa.gov/reference/f-18-harv/.

[11] Space Shuttle, "The Space Shuttle," NASA, accessed June 15, 2024, https://www.nasa.gov/reference/the-space-shuttle/.

[12] SpaceX**.** "Starship," *SpaceX*. accessed June 15, 2024, https://www.spacex.com/vehicles/starship/.

[13] Rayman M. D., "The Successful Conclusion of the Deep Space 1 Mission: Important Results Without a Flashy Title," *Space Technology,* Vol. 23, Nos. 2-3, 2003, pp. 185-196.

[14] Suseno, P. A. P., and Bahri, S., "Experimental Flight Control Function for Electronic Flight Control System of High Aspect Ratio Light Utility Aircraft," *AIP Conference Proceedings,* Vol. 2226, No. 1, 2020. doi: 10.1063/5.0002315

[15] Naji, N. A. M., and Stoica, A., "Design of a Robust Controller for the VEGA TVC Using the L1- Adaptive Control," *INCAS,* Vol. 15, No. 4, 2023, pp.175-185. doi: 10.13111/2066-8201.2023.15.4.15

[16] Deere, K. A., "Summary of Fluidic Thrust Vectoring Research Conducted at NASA Langley Research Center," *AIAA Applied Aerodynamics Conference*, Orlando, FL, June 2003, Paper No. AIAA 2003-3800.

[17] Goebel, D. M., Katz, I., and Mikellides, I. G., *Fundamentals of Electric Propulsion*, Wiley, 2nd ed., Hoboken, NJ, 2023, Chaps. 1, 2, 7.

[18] Eshel, T., "Thrust-Vectoring Upgrade for Typhoon Eurojet EJ200?" *Defense Update*, February 9, 2011. https://defense-update.com/20110209_typhoon_tvn.html.

[19] NASA/JPL, "Deep Space 1 Spacecraft Keeps Going and Going," *NASA Jet Propulsion Laboratory*, accessed June 18, 2024. https://www.jpl.nasa.gov/news/deep-space-1-spacecraft-keeps-going-and-going/.

[20] Microcontrollers Lab, "PID Controller Design Using Simulink MATLAB: Tutorial 3," *Microcontrollers Lab*, accessed June 18, 2024. https://microcontrollerslab.com/pid-controller-design-simulink/.

Appendix A: Magnetic and Electric Field Calculations Script

SJSU\Masters\Electric_and_Magnetic_Field.py

```python
import numpy as np

# Electric Field --------------------------------------------
def calculateElectricField(voltage, distance):
    return voltage / distance                          # Volts/meters

# Mangetic Field --------------------------------------------
def calculateMagneticField(radius, height, magnet2_position, evaluation_point):
    # Constants
    mu_0 = 4 * np.pi * 1e-7                            # Permeability of free space (T·m/A)
    Br = 1.1                                           # Remanent magnetic field (T)

    # Equations
    M = Br / mu_0                                      # Magnetization (A/m)
    v = np.pi * radius ** 2 * height                   # Volume of the magnet (m^3)
    moment = M * v                                     # Magnetic moment magnitude (A·m^2)

    # Magnetic moments
    m1 = np.array([0, 0, moment])                      # Magnet 1 moment (north up)
    m2 = np.array([0, 0, -moment])                     # Magnet 2 moment (north down)

    # Positions
    magnet1_position = np.array([0, 0, 0])             # Magnet 1 at (0, 0, 0)
    r1 = evaluation_point - magnet1_position           # Vector from dipole 1 to evaluation point
    r2 = evaluation_point - magnet2_position           # Vector from dipole 2 to evaluation point
    r1_mag = np.linalg.norm(r1)                        # Distance magnitude 2
    r2_mag = np.linalg.norm(r2)                        # Distance magnitude 2
    r1_unit = r1 / r1_mag                              # Unit vector 1
    r2_unit = r2 / r2_mag                              # Unit vector 2

    # Calculate Magnetic Fields
    B1 = (mu_0 / (4 * np.pi)) * ((3 * np.dot(m1, r1_unit) * r1_unit - m1) / r1_mag ** 3)
    B2 = (mu_0 / (4 * np.pi)) * ((3 * np.dot(m2, r2_unit) * r2_unit - m2) / r2_mag ** 3)
    B = B1 + B2
    B_mag = np.linalg.norm(B)
    return B1, B2, B, B_mag

# Calculate fields
E = calculateElectricField(100, 0.05)
B1, B2, B, B_mag = calculateMagneticField(0.006, 0.04, np.array([0.05657, 0, 0]), np.array([0.05657/2, 0, 0.02]))

# Display results
print(f'Electric Field: {E} V/m\n')
print(f'Magnetic field at evaluation point for:\n    Magnet 1: {B1} Tesla\n    Magnet 2: {B2} Tesla\n    Both magnets: {B} Tesla\n    Magnitude: {round(B_mag*1000, 4)} mTesla')
```

Appendix B: Hall-effect Thruster Parameter Calculations Script

SJSU\Masters\Thrust_Isp_Efficiency.py

```
import numpy as np

# Constants
q = 1.602e-19                                               # Charge of an electron C
# eta_i = .9
M_Xe = 131.3 * 1.66054e-27                                  # Xenon particle weight kg
# g = 9.81                                                    # gravity (Earth) m/s^2

# Assumptions
Q = 8e18                                                    # Particles/sec
Vb = 0.95*100                                               # Volts
Ib = 1.1                                                    # Amps
I_plus = Ib*.96                                             # Amps
I_plusplus = Ib - I_plus                                    # Amps
theta = 10                                                  # degrees
Po = 12 * 1.1                                               # [W] 100 Volts, 1 Amps

# Equations
alpha = (1+0.707*(I_plusplus/I_plus))/(I_plus + (I_plusplus/I_plus))
Ft = np.cos(10*np.pi/180)
gamma = alpha*Ft                                            #
m_doti = Ib*M_Xe/q                                          # kg/s
m_dotp = Q*M_Xe                                             # kg/s
# vi = np.sqrt(2*q*Vb/M_Xe)                                 # m/s
# ve = vi

# Thrust --------------------------------------------------------
T_Xenon = 1.65 * gamma * Ib * np.sqrt(Vb)                   # mNewtons
print(f'Thrust: {round(T_Xenon, 2)} mN')

# Specific Impulse ----------------------------------------------
a_m = (1 + 0.5*(I_plusplus/I_plus))/(I_plus + (I_plusplus/I_plus))
eta_m = m_doti/m_dotp

Isp_Xe = 123.6 * gamma * eta_m * np.sqrt(Vb)
print(f'Specific Impulse: {round(Isp_Xe, 2)} sec')

# Efficiency ----------------------------------------------------
eta_e = Ib*Vb/(Ib*Vb + Po)
eta_t = gamma**2 * eta_e * eta_m
print(f'Mass Utilization Efficiency: {round(eta_m, 4)}\nElectrical Efficiency: {round(eta_e, 4)}\nTotal Efficiency: {round(eta_t, 4)}')

# Power Dissipation
P_in = Po + 100*1.1                                         # Watts
P_dis = P_in*(1-eta_e)                                      # Watts
print(f'Power Dissipated: {round(P_dis, 2)} W')
```

Appendix C: Ansys Maxwell Magnetic Field Script

SJSU\Masters\Magnetic Field Ansys Maxwell Script.py

```python
# ----------------------------------------------
# Script Recorded by Ansys Electronics Desktop Student Version 2024.2.0
# 22:36:31  Dec 01, 2024
# ----------------------------------------------
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("Hall Thruster Final")
oProject.InsertDesign("Maxwell 3D", "Maxwell3DDesign3", "Magnetostatic", "")
oDesign = oProject.SetActiveDesign("Maxwell3DDesign3")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateCylinder(
    [
        "NAME:CylinderParameters",
        "XCenter:="     , "0mm",
        "YCenter:="     , "0mm",
        "ZCenter:="     , "0mm",
        "Radius:="      , "0.8mm",
        "Height:="      , "0.4mm",
        "WhichAxis:="       , "Z",
        "NumSides:="        , "0"
    ],
    [
        "NAME:Attributes",
        "Name:="        , "Cylinder1",
        "Flags:="       , "",
        "Color:="       , "(143 175 143)",
        "Transparency:="    , 0,
        "PartCoordinateSystem:=", "Global",
        "UDMId:="       , "",
        "MaterialValue:="   , "\"vacuum\"",
        "SurfaceMaterialValue:=", "\"\"",
        "SolveInside:="     , True,
        "ShellElement:="    , False,
        "ShellElementThickness:=", "0mm",
        "ReferenceTemperature:=", "20cel",
        "IsMaterialEditable:="  , True,
        "IsSurfaceMaterialEditable:=", True,
        "UseMaterialAppearance:=", False,
        "IsLightweight:="   , False
    ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder1:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Radius",
                    "Value:="       , "5mm"
                ],
                [
                    "NAME:Height",
                    "Value:="       , "40mm"
                ]
            ]
        ]
    ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.Copy(
    [
        "NAME:Selections",
        "Selections:="      , "Cylinder1"
    ])
oEditor.Paste()
oEditor.Paste()
oEditor.Paste()
```

```
oEditor.Paste()
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder2:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Center Position",
                    "X:="           , "40mm",
                    "Y:="           , "40mm",
                    "Z:="           , "0mm"
                ]
            ]
        ]
    ])
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder3:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Center Position",
                    "X:="           , "40mm",
                    "Y:="           , "-40mm",
                    "Z:="           , "0mm"
                ]
            ]
        ]
    ])
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder4:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Center Position",
                    "X:="           , "-40mm",
                    "Y:="           , "40mm",
                    "Z:="           , "0mm"
                ]
            ]
        ]
    ])
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder5:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Center Position",
                    "X:="           , "-40mm",
```

```
                    "Y:="           , "-40mm",
                    "Z:="           , "0mm"
                ]
            ]
        ]
    ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.AssignMaterial(
    [
        "NAME:Selections",
        "AllowRegionDependentPartSelectionForPMLCreation:=", True,
        "AllowRegionSelectionForPMLCreation:=", True,
        "Selections:="      , "Cylinder1"
    ],
    [
        "NAME:Attributes",
        "MaterialValue:="   , "\"NdFe30 North\"",
        "SolveInside:="     , True,
        "ShellElement:="    , False,
        "ShellElementThickness:=", "nan ",
        "ReferenceTemperature:=", "nan ",
        "IsMaterialEditable:="  , True,
        "IsSurfaceMaterialEditable:=", True,
        "UseMaterialAppearance:=", False,
        "IsLightweight:="   , False
    ])
oEditor.AssignMaterial(
    [
        "NAME:Selections",
        "AllowRegionDependentPartSelectionForPMLCreation:=", True,
        "AllowRegionSelectionForPMLCreation:=", True,
        "Selections:="      , "Cylinder2,Cylinder3,Cylinder4,Cylinder5"
    ],
    [
        "NAME:Attributes",
        "MaterialValue:="   , "\"NdFe30 North South\"",
        "SolveInside:="     , True,
        "ShellElement:="    , False,
        "ShellElementThickness:=", "nan ",
        "ReferenceTemperature:=", "nan ",
        "IsMaterialEditable:="  , True,
        "IsSurfaceMaterialEditable:=", True,
        "UseMaterialAppearance:=", False,
        "IsLightweight:="   , False
    ])
oEditor.CreateRegion(
    [
        "NAME:RegionParameters",
        "+XPaddingType:="   , "Percentage Offset",
        "+XPadding:="       , "50",
        "-XPaddingType:="   , "Percentage Offset",
        "-XPadding:="       , "50",
        "+YPaddingType:="   , "Percentage Offset",
        "+YPadding:="       , "50",
        "-YPaddingType:="   , "Percentage Offset",
        "-YPadding:="       , "50",
        "+ZPaddingType:="   , "Percentage Offset",
        "+ZPadding:="       , "50",
        "-ZPaddingType:="   , "Percentage Offset",
        "-ZPadding:="       , "50",
        [
            "NAME:BoxForVirtualObjects",
            [
                "NAME:LowPoint",
                1,
                1,
                1
            ],
            [
                "NAME:HighPoint",
                -1,
                -1,
                -1
            ]
        ]
    ],
```

```
    [
        "NAME:Attributes",
        "Name:="        , "Region",
        "Flags:="       , "Wireframe#",
        "Color:="       , "(143 175 143)",
        "Transparency:="    , 0,
        "PartCoordinateSystem:=", "Global",
        "UDMId:="       , "",
        "MaterialValue:="   , "\"vacuum\"",
        "SurfaceMaterialValue:=", "\"\"",
        "SolveInside:="     , True,
        "ShellElement:="    , False,
        "ShellElementThickness:=", "nan ",
        "ReferenceTemperature:=", "nan ",
        "IsMaterialEditable:="  , True,
        "IsSurfaceMaterialEditable:=", True,
        "UseMaterialAppearance:=", False,
        "IsLightweight:="   , False
    ])
oEditor.CreateCutplane(
    [
        "NAME:PlaneParameters",
        "PlaneBaseX:="      , "0mm",
        "PlaneBaseY:="      , "0mm",
        "PlaneBaseZ:="      , "0mm",
        "PlaneNormalX:="    , "40mm",
        "PlaneNormalY:="    , "-40mm",
        "PlaneNormalZ:="    , "0mm"
    ],
    [
        "NAME:Attributes",
        "Name:="        , "Plane1",
        "Color:="       , "(143 175 143)"
    ])
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("Magnetostatic",
    [
        "NAME:Setup1",
        "Enabled:="     , True,
        [
            "NAME:MeshLink",
            "ImportMesh:="      , False
        ],
        "MaximumPasses:="   , 15,
        "MinimumPasses:="   , 2,
        "MinimumConvergedPasses:=", 1,
        "PercentRefinement:="   , 30,
        "SolveFieldOnly:="  , False,
        "PercentError:="    , 1,
        "SolveMatrixAtLast:="   , True,
        "UseNonLinearIterNum:=" , False,
        "UseIterativeSolver:="  , False,
        "RelativeResidual:="    , 1E-06,
        "NonLinearResidual:="   , 0.001,
        "RelaxationFactor:="    , 1,
        "SmoothBHCurve:="   , False,
        [
            "NAME:MuOption",
            "MuNonLinearBH:="   , True
        ]
    ])
oModule = oDesign.GetModule("FieldsReporter")
oModule.CreateFieldPlot(
    [
        "NAME:Mag_B1",
        "SolutionName:="    , "Setup1 : LastAdaptive",
        "UserSpecifyName:=" , 0,
        "UserSpecifyFolder:="   , 0,
        "QuantityName:="    , "Mag_B",
        "PlotFolder:="      , "B",
        "StreamlinePlot:="  , False,
        "AdjacentSidePlot:="    , False,
        "FullModelPlot:="   , False,
        "IntrinsicVar:="    , "",
        "PlotGeomInfo:="    , [1,"Surface","CutPlane",1,"Plane1"],
        "FilterBoxes:="     , [0],
```

```
        [
            "NAME:PlotOnSurfaceSettings",
            "Filled:="      , False,
            "IsoValType:="      , "Tone",
            "AddGrid:="     , False,
            "MapTransparency:=" , True,
            "Refinement:="      , 0,
            "Transparency:="    , 0,
            "SmoothingLevel:="  , 0,
            "ShadingType:="     , 0,
            [
                "NAME:Arrow3DSpacingSettings",
                "ArrowUniform:="    , True,
                "ArrowSpacing:="    , 0,
                "MinArrowSpacing:=" , 0,
                "MaxArrowSpacing:=" , 0
            ],
            "GridColor:="       , [255,255,255]
        ],
        "EnableGaussianSmoothing:=", False,
        "SurfaceOnly:="     , False
    ], "Field")
oModule.CreateFieldPlot(
    [
        "NAME:B_Vector1",
        "SolutionName:="    , "Setup1 : LastAdaptive",
        "UserSpecifyName:=" , 0,
        "UserSpecifyFolder:="   , 0,
        "QuantityName:="    , "B_Vector",
        "PlotFolder:="      , "B",
        "StreamlinePlot:="  , False,
        "AdjacentSidePlot:="    , False,
        "FullModelPlot:="   , False,
        "IntrinsicVar:="    , "",
        "PlotGeomInfo:="    , [1,"Surface","CutPlane",1,"Plane1"],
        "FilterBoxes:="     , [0],
        [
            "NAME:PlotOnSurfaceSettings",
            "Filled:="      , False,
            "IsoValType:="      , "Tone",
            "AddGrid:="     , False,
            "MapTransparency:=" , True,
            "Refinement:="      , 0,
            "Transparency:="    , 0,
            "SmoothingLevel:="  , 0,
            "ShadingType:="     , 0,
            [
                "NAME:Arrow3DSpacingSettings",
                "ArrowUniform:="    , True,
                "ArrowSpacing:="    , 0,
                "MinArrowSpacing:=" , 0,
                "MaxArrowSpacing:=" , 0
            ],
            "GridColor:="       , [255,255,255]
        ],
        "EnableGaussianSmoothing:=", False,
        "SurfaceOnly:="     , False
    ], "Field")
oProject.Save()
oDesign.AnalyzeAll()
oModule.SetPlotFolderSettings("B",
    [
        "NAME:FieldsPlotSettings",
        "Real time mode:="  , True,
        [
            "NAME:ColorMapSettings",
            "ColorMapType:="    , "Spectrum",
            "SpectrumType:="    , "Rainbow",
            "UniformColor:="    , [127,255,255],
            "RampColor:="       , [255,127,127]
        ],
        [
            "NAME:Scale3DSettings",
            "unit:="        , 165,
            "m_nLevels:="       , 10,
            "minvalue:="        , 3.8E-05,
```

```
            "maxvalue:="        , 1.073112,
            "log:="         , False,
            "IntrinsicMin:="    , 3.75608506680939E-05,
            "IntrinsicMax:="    , 1.07311249916918,
            "LimitFieldValuePrecision:=", False,
            "FieldValuePrecisionDigits:=", 4,
            "dB:="          , False,
            "AnimationStaticScale:=", False,
            "ScaleType:="       , 1,
            "UserSpecifyValues:="   ,
[11,1,10.8999996185303,20.7999992370605,30.6999988555908,40.5999984741211,50.5,60.3999977111816,70.2999954223633,80.1999969482422,90.0999984741211,
            "ValueNumberFormatTypeAuto:=", 0,
            "ValueNumberFormatTypeScientific:=", False,
            "ValueNumberFormatWidth:=", 8,
            "ValueNumberFormatPrecision:=", 3
        ],
        [
            "NAME:Marker3DSettings",
            "MarkerType:="      , 0,
            "MarkerMapSize:="   , False,
            "MarkerMapColor:="  , False,
            "MarkerSize:="      , 0.25
        ],
        [
            "NAME:Arrow3DSettings",
            "ArrowType:="       , 1,
            "ArrowMapSize:="    , False,
            "ArrowMapColor:="   , True,
            "ShowArrowTail:="   , True,
            "ArrowSize:="       , 0.100000001490116,
            "ArrowMinMagnitude:="   , -0.498564072811303,
            "ArrowMaxMagnitude:="   , 1.61126245886348,
            "ArrowMagnitudeThreshold:=", 0,
            "ArrowMagnitudeFilteringFlag:=", False,
            "ArrowMinIntrinsicMagnitude:=", 0,
            "ArrowMaxIntrinsicMagnitude:=", 1
        ],
        [
            "NAME:DeformScaleSettings",
            "ShowDeformation:=" , True,
            "MinScaleFactor:="  , 0,
            "MaxScaleFactor:="  , 1,
            "DeformationScale:="    , 0,
            "ShowDeformationOutline:=", False
        ]
    ])
oModule.SetPlotFolderSettings("B",
    [
        "NAME:FieldsPlotSettings",
        "Real time mode:="  , True,
        [
            "NAME:ColorMapSettings",
            "ColorMapType:="    , "Spectrum",
            "SpectrumType:="    , "Rainbow",
            "UniformColor:="    , [127,255,255],
            "RampColor:="       , [255,127,127]
        ],
        [
            "NAME:Scale3DSettings",
            "unit:="        , 165,
            "m_nLevels:="       , 10,
            "minvalue:="        , 3.8E-05,
            "maxvalue:="        , 0.05,
            "log:="         , False,
            "IntrinsicMin:="    , 3.75608506680939E-05,
            "IntrinsicMax:="    , 1.07311249916918,
            "LimitFieldValuePrecision:=", False,
            "FieldValuePrecisionDigits:=", 4,
            "dB:="          , False,
            "AnimationStaticScale:=", False,
            "ScaleType:="       , 1,
            "UserSpecifyValues:="   ,
[11,1,10.8999996185303,20.7999992370605,30.6999988555908,40.5999984741211,50.5,60.3999977111816,70.2999954223633,80.1999969482422,90.0999984741211,
            "ValueNumberFormatTypeAuto:=", 0,
            "ValueNumberFormatTypeScientific:=", False,
            "ValueNumberFormatWidth:=", 8,
```

```
                "ValueNumberFormatPrecision:=", 3
            ],
            [
                "NAME:Marker3DSettings",
                "MarkerType:="      , 0,
                "MarkerMapSize:="   , False,
                "MarkerMapColor:="  , False,
                "MarkerSize:="      , 0.25
            ],
            [
                "NAME:Arrow3DSettings",
                "ArrowType:="       , 1,
                "ArrowMapSize:="    , False,
                "ArrowMapColor:="   , True,
                "ShowArrowTail:="   , True,
                "ArrowSize:="       , 0.100000001490116,
                "ArrowMinMagnitude:="   , -0.498564072811303,
                "ArrowMaxMagnitude:="   , 1.61126245886348,
                "ArrowMagnitudeThreshold:=", 0,
                "ArrowMagnitudeFilteringFlag:=", False,
                "ArrowMinIntrinsicMagnitude:=", 0,
                "ArrowMaxIntrinsicMagnitude:=", 1
            ],
            [
                "NAME:DeformScaleSettings",
                "ShowDeformation:=" , True,
                "MinScaleFactor:="  , 0,
                "MaxScaleFactor:="  , 1,
                "DeformationScale:="    , 0,
                "ShowDeformationOutline:=", False
            ]
    ])
oModule.SetPlotFolderSettings("B",
    [
        "NAME:FieldsPlotSettings",
        "Real time mode:="  , True,
        [
            "NAME:ColorMapSettings",
            "ColorMapType:="    , "Spectrum",
            "SpectrumType:="    , "Rainbow",
            "UniformColor:="    , [127,255,255],
            "RampColor:="       , [255,127,127]
        ],
        [
            "NAME:Scale3DSettings",
            "unit:="        , 165,
            "m_nLevels:="       , 10,
            "minvalue:="        , 3.8E-05,
            "maxvalue:="        , 0.03,
            "log:="         , False,
            "IntrinsicMin:="    , 3.75608506680939E-05,
            "IntrinsicMax:="    , 1.07311249916918,
            "LimitFieldValuePrecision:=", False,
            "FieldValuePrecisionDigits:=", 4,
            "dB:="          , False,
            "AnimationStaticScale:=", False,
            "ScaleType:="       , 1,
            "UserSpecifyValues:="   ,
[11,1,10.8999996185303,20.7999992370605,30.6999988555908,40.5999984741211,50.5,60.3999977111816,70.2999954223633,80.1999969482422,90.0999984741211,
            "ValueNumberFormatTypeAuto:=", 0,
            "ValueNumberFormatTypeScientific:=", False,
            "ValueNumberFormatWidth:=", 6,
            "ValueNumberFormatPrecision:=", 3
        ],
        [
            "NAME:Marker3DSettings",
            "MarkerType:="      , 0,
            "MarkerMapSize:="   , False,
            "MarkerMapColor:="  , False,
            "MarkerSize:="      , 0.25
        ],
        [
            "NAME:Arrow3DSettings",
            "ArrowType:="       , 1,
            "ArrowMapSize:="    , False,
            "ArrowMapColor:="   , True,
```

```
				"ShowArrowTail:="	, True,
				"ArrowSize:="		, 0.100000001490116,
				"ArrowMinMagnitude:="	, -0.498564072811303,
				"ArrowMaxMagnitude:="	, 1.61126245886348,
				"ArrowMagnitudeThreshold:=", 0,
				"ArrowMagnitudeFilteringFlag:=", False,
				"ArrowMinIntrinsicMagnitude:=", 0,
				"ArrowMaxIntrinsicMagnitude:=", 1
			],
			[
				"NAME:DeformScaleSettings",
				"ShowDeformation:="	, True,
				"MinScaleFactor:="	, 0,
				"MaxScaleFactor:="	, 1,
				"DeformationScale:="	, 0,
				"ShowDeformationOutline:=", False
			]
		])
oModule.SetPlotFolderSettings("B",
	[
		"NAME:FieldsPlotSettings",
		"Real time mode:="	, True,
		[
			"NAME:ColorMapSettings",
			"ColorMapType:="	, "Spectrum",
			"SpectrumType:="	, "Rainbow",
			"UniformColor:="	, [127,255,255],
			"RampColor:="		, [255,127,127]
		],
		[
			"NAME:Scale3DSettings",
			"unit:="		, 165,
			"m_nLevels:="		, 10,
			"minvalue:="		, 0.01,
			"maxvalue:="		, 0.03,
			"log:="			, False,
			"IntrinsicMin:="	, 3.75608506680939E-05,
			"IntrinsicMax:="	, 1.07311249916918,
			"LimitFieldValuePrecision:=", False,
			"FieldValuePrecisionDigits:=", 4,
			"dB:="			, False,
			"AnimationStaticScale:=", False,
			"ScaleType:="		, 1,
			"UserSpecifyValues:="	,
[11,1,10.8999996185303,20.7999992370605,30.6999988555908,40.5999984741211,50.5,60.3999977111816,70.2999954223633,80.1999969482422,90.0999984741211,
			"ValueNumberFormatTypeAuto:=", 0,
			"ValueNumberFormatTypeScientific:=", False,
			"ValueNumberFormatWidth:=", 6,
			"ValueNumberFormatPrecision:=", 3
		],
		[
			"NAME:Marker3DSettings",
			"MarkerType:="		, 0,
			"MarkerMapSize:="	, False,
			"MarkerMapColor:="	, False,
			"MarkerSize:="		, 0.25
		],
		[
			"NAME:Arrow3DSettings",
			"ArrowType:="		, 1,
			"ArrowMapSize:="	, False,
			"ArrowMapColor:="	, True,
			"ShowArrowTail:="	, True,
			"ArrowSize:="		, 0.100000001490116,
			"ArrowMinMagnitude:="	, -0.498564072811303,
			"ArrowMaxMagnitude:="	, 1.61126245886348,
			"ArrowMagnitudeThreshold:=", 0,
			"ArrowMagnitudeFilteringFlag:=", False,
			"ArrowMinIntrinsicMagnitude:=", 0,
			"ArrowMaxIntrinsicMagnitude:=", 1
		],
		[
			"NAME:DeformScaleSettings",
			"ShowDeformation:="	, True,
			"MinScaleFactor:="	, 0,
			"MaxScaleFactor:="	, 1,
```

```
            "DeformationScale:="    , 0,
            "ShowDeformationOutline:=", False
        ]
    ])

```

Appendix D: Ansys Maxwell Electric Field Script

SJSU\Masters\Electric Field Ansys Maxwell Script.py

```
# ----------------------------------------------
# Script Recorded by Ansys Electronics Desktop Student Version 2024.2.0
# 22:05:06  Dec 01, 2024
# ----------------------------------------------
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.NewProject()
oProject.InsertDesign("Maxwell 3D", "Maxwell3DDesign1", "Magnetostatic", "")
oDesign = oProject.SetActiveDesign("Maxwell3DDesign1")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateCylinder(
    [
        "NAME:CylinderParameters",
        "XCenter:="     , "0mm",
        "YCenter:="     , "0mm",
        "ZCenter:="     , "0mm",
        "Radius:="      , "0.8mm",
        "Height:="      , "0.2mm",
        "WhichAxis:="       , "Z",
        "NumSides:="        , "0"
    ],
    [
        "NAME:Attributes",
        "Name:="        , "Cylinder1",
        "Flags:="       , "",
        "Color:="       , "(143 175 143)",
        "Transparency:="    , 0,
        "PartCoordinateSystem:=", "Global",
        "UDMId:="       , "",
        "MaterialValue:="   , "\"vacuum\"",
        "SurfaceMaterialValue:=", "\"\"",
        "SolveInside:="     , True,
        "ShellElement:="    , False,
        "ShellElementThickness:=", "0mm",
        "ReferenceTemperature:=", "20cel",
        "IsMaterialEditable:="  , True,
        "IsSurfaceMaterialEditable:=", True,
        "UseMaterialAppearance:=", False,
        "IsLightweight:="   , False
    ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder1:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Radius",
                    "Value:="       , "40mm"
                ],
                [
                    "NAME:Height",
                    "Value:="       , "3mm"
                ]
            ]
        ]
    ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.Copy(
    [
        "NAME:Selections",
        "Selections:="      , "Cylinder1"
    ])
oEditor.Paste()
oEditor.Paste()
oEditor = oDesign.SetActiveEditor("3D Modeler")
```

```
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder2:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Radius",
                    "Value:="       , "20mm"
                ]
            ]
        ]
    ])
oEditor.ChangeProperty(
    [
        "NAME:AllTabs",
        [
            "NAME:Geometry3DCmdTab",
            [
                "NAME:PropServers",
                "Cylinder3:CreateCylinder:1"
            ],
            [
                "NAME:ChangedProps",
                [
                    "NAME:Radius",
                    "Value:="       , "10mm"
                ],
                [
                    "NAME:Center Position",
                    "X:="           , "0mm",
                    "Y:="           , "0mm",
                    "Z:="           , "50mm"
                ]
            ]
        ]
    ])
oDesign.SetSolutionType("Electrostatic")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateRegion(
    [
        "NAME:RegionParameters",
        "+XPaddingType:="   , "Percentage Offset",
        "+XPadding:="       , "50",
        "-XPaddingType:="   , "Percentage Offset",
        "-XPadding:="       , "50",
        "+YPaddingType:="   , "Percentage Offset",
        "+YPadding:="       , "50",
        "-YPaddingType:="   , "Percentage Offset",
        "-YPadding:="       , "50",
        "+ZPaddingType:="   , "Percentage Offset",
        "+ZPadding:="       , "50",
        "-ZPaddingType:="   , "Percentage Offset",
        "-ZPadding:="       , "50",
        [
            "NAME:BoxForVirtualObjects",
            [
                "NAME:LowPoint",
                1,
                1,
                1
            ],
            [
                "NAME:HighPoint",
                -1,
                -1,
                -1
            ]
        ]
    ],
    [
```

```
        "NAME:Attributes",
        "Name:="        , "Region",
        "Flags:="       , "Wireframe#",
        "Color:="       , "(143 175 143)",
        "Transparency:="    , 0,
        "PartCoordinateSystem:=", "Global",
        "UDMId:="       , "",
        "MaterialValue:="   , "\"vacuum\"",
        "SurfaceMaterialValue:=", "\"\"",
        "SolveInside:="     , True,
        "ShellElement:="    , False,
        "ShellElementThickness:=", "nan ",
        "ReferenceTemperature:=", "nan ",
        "IsMaterialEditable:="  , True,
        "IsSurfaceMaterialEditable:=", True,
        "UseMaterialAppearance:=", False,
        "IsLightweight:="   , False
    ])
oEditor.Subtract(
    [
        "NAME:Selections",
        "Blank Parts:="     , "Cylinder1",
        "Tool Parts:="      , "Cylinder2"
    ],
    [
        "NAME:SubtractParameters",
        "KeepOriginals:="   , False,
        "TurnOnNBodyBoolean:="  , True
    ])
oEditor.AssignMaterial(
    [
        "NAME:Selections",
        "AllowRegionDependentPartSelectionForPMLCreation:=", True,
        "AllowRegionSelectionForPMLCreation:=", True,
        "Selections:="      , "Cylinder1,Cylinder3"
    ],
    [
        "NAME:Attributes",
        "MaterialValue:="   , "\"tungsten\"",
        "SolveInside:="     , True,
        "ShellElement:="    , False,
        "ShellElementThickness:=", "nan ",
        "ReferenceTemperature:=", "nan ",
        "IsMaterialEditable:="  , True,
        "IsSurfaceMaterialEditable:=", True,
        "UseMaterialAppearance:=", False,
        "IsLightweight:="   , False
    ])
oModule = oDesign.GetModule("BoundarySetup")
oModule.AssignVoltage(
    [
        "NAME:Voltage1",
        "Objects:="     , ["Cylinder1"],
        "Voltage:="     , "100V",
        "CoordinateSystem:="    , ""
    ])
oModule.AssignVoltage(
    [
        "NAME:Voltage2",
        "Objects:="     , ["Cylinder3"],
        "Voltage:="     , "0V",
        "CoordinateSystem:="    , ""
    ])
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("Electrostatic",
    [
        "NAME:Setup1",
        "Enabled:="     , True,
        [
            "NAME:MeshLink",
            "ImportMesh:="      , False
        ],
        "MaximumPasses:="   , 15,
        "MinimumPasses:="   , 2,
        "MinimumConvergedPasses:=", 1,
        "PercentRefinement:="   , 30,
```

```
        "SolveFieldOnly:="  , False,
        "PercentError:="    , 0.5,
        "SolveMatrixAtLast:="   , True,
        "UseNonLinearIterNum:=" , False,
        "UseIterativeSolver:="  , False,
        "RelativeResidual:="    , 1E-06,
        "NonLinearResidual:="   , 0.001,
        "RelaxationFactor:="    , 1
    ])
oModule = oDesign.GetModule("FieldsReporter")
oModule.CreateFieldPlot(
    [
        "NAME:Mag_E1",
        "SolutionName:="    , "Setup1 : LastAdaptive",
        "UserSpecifyName:=" , 0,
        "UserSpecifyFolder:="   , 0,
        "QuantityName:="    , "Mag_E",
        "PlotFolder:="      , "E",
        "StreamlinePlot:="  , False,
        "AdjacentSidePlot:="    , False,
        "FullModelPlot:="   , False,
        "IntrinsicVar:="    , "",
        "PlotGeomInfo:="    , [1,"Surface","CutPlane",1,"Global:XZ"],
        "FilterBoxes:="     , [0],
        [
            "NAME:PlotOnSurfaceSettings",
            "Filled:="      , False,
            "IsoValType:="      , "Tone",
            "AddGrid:="     , False,
            "MapTransparency:=" , True,
            "Refinement:="      , 0,
            "Transparency:="    , 0,
            "SmoothingLevel:="  , 0,
            "ShadingType:="     , 0,
            [
                "NAME:Arrow3DSpacingSettings",
                "ArrowUniform:="    , True,
                "ArrowSpacing:="    , 0,
                "MinArrowSpacing:=" , 0,
                "MaxArrowSpacing:=" , 0
            ],
            "GridColor:="       , [255,255,255]
        ],
        "EnableGaussianSmoothing:=", False,
        "SurfaceOnly:="     , False
    ], "Field")
oModule.CreateFieldPlot(
    [
        "NAME:E_Vector1",
        "SolutionName:="    , "Setup1 : LastAdaptive",
        "UserSpecifyName:=" , 0,
        "UserSpecifyFolder:="   , 0,
        "QuantityName:="    , "E_Vector",
        "PlotFolder:="      , "E",
        "StreamlinePlot:="  , False,
        "AdjacentSidePlot:="    , False,
        "FullModelPlot:="   , False,
        "IntrinsicVar:="    , "",
        "PlotGeomInfo:="    , [1,"Surface","CutPlane",1,"Global:XZ"],
        "FilterBoxes:="     , [0],
        [
            "NAME:PlotOnSurfaceSettings",
            "Filled:="      , False,
            "IsoValType:="      , "Tone",
            "AddGrid:="     , False,
            "MapTransparency:=" , True,
            "Refinement:="      , 0,
            "Transparency:="    , 0,
            "SmoothingLevel:="  , 0,
            "ShadingType:="     , 0,
            [
                "NAME:Arrow3DSpacingSettings",
                "ArrowUniform:="    , True,
                "ArrowSpacing:="    , 0,
                "MinArrowSpacing:=" , 0,
                "MaxArrowSpacing:=" , 0
```

```
            ],
            "GridColor:="       , [255,255,255]
        ],
        "EnableGaussianSmoothing:=", False,
        "SurfaceOnly:="     , False
    ], "Field")
oProject.SaveAs("C:\\Users\\hyydd\\OneDrive\\Documents\\Ansoft\\Hall Thruster Final.aedt", True)
oDesign.AnalyzeAll()
oModule.SetPlotFolderSettings("E",
    [
        "NAME:FieldsPlotSettings",
        "Real time mode:="  , True,
        [
            "NAME:ColorMapSettings",
            "ColorMapType:="    , "Spectrum",
            "SpectrumType:="    , "Rainbow",
            "UniformColor:="    , [127,255,255],
            "RampColor:="       , [255,127,127]
        ],
        [
            "NAME:Scale3DSettings",
            "unit:="        , 154,
            "m_nLevels:="       , 10,
            "minvalue:="        , 0,
            "maxvalue:="        , 13233.764,
            "log:="         , False,
            "IntrinsicMin:="    , 0,
            "IntrinsicMax:="    , 13233.7637244721,
            "LimitFieldValuePrecision:=", False,
            "FieldValuePrecisionDigits:=", 4,
            "dB:="          , False,
            "AnimationStaticScale:=", False,
            "ScaleType:="       , 1,
            "UserSpecifyValues:="   ,
[11,1,10.8999996185303,20.7999992370605,30.6999988555908,40.5999984741211,50.5,60.3999977111816,70.2999954223633,80.1999969482422,90.0999984741211,
            "ValueNumberFormatTypeAuto:=", 0,
            "ValueNumberFormatTypeScientific:=", False,
            "ValueNumberFormatWidth:=", 9,
            "ValueNumberFormatPrecision:=", 3
        ],
        [
            "NAME:Marker3DSettings",
            "MarkerType:="      , 0,
            "MarkerMapSize:="   , False,
            "MarkerMapColor:="  , False,
            "MarkerSize:="      , 0.25
        ],
        [
            "NAME:Arrow3DSettings",
            "ArrowType:="       , 1,
            "ArrowMapSize:="    , False,
            "ArrowMapColor:="   , True,
            "ShowArrowTail:="   , True,
            "ArrowSize:="       , 0.100000001490116,
            "ArrowMinMagnitude:="   , -0.5,
            "ArrowMaxMagnitude:="   , 9547.33729221799,
            "ArrowMagnitudeThreshold:=", 0,
            "ArrowMagnitudeFilteringFlag:=", False,
            "ArrowMinIntrinsicMagnitude:=", 0,
            "ArrowMaxIntrinsicMagnitude:=", 1
        ],
        [
            "NAME:DeformScaleSettings",
            "ShowDeformation:=" , True,
            "MinScaleFactor:="  , 0,
            "MaxScaleFactor:="  , 1,
            "DeformationScale:="    , 0,
            "ShowDeformationOutline:=", False
        ]
    ])
oModule.SetPlotFolderSettings("E",
    [
        "NAME:FieldsPlotSettings",
        "Real time mode:="  , True,
        [
            "NAME:ColorMapSettings",
```

```
            "ColorMapType:="    , "Spectrum",
            "SpectrumType:="    , "Rainbow",
            "UniformColor:="    , [127,255,255],
            "RampColor:="       , [255,127,127]
        ],
        [
            "NAME:Scale3DSettings",
            "unit:="        , 154,
            "m_nLevels:="       , 10,
            "minvalue:="        , 0,
            "maxvalue:="        , 3000,
            "log:="         , False,
            "IntrinsicMin:="    , 0,
            "IntrinsicMax:="    , 13233.7637244721,
            "LimitFieldValuePrecision:=", False,
            "FieldValuePrecisionDigits:=", 4,
            "dB:="          , False,
            "AnimationStaticScale:=", False,
            "ScaleType:="       , 1,
            "UserSpecifyValues:="   ,
[11,1,10.8999996185303,20.7999992370605,30.6999988555908,40.5999984741211,50.5,60.3999977111816,70.2999954223633,80.1999969482422,90.0999984741211,
            "ValueNumberFormatTypeAuto:=", 0,
            "ValueNumberFormatTypeScientific:=", False,
            "ValueNumberFormatWidth:=", 9,
            "ValueNumberFormatPrecision:=", 3
        ],
        [
            "NAME:Marker3DSettings",
            "MarkerType:="      , 0,
            "MarkerMapSize:="   , False,
            "MarkerMapColor:="  , False,
            "MarkerSize:="      , 0.25
        ],
        [
            "NAME:Arrow3DSettings",
            "ArrowType:="       , 1,
            "ArrowMapSize:="    , False,
            "ArrowMapColor:="   , True,
            "ShowArrowTail:="   , True,
            "ArrowSize:="       , 0.100000001490116,
            "ArrowMinMagnitude:="   , -0.5,
            "ArrowMaxMagnitude:="   , 9547.33729221799,
            "ArrowMagnitudeThreshold:=", 0,
            "ArrowMagnitudeFilteringFlag:=", False,
            "ArrowMinIntrinsicMagnitude:=", 0,
            "ArrowMaxIntrinsicMagnitude:=", 1
        ],
        [
            "NAME:DeformScaleSettings",
            "ShowDeformation:=" , True,
            "MinScaleFactor:="  , 0,
            "MaxScaleFactor:="  , 1,
            "DeformationScale:="    , 0,
            "ShowDeformationOutline:=", False
        ]
    ])

```