



**SHEET VELOCITY MEASUREMENTS USING
LASER ABSORPTION SPECTROSCOPY IN A
XENON HALL EFFECT THRUSTER PLUME**

THESIS

Avery W. Leonard, 2d Lt, USAF
AFIT-ENY-MS-19-M-227

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-MS-19-M-227

SHEET VELOCITY MEASUREMENTS USING
LASER ABSORPTION SPECTROSCOPY IN A
XENON HALL EFFECT THRUSTER PLUME

THESIS

Presented to the Faculty
Department of Aeronautics and Astronautics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Aeronautical Engineering

Avery W. Leonard, BSAE

2d Lt, USAF

March 7, 2019

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENY-MS-19-M-227

SHEET VELOCITY MEASUREMENTS USING
LASER ABSORPTION SPECTROSCOPY IN A
XENON HALL EFFECT THRUSTER PLUME

THESIS

Avery W. Leonard, BSAE
2d Lt, USAF

Committee Membership:

Dr. Carl Hartsfield
Chair

Dr. William Hargus
Member

Maj David Liu, PhD
Member

Abstract

A new laser absorption spectroscopy (LAS) velocimetry system, designed to obtain 2D planar velocity fields for ionized Xenon in the plume of a Hall effect thruster by probing the transition of Xe II at 834.72 nm, was implemented at the Air Force Institute of Technology (AFIT) Space Propulsion Analysis and System Simulator (SPASS) Lab vacuum chamber. A single horizontal laser sheet was used to probe singly-ionized Xenon in the plume of a Busek BHT-600 Hall thruster and obtain a histogram of estimated axial velocity, to validate the system. Similar velocities to those obtained by an earlier intrusive characterization of the thruster were observed.

Hall thrusters can be characterized by analysis of their plumes; however, many analysis methods require intrusive probes. Intrusive measurements, by their nature, affect the flow of the exhaust plume and may obtain inaccurate results, or results not representative of the reality of operating in orbit. They are also subject to probe erosion by ion bombardment, limiting probe lifetimes. For these reasons non-intrusive measurements are desirable. LAS provides a method to take these measurements in a non-intrusive manner.

This new system captures linear absorption data using cameras with a resolution of 2048 pixels at six different viewpoints to reconstruct the velocity field. In future research, simultaneous full-field measurements will allow observation of time-resolved velocity data of ions at all points in the plume at rates of up to 1 kHz. Hall thrusters often have different performance characteristics between laboratory vacuum chamber characterizations and actual on-orbit performance. Implementation of this system provides the first of two complimentary systems which will be used for complete particle analysis of Hall thruster plumes.

For my fiancée, who dealt with my many late nights and provided unwavering support throughout this research.

Acknowledgements

I would like to express thanks to my advisor, Dr. Carl Hartsfield, for his support throughout this research. I would also like to recognize Jamie Smith and Josh Dewitt for their efforts troubleshooting in the lab, as well as the employees at Spectral Energies who offered assistance. Further thanks goes to Greg Smith for his last-minute help in securing a magnetometer and wavemeter, and Dr. Marc Polanka for going above and beyond to review this document throughout its development and providing many helpful suggestions. Finally, I would like to acknowledge Dr. Mitat Birkan of the Air Force Office of Scientific Research for providing funding and support for this research.

Avery W. Leonard

Contents

	Page
Abstract	iv
Acknowledgements	vi
List of Figures	x
List of Tables	xiv
List of Symbols	xv
List of Abbreviations	xxi
I. Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Scope	3
1.4 Objective	4
II. Background	6
2.1 Rocket Fundamentals	6
2.2 Plasma Fundamentals	9
2.3 Electric Propulsion	15
2.3.1 Electrostatic propulsion	16
2.4 Hall Thrusters	19
2.4.1 Hollow Cathodes	23
2.4.2 Lifetimes	26
2.4.3 Plumes	26
2.5 Laser Diagnostics	27
2.5.1 Laser Fundamentals	28
2.5.2 Laser Absorption Spectroscopy	32
2.5.3 Laser Induced Fluorescence	34
2.6 Prior Experiments at AFIT	35
2.6.1 Busek 600W Hall thruster characterization	35
2.6.2 LIF analysis of Busek 200W Hall thruster plume	36
III. Methodology	37
3.1 Laboratory	37
3.1.1 Pumps and Pressure Gauges	38
3.1.2 Propellant Supply	40
3.1.3 Traverses	42

	Page
3.2 Thruster	43
3.2.1 Busek BHT-600 Hall Thruster	43
3.2.2 Power Processing Unit and Control System	44
3.2.3 Busek BHC-1500 Hollow Cathode	47
3.2.4 Wiring Setup	49
3.3 Instrumentation	50
3.3.1 Diode Laser and Laser Controllers	51
3.3.2 Fiber Coupling and Free-Beam Path	52
3.3.3 Cameras	57
3.3.4 Beam Path Alignment	58
3.3.5 Surveyed Region	59
3.3.6 Data Collection	61
3.4 Test Conditions	63
3.5 Uncertainty Analysis	65
IV. Results	66
4.1 Post-processing	67
4.2 Velocity Distributions	77
V. Conclusions	88
5.1 Significance	88
5.2 Future Work	89
Appendix A. Thruster Operation Guide	93
Appendix B. LAS System Operation Guide	99
Appendix C. MATLAB Code	103
3.1 multiprocessdata.m	103
3.2 analyzedata.m	105
3.3 filefixer.m	114
3.4 selectfiles.m	117
3.5 importother.m	118
3.6 importdata.m	120
3.7 findUncertainty.m	120
3.8 findPossibleVelsFromExB.m	129
3.9 findAxialVelocityForOffAngle.m	129
3.10 getAveragedAbsorbanceByVelocityBin.m	130
3.11 getAveragedNumberDensities.m	130
3.12 pixel2mm.m	132
3.13 getAveragedDAQFromSweep1.m	133
3.14 miscCalculations.m	133
3.15 plotAveragedAbsorbanceByVelocityBin.m	134

	Page
3.16 plotThingsFiltered.m	139
Bibliography	168

List of Figures

Figure		Page
1	Production of thrust from a generic rocket propulsion system.....	7
2	Larmor radius and particle drift for positively and negatively charged particles.....	12
3	Plasma potential near the boundary.	13
4	Cross-section of a typical Hall thruster.	20
5	Particle paths in crossed E and B fields.	22
6	Hollow cathode geometry.	24
7	Hollow cathode erosion, with visible degradation of the orifice and cathode keeper.	25
8	Illustration of an absorption line profile.	31
9	Illustration of a typical LAS setup.	33
10	The SPASS Lab vacuum chamber.	38
11	Extorr’s VacuumPlus software, used with the Residual Gas Analyzer to monitor partial pressures inside the chamber.	39
12	Propellant supply system.....	41
13	Front traverse, used for mounting the thruster and adjusting which portion of the plume is imaged.	42
14	Thruster connection diagram. Component sizes and locations not to scale.	43
15	BHT-600 Thruster.	44
16	Busek BPU-600 control software, used to operate the thruster.....	45
17	Agilent 6038A power supply, used to power the magnets.....	46

Figure		Page
18	Matsusada REK650-2.5 power supply, used to power the anode. Also pictured, MKS Type 247 mass flow readout.	46
19	Matsusada REK650-2.5 power supply, used to power the cathode keeper, and Sorensen DCS33-33E power supply, used to power the cathode heater.	47
20	Busek BHT-1500 dispenser cathode, a. Rear view mounted to thruster, showing propellant and electrical connections, b. Side view mounted to thruster, installed in vacuum chamber.	48
21	Interior wiring setup for Busek BHT-600 Hall thruster.	49
22	Simplified LAS system schematic.	50
23	Sacher Lasertechnik laser amplifiers.	51
24	Sacher Lasertechnik Tec 420 diode laser with cover removed.	51
25	Scan profile for the Sacher Lasertechnik diode laser tuned to 834.72 nm	52
26	Custom fiber pass-through, exterior, with strain relief.	53
27	Interior strain relief.	53
28	One of the six optic assemblies.	54
29	Ring of six collimator and lens assemblies mounted to Thorlabs 66mm dovetail rails, surrounding the imaging plane.	55
30	80/20 mounting ring with optical plate for mounting mirrors, circled in white, which direct the beam through the red-circled viewing ports.	55
31	Sheet path for the horizontal laser sheet used to probe the Hall thruster plume.	56
32	Camera assemblies, consisting of a custom 3D-printed housing for three lenses and Basler raL2048-80km cameras, a. Side-mounted camera assembly, b. Top-mounted camera assembly.	57

Figure		Page
33	Modified camera assembly, with lower camera removed and cage attached.	58
34	Tabletop setup for magnetic field measurements.	59
35	Mounted Busek BHT-600 Hall thruster with BHC-1500 hollow cathode, and LAS system optic assembly, viewed from the rear.	60
36	Mounted Busek BHT-600 Hall thruster with BHC-1500 hollow cathode, and LAS system optic assembly, viewed from the side.	60
37	Bristol Model 621 wavemeter, fiber-coupled to the LAS system.	62
38	Averaged camera reference over one sweep of 160 camera frames.	68
39	Averaged photodiode reference over one sweep of 1000 DAQ frames.	68
40	Bessel function of the first kind, generated using <i>besselj</i> with MATLAB®.	69
41	Averaged photodiode signal from data set 1 over one sweep of 1000 DAQ frames, presented as photodiode intensity as a function of the time since the start of the sweep, with the laser wavelength also as a function of time since the start of the sweep.	70
42	Relative absorption as a function of estimated axial velocity, calculated from Doppler-shifted wavelength, using data processed with both normal filtering methods and zero-phase filtering.	72
43	Ion current density in the plume of a Busek BHT-600 Hall thruster, operating on Xenon propellant at 300V discharge, at various axial locations (cm).	73
44	$E \times B$ probe collector current with visible peaks relating to Xenon ion species in the plume of a Busek BHT-600 Hall thruster, operating on Xenon propellant at 300V discharge.	75

Figure		Page
45	Ion current density, which is analogous to absorption, as a function of reported axial velocity.	76
46	Average velocity at each sweep weighted by relative absorbance, for each pixel, for data sets 1 through 4, from the top of the plume down 120mm.	78
47	Average relative absorbance at each sweep, for each pixel, for data sets 1 through 4, from the top of the plume down 120mm.	79
48	Average relative absorbance and average velocity, for each pixel, for data sets 1 through 4, from the top of the plume down 120mm.	80
49	Average relative absorbance at each pixel, for data sets 1 through 4, from the top of the plume down 120mm.	82
50	Average velocity at each pixel, for data sets 1 through 4, from the top of the plume down 120mm.	82
51	Standard deviation in km/s of weighted average velocity at each pixel, for data sets 1 through 4, from the top of the plume down 120mm. Standard deviations above 1 km/s are not displayed; values of up to 4,270 km/s were observed.	83
52	Time-averaged absorbance at each velocity bin, selected pixels, for data sets 1 through 4, from the top of the plume down 120mm.	85
53	Standard deviation of relative absorbance at each pixel, for data sets 1 through 4, from the top of the plume down 120mm.	86

List of Tables

Table		Page
1	Operating parameters for various thrusters.	16
2	Nominal performance parameters for the Busek 600W Hall thruster.	36
3	Set thruster operating conditions.	63
4	Determined thruster operating conditions.	63
5	Laser operating conditions.....	64
6	LAS system performance characteristics.	65
7	Ionized Xenon species fractions for discharge at 300V.	76

List of Symbols

Symbol	Page
T	Thrust 7
$\dot{m}$	Mass Flow Rate 7
u_{ex}	Exit Velocity 7
p_{ex}	Exit Pressure 7
p_0	Ambient Pressure 7
A_{ex}	Nozzle Exit Area 7
m	Mass, Instantaneous 8
u	Velocity 8
m_0	Mass, Initial 8
t	Burn Time 8
Δv	Change in Velocity 8
m_p	Mass, Propellant 8
m_d	Mass, Dry 8
I_{sp}	Specific Impulse 8
g_0	Standard Gravitational Acceleration 9
P_{jet}	Jet Power 9
T_e	Electron Temperature 10
T_i	Ion Temperature 10
n_i	Ion Number Density 10
n_e	Electron Number Density 10
E	Electric Field 11
B	Magnetic Field 11

Symbol		Page
ϵ_0	Permittivity of Free Space	11
μ_0	Permeability of Free Space	11
ρ	Charge Density	11
J	Current Density	11
Z	Charge State	11
e	Elementary Charge	11
u_i	Ion Velocity	11
u_e	Electron Velocity	11
F	Force	11
m_c	Mass, Charged Particle	11
q	Charge	11
u_c	Velocity, Charged Particle	11
ω_c	Cyclotron Frequency	11
r_L	Larmor Radius	11
$u_\perp$	Perpendicular Component of Velocity	11
$V_\perp$	Perpendicular Component of Electric Field	11
u_{drift}	Drift Velocity	12
ω_p	Electron Plasma Frequency	12
Ω_p	Ion Plasma Frequency	12
m_e	Electron Mass	12
M_i	Ion Mass	12
u_a	Ion Acoustic Velocity	13
k	Boltzmann's Constant	13
λ_D	Debye Length	13

Symbol	Page
n_0	Equilibrium Electron Number Density 13
λ	Mean Free Path 14
σ	Particle Cross-Sectional Area 14
n_a	Neutral Particle Number Density 14
τ	Mean Time Between Particle Collisions 14
ν	Particle Collision Frequency 14
$D_{\perp}$	Cross-Field Diffusion Coefficient 14
Ω_e	Electron Hall Parameter 15
D	Standard Diffusion Coefficient 15
$\dot{m}_i$	Ion Mass Flow Rate 17
u_i	Ion Velocity 17
V_b	Acceleration Voltage 17
γ	Thrust Correction Coefficient 17
$\frac{I^{++}}{I^{+}}$	Doubly-Ionized to Singly-Ionized Ratio 17
θ	Average Half-Angle Beam Divergence 17
η_m	Mass Utilization Efficiency 18
$\dot{m}_p$	Propellant Mass Flow Rate 18
α_m	Mass Utilization Correction Coefficient 18
P_b	Beam Power 19
P_T	Input Power 19
P_d	Power Required for Ion Production 19
I_b	Ion Beam Current 19
P_{in}	Total Electrical Power Consumed 19
B_r	Magnetic Field, Radial Component 20

Symbol		Page
E_z	Electric Field, Axial Component	20
I_H	Hall Current	21
w	Plasma Width	21
V_d	Discharge Voltage	21
L	Characteristic Length	21
u_{th}	Thermal Velocity	21
r_e	Electron Larmor Radius	21
r_i	Ion Larmor Radius	21
D_B	Bohm Diffusion Coefficient	22
λ_i	Ionization Mean Free Path	23
u_n	Neutral Particle Velocity	23
σ_i	Ionization Cross-Section	23
$\dot{m}_a$	Propellant Mass Flow Rate Through Anode	23
$\dot{m}_c$	Propellant Mass Flow Rate Through Cathode	23
I_c	Cathode Discharge Current	24
$u_{e,d}$	Electron Drift Velocity	24
A_o	Orifice Cross-Sectional Area	24
p_i	Ion Momentum	27
F_m	Momentum Transfer Per Unit Area	27
A_c	Surface Accomodation Coefficient	27
N	Population Density	28
E_k	Higher Energy State	28
$N(E_k)$	Population Density at Higher Energy State	28
$N(E_i)$	Population Density at Lower Energy State	28

Symbol		Page
$\alpha(\nu)$	Absorption Coefficient	29
N_i	Population Density of Arbitrary Energy State “i”	29
N_k	Population Density of Arbitrary Energy State “k”	29
g_i	Statistical Weight of Energy State “i”	29
g_k	Statistical Weight of Energy State “k”	29
σ_{ik}	Absorption Cross-Section	29
ν	Absorption Frequency	29
L_m	Mirror Separation Distance	29
R	Reflectivity	29
γ_r	Intensity Loss Coefficient	29
I	Intensity	29
I_i	Initial Intensity	29
ΔN	Population Difference	30
ΔN_{thr}	Population Difference Threshold for Amplification	30
h	Photon	30
ν_{ik}	Frequency at Maxima	30
$I(\nu)$	Spectral Distribution of Intensity	31
I_0	Absorbed Intensity at Line Center	31
ν_a	Broadened Absorption Frequency	31
ν_0	Eigenfrequency	31
k	Radiation Propagation Vector	31
u	Velocity Vector	32
Ab	Absorbance	33
ϵ	Absorptivity	33

Symbol		Page
l	Absorbance Path Length	33
C	Concentration of Absorbing Species	33
$\delta\nu_{12}$	Doppler Frequency Shift	34
ν_{12}	Stationary Absorber Line Center	34
c	Speed of Light	34
α_{rel}	Relative Absorption	71
d	Plate Separation Distance	74
V_{probe}	Plate Voltage	74

List of Abbreviations

Abbreviation	Page
LAS	Laser Absorption Spectroscopy 1
LEO	Low Earth Orbit 2
GEO	Geostationary Orbit 2
DoD	Department of Defense 2
SPASS	Space Propulsion Analysis and Systems Simulator 3
PPT	Pulsed Plasma Thruster 15
SPT	Stationary Plasma Thruster 20
TAL	Thruster with Anode Layer 20
BaO	Barium Oxide 25
LaB ₆	Lanthanum Hexaboride 25
CEX	Charge Exchange Reactions 26
EM	Electromagnetic 28
FWHM	Full-width at Half Maximum 31
MPD	Magnetoplasmadynamic 32
RGA	Residual Gas Analyzer 39
AMU	Atomic Mass Unit 39
SCCM	Standard Cubic Centimeters per Minute 40
PPU	Power Processing Unit 44
TTL	Transistor-Transistor Logic 61
DAQ	Data Acquisition 64

SHEET VELOCITY MEASUREMENTS USING LASER ABSORPTION SPECTROSCOPY IN A XENON HALL EFFECT THRUSTER PLUME

I. Introduction

A new laser absorption spectroscopy (LAS) velocimetry system, designed to obtain 2D planar velocity and number density fields for ionized Xenon in the plume of a Hall effect thruster, was implemented for this research to obtain the first measurements on the system with a single horizontal laser sheet. This horizontal laser sheet was used to probe singly-ionized Xenon in the plume of a Busek BHT-600 Hall thruster. This research gathered initial data to validate the aforementioned sensor system, and developed MATLAB® code for obtaining velocity and relative number density measurements. The new LAS system is one part of a larger system combining it, and a microwave sensor for neutral Xenon, for full particle analysis in thruster plumes.

1.1 Background

LAS has been in use since the 1970s to analyze gases [1], and is one non-intrusive method of analyzing a gas flow. More recently, the method has been applied to Hall thruster plume analysis by researchers at the University of Tokyo [2, 3, 4, 5] with attempted application for point measurements at AFIT in 2010 [6].

Hall effect thrusters were a focus of research in the US in the 1960s [7, 8, 9], but interest waned in Hall thrusters between this period and the 1990s in the US, where Ion thrusters became more popular. Interest re-emerged in the 1990s in the

US, Europe, and Japan; before renewed Western interest in the Hall thruster, Russia was the primary source of Hall thruster research [10].

The Hall thruster is an ideal form of propulsion for low Earth orbit (LEO) and geostationary orbit (GEO) missions, where solar power is in ample supply and low thrust is tolerable. By using Hall thrusters for stationkeeping purposes or for orbit raising purposes instead of chemical thrusters, satellites require significantly less propellant mass, allowing for higher payload masses. Hall thrusters, then, increase the effectiveness of satellites by allowing for more mission hardware, or useful mass, in a given desired orbit.

All Department of Defense (DoD) missions operate in Earth orbit, so Hall thrusters are of a particular interest to the Air Force. Hall thrusters have been a major focus of electric propulsion research at the Air Force Research Laboratory since the early 2000s [11].

1.2 Motivation

Many methods exist to characterize an electric propulsion system such as the Hall thruster; however, these methods are often intrusive and involve analyzing the exhaust plume with probes. These probes may affect the flow of the exhaust plume and thus obtain inaccurate measurements. Intrusive measurements may also lead to probe erosion by ion bombardment, in the case of electrostatic propulsion systems such as the Hall thruster, which significantly limits their lifetimes. It is desirable to take non-intrusive measurements for these reasons. LAS provides a method to take these measurements in a non-intrusive manner.

Further, existing laser-based Hall thruster analysis systems generate point data. Simultaneous full-field measurements are desirable as they allow observation of time-resolved velocity data of ions at all points in the plume, albeit for this system at

relatively low sampling rates up to 1 kHz. This allows for analysis of plume divergence, plume symmetry, low-frequency oscillatory modes, charge-exchange reactions, and basic thruster performance characteristics from number density and velocity measurements.

Hall thrusters often have different performance characteristics on-orbit compared to those obtained from laboratory vacuum chamber characterizations. This is partially due to the recirculation and reacceleration of propellant gases in the chamber. More broadly, the vacuum chamber offers a denser environment than a Hall thruster would experience in LEO or GEO, and has a grounded, conductive wall in the near-field of the plume. These different ambient conditions result in the variation in performance for chamber tests. Implementation of this LAS system provides a tool to analyze the ions in the exhaust plume, and is the first part of a larger system which also includes a microwave-based system for neutral particle analysis. Combined, the two should allow for time-resolved full-field analysis of all particles in the exhaust plume, which should allow for better isolation of recirculated propellant effects. This in turn would allow more detailed analysis of the effects of chamber pressure on thruster performance in future research, and better predictions of on-orbit performance. This experimental data would also provide useful comparison for validation of numerical models of thruster performance.

1.3 Scope

The primary focus of this research was the successful integration of a novel LAS system into AFIT's Space Propulsion Analysis and Systems Simulator (SPASS) lab, one of AFIT's vacuum chambers. After integration, first measurements were obtained of velocity and relative number density in the plume of a Xenon Hall thruster. These measurements were intended to be used to develop methods for obtaining thruster

performance characteristics from the sensor data, which in turn would be compared against existing performance characteristics for the thruster. However, analysis of the velocity measurements showed that single-sheet analysis would not produce results accurate enough for determination of performance characteristics; code to do so was therefore not developed. This work validated the new system and provided procedures for its use in future work with Hall thrusters operating on Xenon propellant. It provided proof of concept verification for the first of two systems to be used for complete particle analysis of Hall thruster plumes, for future use developing better predictions of on-orbit thruster performance.

1.4 Objective

The objectives of this research included:

1. Build out and integrate a near-infrared Laser Absorption Spectroscopy system into AFIT's SPASS lab
2. Obtain the first full-field, simultaneous Xenon ion velocity and relative number density measurements in the plume of a Xenon Hall effect thruster
3. Develop methods for correlation of the resulting ion velocity and number density fields to basic performance characteristics including thrust and specific impulse
4. Validate results against intrusively-obtained characterization of the thruster

Completion of these objectives will provide AFIT with an operational Laser Absorption Spectroscopy system tuned for analysis of plumes of Xenon-based propulsion systems. Due to issues with delivery of software required for correlation of the six sets of linear view-angle absorption data to full 2D field data, and issues with the fiber pass-through causing only one of the seven pass-throughs to be operational, objective

2) was down-scoped to obtain data from a single horizontal sheet to validate the system as a whole. Further, as a result of the single-sheet measurements not providing accurate average exit velocities due to plume divergence, code for objective 3) was not developed.

II. Background

The Hall thruster is an electric propulsion system, used on satellites for station-keeping and orbit-raising purposes. The research described herein presents velocity and relative number density measurements for singly-ionized Xenon in the plume of a Hall thruster using Laser Absorption Spectroscopy (LAS), using one laser sheet from a system which will eventually provide 2D, full-field measurements. These measurements validate the LAS system against previously-obtained data for future use at AFIT’s Space Propulsion Analysis and System Simulator (SPASS) lab. This work builds upon the basics of rocket propulsion, plasma physics, and laser absorption spectroscopy. To this extent, Sections 2.1 and 2.2 contain an overview of rocket and plasma fundamentals, Sections 2.3 and 2.4 discuss electrostatic propulsion and Hall thrusters, and Section 2.5 provides an overview of laser fundamentals, Laser Absorption Spectroscopy, and Laser Induced Fluorescence as it pertains to Hall thruster plume analysis.

2.1 Rocket Fundamentals

Sutton and Biblarz define rocket propulsion as “a class of jet propulsion that produces thrust by ejecting matter, called the working fluid or propellant, stored entirely in the flying vehicle” [12]. Rocket propulsion systems may expel this propellant through a variety of methods, from heating and expanding a gas through a nozzle to using electric and/or magnetic fields to accelerate the propellant out of the system. Hall thrusters are one form of rocket propulsion, and operate under many of the same principles of chemical rocket propulsion systems, albeit at significantly reduced thrust and greatly improved specific impulse. A few performance characteristics of interest include thrust, specific impulse, and jet power. These characteristics of a

rocket propulsion system define the system's capabilities, from high-thrust launch vehicles to high-specific impulse thrusters designed for orbit maintenance. These are important characteristics for any rocket propulsion system, including Hall thrusters, and are used in the mission planning and satellite design process to choose the proper propulsion system.

Thrust is the force which a propulsion system applies to a vehicle, and is illustrated in Figure 1. Thrust has two primary components; first, the time rate of change of momentum of the propellant ejected from the thruster, and second, the pressure difference between the exit and ambient. Thrust (T) is defined as:

$$T = \dot{m}u_{ex} + (p_{ex} - p_0)A_{ex} \quad (1)$$

where $\dot{m}$ is the mass flow rate of the propellant exiting the propulsion system, u_{ex} is the exit velocity of the propellant, p_{ex} is the pressure at the exit, p_0 is the ambient pressure, and A_{ex} is the area of the nozzle at the exit.

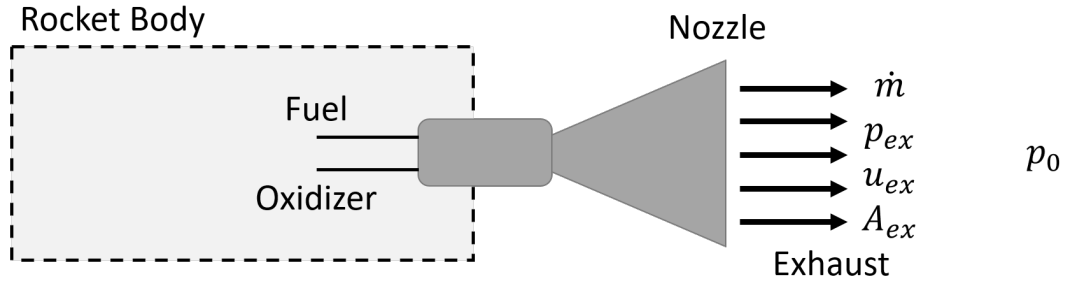


Figure 1. Production of thrust from a generic rocket propulsion system.

Assuming ideal expansion, where the exit pressure is equal to the ambient pressure, the second term will cancel and Equation 1 simplifies to:

$$T = \dot{m}u_{ex} \quad (2)$$

The thrust force can also be defined by:

$$T = m \frac{du}{dt} \quad (3)$$

where m is the instantaneous mass of the spacecraft and u is the spacecraft's velocity.

Here, the instantaneous mass of the spacecraft can be written as:

$$m = m_0 - \dot{m}t \quad (4)$$

where $\dot{m}$ is assumed constant, m_0 is the initial mass composed of the sum of the dry (structural) mass and propellant mass, and t is the elapsed burn time. Equation 3 can then be rewritten:

$$du = \frac{T}{m} dt \quad (5)$$

Replacing T with its definition in Equation 2, Equation 5 becomes:

$$du = \frac{\dot{m}u_e}{m} dt \quad (6)$$

By manipulating and integrating Equation 6, the ideal rocket equation is obtained [12]:

$$\Delta v = -u_{ex} \ln \frac{m_d}{m_d + m_p} \quad (7)$$

where Δv is the change in velocity for the vehicle resulting from the ejection of a certain mass of propellant, m_p , leaving only the dry mass, m_d , assuming that the entire mass of propellant is used or ejected.

A useful performance characteristic for a propulsion system is the specific impulse (I_{sp}), which provides a measure of efficiency. The specific impulse is defined as a ratio

of the thrust to the rate of propellant consumption and is given by:

$$I_{sp} = \frac{T}{\dot{m}_p g_0} = \frac{u_{ex}}{g_0} \quad (8)$$

where g_0 is the standard acceleration due to gravity, a constant defined as 9.80665 m/s^2 [13]. From this definition of I_{sp} , Equation 7 can be redefined:

$$\Delta v = -I_{sp} g_0 \ln \frac{m_d}{m_d + m_p} \quad (9)$$

From this definition of the ideal rocket equation, the effect of I_{sp} is clear; as it increases, for a given Δv , the mass of propellant required decreases. This is one major reason electric propulsion systems such as the Hall thruster are desirable; compared to chemical propulsion systems, their greatly increased I_{sp} indicates they require less propellant to produce the same Δv , allowing for more payload mass.

The exhaust has a certain kinetic energy output rate, termed the jet power (P_{jet}), which is defined as:

$$P_{jet} = \frac{1}{2} \dot{m}_p u_{ex}^2 \quad (10)$$

where the p subscript specifies that the mass flow rate as given is the propellant mass flow rate. The jet power can also be defined in terms of thrust:

$$P_{jet} = \frac{T^2}{2\dot{m}_p} \quad (11)$$

2.2 Plasma Fundamentals

The Hall thruster, like many electric propulsion systems, ionizes a propellant gas in order to accelerate the propellant using an electric field. The ionization process of impacting neutral propellant gases with high-velocity electrons forms a plasma inside

the Hall thruster. Thus, to understand Hall thruster operation, an understanding of plasma is required. Goebel and Katz [14] define plasma as “a collection of the various charged particles that are free to move in response to fields they generate or fields that are applied to the collection and, on the average, is almost electrically neutral.” Plasma can be described as a fluid, consisting of Maxwellian-distributed neutral particles and electrical charges. However, its actual consistency of ions and electrons typically have different energetic states. The ionization process does not heat ions extensively; they tend to possess low random velocities compared to electrons distributed in the plasma, where the electron temperature (T_e) to ion temperature (T_i) ratio $T_e/T_i \approx 10$.

The condition of electrically neutral indicates that the ion and electron densities must be approximately equal in a plasma; this is often called quasi-neutrality. Thus, in a plasma

$$n_i \approx n_e \quad (12)$$

where n_i is the ion number density and n_e is the electron number density.

Plasmas in a vacuum are governed by Maxwell’s equations, given in Equations 13 through 16. Maxwell’s equations are formed by Gauss’s law, Faraday’s law, Gauss’s law for magnetism, and the Ampere-Maxwell law, respectively [15].

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad (13)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (14)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (15)$$

$$\nabla \times \mathbf{B} = \mu_0 \left(J + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right) \quad (16)$$

In Maxwell's equations, E is the electric field, B is the magnetic field, ϵ_0 is the permittivity of free space, μ_0 is the permeability of free space, ρ represents the charge density in the plasma, including multiply charged ions, and J is the current density in the plasma, again including multiply charged ions. Charge density is found using

$$\rho = e(Zn_i - n_e) \quad (17)$$

where Z is the charge state and e is the elementary charge, defined as 1.602×10^{-19} Coulomb [13]. Similarly, the current density is found using

$$\mathbf{J} = e(Zn_i \mathbf{u}_i - n_e \mathbf{u}_e) \quad (18)$$

where u_i is the ion velocity and u_e is the electron velocity.

The Lorentz force equation is the equation of motion for a charged particle in a magnetic field, given as

$$\mathbf{F} = m_c \frac{d\mathbf{u}_c}{dt} = q(\mathbf{E} + \mathbf{u}_c \times \mathbf{B}) \quad (19)$$

where F is the force applied to the particle, m_c is the charged particle mass, q is its charge, and u_c is the particle velocity. When split into its components and manipulated, the Lorentz force equation describes a harmonic oscillator. This harmonic oscillator operates at a frequency defined as the cyclotron frequency, ω_c :

$$\omega_c = \frac{|q|B}{m_c} \quad (20)$$

The particle, moving through a magnetic field in some direction, takes on a helical path as a result of the Lorentz force. This helical path has a constant radius, called the Larmor radius, r_L , where the subscript $\perp$ indicates for $u_\perp$ and $V_\perp$ that the term

represents its component perpendicular to the direction of motion, tangential to the circle defined by the Larmor radius:

$$r_L = \frac{u_{\perp}}{\omega_c} = \frac{mu_{\perp}}{qB} = \frac{1}{B} \sqrt{\frac{2mV_{\perp}}{e}} \quad (21)$$

The Larmor radius is depicted in Figure 2 for positively and negatively charged particles. Considering an electric field applied to the same system, the particle will obtain an $E \times B$ drift velocity, u_{drift} , perpendicular to both the electric and magnetic fields:

$$\mathbf{u}_{drift} = \frac{\mathbf{E} \times \mathbf{B}}{B^2} \quad (22)$$

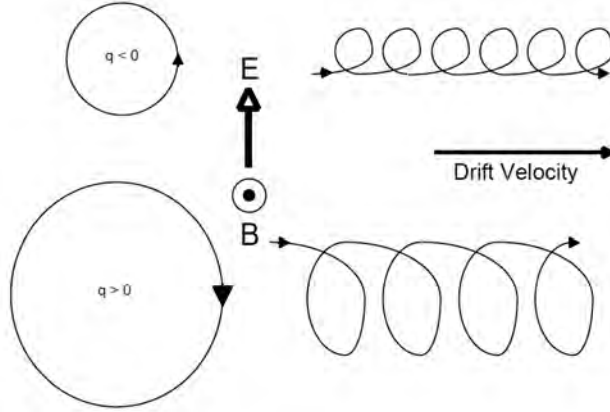


Figure 2. Larmor radius and particle drift for positively and negatively charged particles, adapted from [16].

The plasma can further be defined by the electron plasma frequency ω_p , given in Equation 23, and the ion plasma frequency Ω_p , given in Equation 24, which give measures of the minimum reaction time of the plasma to changes in its boundaries, and the time scale for ion movement within the plasma, respectively. By quasi-neutrality, the electron number density n_e is used in both equations; m_e is the electron mass and M_i is the ion mass. The values are significantly different primarily due to mass; ions are massive enough for inertia to play a role in their movement. The

ion acoustic velocity, u_a , describes the actual reaction to disturbances for ions in the plasma, and is given in Equation 25, where k is Boltzmann's constant.

$$\omega_p = \sqrt{\frac{n_e e^2}{\epsilon_0 m_e}} \quad (23)$$

$$\Omega_p = \sqrt{\frac{n_e e^2}{\epsilon_0 M_i}} \quad (24)$$

$$u_a = \sqrt{\frac{k T_e}{M_i}} \quad (25)$$

The plasma has a boundary region, illustrated in Figure 3, through which particles must enter and exit and where potential and density gradients occur. This region is the plasma sheath, which typically has a thickness of the order of the Debye length, λ_D :

$$\lambda_D = \sqrt{\frac{\epsilon_0 k T_e}{n_0 e^2}} \quad (26)$$

which provides a characteristic length defining a distance for plasma potential changes [14] and where n_0 is the equilibrium electron number density.

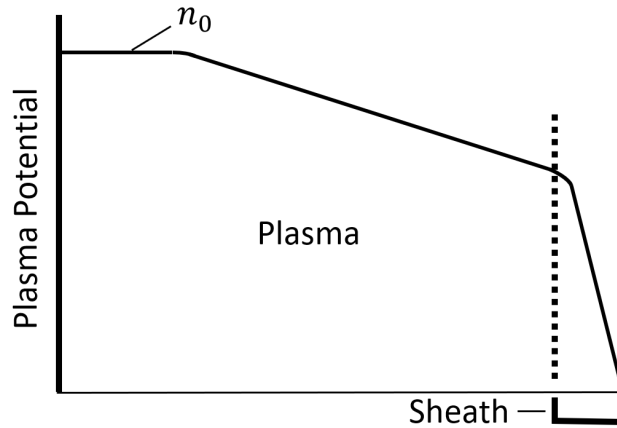


Figure 3. Plasma potential near the boundary.

Diffusion in the plasma is important in terms of particle transport within the plasma. Pressure gradients occur in the plasma, and collisions occur between different charged and neutral species in the plasma. Collisions cause diffusion from high to low plasma density regions, even across magnetic field lines. These may occur in the form of Coulomb collisions between charged particles or collisions with neutral particles. The mean free path (λ) is one metric of collisions, and can be defined for a plasma as:

$$\lambda = \frac{1}{n_a \sigma} \quad (27)$$

where σ is the cross-sectional area of one of the particles, assuming a simple sphere, and n_a is the neutral particle number density. The mean time between collisions, τ , is then:

$$\tau = \frac{1}{n_a \sigma u_c} \quad (28)$$

By taking the inverse of the mean time between collisions, the collision frequency ν is obtained:

$$\nu = \frac{1}{\tau} \quad (29)$$

Electron diffusion across magnetic field lines is of particular interest to Hall thrusters, as electrons diffuse across magnetic field lines from the external cathode to the anode. The magnetic field acts to reduce this diffusion and allow for the concentration of electrons to develop near the exit plane of the Hall thruster. This diffusion is described classically by the cross-field diffusion coefficient $D_{\perp}$:

$$D_{\perp} = \frac{D}{1 + \omega_c^2 \tau^2} = \frac{D}{1 + \Omega_e^2} \quad (30)$$

where Ω_e is the electron Hall parameter, given as:

$$\Omega_e = \frac{eB}{m\nu} \quad (31)$$

and D is the standard diffusion coefficient, given as:

$$D = \frac{kT_e}{m\nu} \quad (32)$$

2.3 Electric Propulsion

Chemical propulsion systems have a limit to their exhaust velocity, set by the energy release of breaking chemical bonds in the propellant [14]. Electric propulsion, or EP, separates the energy source from the propellant, removing this chemical limitation on energy and allowing greater exhaust velocities. Modern Hall thrusters, a form of electrostatic propulsion, can reach exhaust velocities of up to 20 *km/s* using Xenon gas [14].

There are three forms of electric propulsion: electrothermal, which uses electricity to heat a propellant gas and expel it in methods similar to chemical propulsion systems, electrostatic, which uses electric body forces to accelerate ionized propellant, and electromagnetic, which utilizes both electric and magnetic fields to accelerate propellant. Various propulsion systems are presented in Table 1, of which the Resistojet, Arcjet, Ion thruster, Hall thruster, and Pulsed Plasma Thruster (PPT) are forms of electric propulsion. Per the Ideal Rocket Equation (Equation 7), for a given Δv , as I_{sp} increases the required propellant mass decreases. High I_{sp} is achievable using Hall or ion thrusters.

Electric propulsion systems such as the Hall and ion thrusters require both a certain mass of propellant, but also a certain mass of power equipment, necessary to convert the relatively low voltages from the spacecraft bus supplied by the batteries

Table 1. Operating parameters for various thrusters, values from [14].

Thruster	Specific Impulse (s)	Input Power (kW)	Efficiency Range (%)	Propellant
Cold gas	50-75	–	–	Various
Chemical (monopropellant)	150-225	–	–	N_2H_4 H_2O_2
Chemical (bipropellant)	300-450	–	–	Various
Resistojet	300	0.5-1	65-90	N_2H_4 monoprop
Arcjet	500-600	0.9-2.2	25-45	N_2H_4 monoprop
Ion thruster	2500-3600	0.4-4.3	40-80	Xenon
Hall thruster	1500-2000	1.5-4.5	35-60	Xenon
PPTs	850-1200	<0.2	7-13	Teflon

and solar panels to the high voltages required for thruster operation. This mass of power equipment tends to scale with I_{sp} , so there exists for a given Δv an optimal I_{sp} . Given a time constraint, there also exists an optimal level of thrust. For smaller increments of Δv , propellant mass savings from an ion engine with high I_{sp} may be offset by increases in power system mass when compared to a Hall thruster with a lower I_{sp} and smaller power system mass. For many Earth-orbit applications such as station-keeping and LEO to GEO transfer, the Hall thruster offers the lowest total propulsion system mass and highest payload mass to orbit [12] at a cost of long firing intervals on the order of months. Thus, electrostatic propulsion systems such as the Hall thruster are of great interest.

2.3.1 Electrostatic propulsion.

Electrostatic propulsion systems operate by ionizing propellant and using electric fields to accelerate the ions out of the propulsion system. The exhaust consists of high-energy ions, low-energy ions generated from charge-exchange reactions, neutrals,

and electrons. Ion and neutral masses greatly exceed electron masses, with an approximately 10^3 difference in momentum, thus electron contributions to thrust can be neglected. High-energy ion exit velocities greatly exceed neutral and low-energy ion exit velocities, thus neutral and low-energy ion contributions to thrust can similarly be neglected. Therefore, thrust for electrostatic systems can be re-defined in terms of only high-energy ion contributions as:

$$T \approx \dot{m}_i u_i \quad (33)$$

where the i subscript indicates ion mass flow rate ($\dot{m}_i$) and velocity (u_i). The ion velocity, then, is given by:

$$u_i = \sqrt{\frac{2qV_b}{M_i}} \quad (34)$$

where V_b is the voltage across which the ion is accelerated, q is its charge, and M_i is its mass. This version of the thrust equation can be corrected for divergence in the beam and multiply charged species; simply multiply the thrust by a correction term, γ , where:

$$\gamma = \alpha F_t \quad (35)$$

$$\alpha = \frac{1 + 0.707 \left(\frac{I^{++}}{I^+} \right)}{1 + \frac{I^{++}}{I^+}} \quad (36)$$

$$F_t = \cos \theta \quad (37)$$

where $\frac{I^{++}}{I^+}$ is the ratio of doubly-ionized ions to singly-ionized ions, and θ is the average half-angle beam divergence. This correction neglects ionization beyond double-

ionization. Thrust, then, is given simply by:

$$T = \gamma \dot{m}_i u_i \quad (38)$$

For electrostatic propulsion systems, the mass utilization efficiency η_m reflects how well the thruster ionizes propellant, and is given by:

$$\eta_m = \frac{\dot{m}_i}{\dot{m}_p} \quad (39)$$

which assumes singly-ionized ions. Here, $\dot{m}_p$ is the total propellant mass flow rate. With this efficiency term, the specific impulse can be re-defined as:

$$I_{sp} = \frac{u_i}{g_0} \eta_m \quad (40)$$

The specific impulse can further be corrected to account for doubly-charged ions with a mass utilization correction term, α_m :

$$\alpha_m = \frac{1 + \frac{1}{2} \left(\frac{I^{++}}{I^+} \right)}{1 + \frac{I^{++}}{I^+}} \quad (41)$$

$$I_{sp} = \frac{u_i}{g_0} \eta_m \alpha_m \quad (42)$$

The thrust correction term accounting for divergence given in Equation 35 can alternatively also be applied in this case to provide another definition of specific impulse:

$$I_{sp} = \frac{u_i}{g_0} \eta_m \gamma \quad (43)$$

A number of other efficiency terms apply to electrostatic propulsion systems, including electrical efficiency, ion production efficiency, and total efficiency [14]. The

electrical efficiency is a measure of the beam power P_b compared to input power P_T , and is given in Equation 44. The ion production efficiency is actually a loss term, sometimes termed the discharge loss, and represents the power to produce ions, P_d , compared to the ion current produced, I_b . It is given in Equation 45; it is desirable to minimize the ion production efficiency. The total efficiency then provides a measure of the jet power, P_{jet} compared to the total electrical power consumed by the system, P_{in} , and is given by Equation 46.

$$\eta_e = \frac{P_b}{P_T} \quad (44)$$

$$\eta_d = \frac{P_d}{I_b} \quad (45)$$

$$\eta_T = \frac{P_{jet}}{P_{in}} = \frac{T^2}{2\dot{m}_p P_{in}} \quad (46)$$

2.4 Hall Thrusters

The Hall thruster makes use of both electric and magnetic fields in its operation, but is classified as an electrostatic propulsion system. This is because only the electric field is used to directly accelerate the propellant. A cross-section of a typical Hall thruster is presented in Figure 4. The Hall thruster typically consists of an annular acceleration channel, open at one end and sealed at the other. The anode resides at the sealed end, through which neutral propellant gas is fed into the acceleration channel. An external cathode provides electrons both for ionization and also for neutralization of the exhaust; electrons become trapped in the radial magnetic field and drift azimuthally in the $\mathbf{E} \times \mathbf{B}$ direction around the channel. Some electrons will diffuse toward the anode, with mobility reduced by the radial magnetic field.

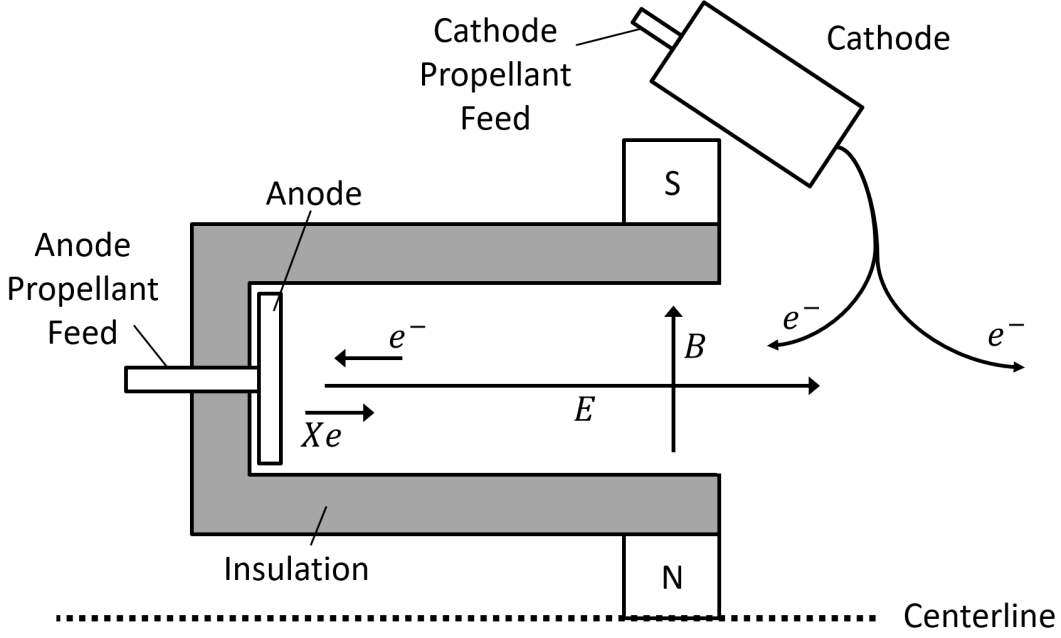


Figure 4. Cross-section of a typical Hall thruster.

Two main types of Hall thruster exist; one, characterized by a dielectric insulating wall, has several names including Hall thruster, stationary plasma thruster (SPT), and magnetic-layer thruster. The second, characterized by a metallic wall which typically is part of the magnetic circuit, is referred to as the thruster with anode layer (TAL). The first type of Hall thruster, with a dielectric insulating wall, is the type of thruster used in this research and thus is the type discussed henceforth.

The radial magnetic field tends to be strongest toward the exit plane of the thruster; this region captures the bulk of the electrons and acts to reduce their mobility toward the anode. Electrons drift azimuthally in the crossed E and B fields with a drift velocity u_E :

$$u_E = \frac{\mathbf{E} \times \mathbf{B}}{B^2} \approx \frac{E_z}{B_r} \quad (47)$$

where the subscript r indicates the radial component (B_r) and the subscript z indicates the axial component (E_z). The current in this region generated by the azimuthal

drift is the Hall current (I_H) for which the thruster is named:

$$I_H \approx n_e e w \frac{V_d}{B} \quad (48)$$

where n_e is the electron number density, w is the plasma width, and V_d is the discharge voltage. In order for this magnetic field to capture the electrons, their Larmor radii must be much less than the characteristic length, L . The electron velocity is characterized by the thermal velocity u_{th} , thus the electron Larmor radius r_e is given as:

$$r_e = \frac{u_{th}}{\omega_c} = \frac{m_e}{eB} \sqrt{\frac{8kT_e}{\pi m_e}} = \frac{1}{B} \sqrt{\frac{8}{\pi} \frac{m_e}{e} T_e} \ll L \quad (49)$$

where the characteristic length L is the magnetized plasma depth in the channel and T_e is the electron temperature in eV. A similar comparison exists for ions; for ion acceleration by the electric field, the ion Larmor radius r_i must be much greater than L :

$$r_i = \frac{u_i}{\omega_c} = \frac{M_i}{eB} \sqrt{\frac{2eV_b}{M_i}} = \frac{1}{B} \sqrt{\frac{2M_i}{e} V_b} \gg L \quad (50)$$

The strong concentration of electrons near the exit plane maintains the axial electric field for ion acceleration. An acceleration region and an ionization region exist in the annular channel; the regions overlap, with the ionization region upstream of the electric field peak, closer to the anode, and the acceleration region closer to the exit plane. Most electrons have trajectories as seen in Figure 5, however some diffusion will occur across magnetic field lines. Electrons diffusing across magnetic field lines toward the anode will still drift azimuthally and experience a weakening magnetic field, as the magnetic field is strongest at the exit plane of the thruster and goes to approximately zero at the anode. This increases the azimuthal drift velocity

around the channel as the electrons drift further away from the exit plane of the thruster. These high-energy electrons collide with neutral propellant atoms in the ionization region, generating ionized propellant.

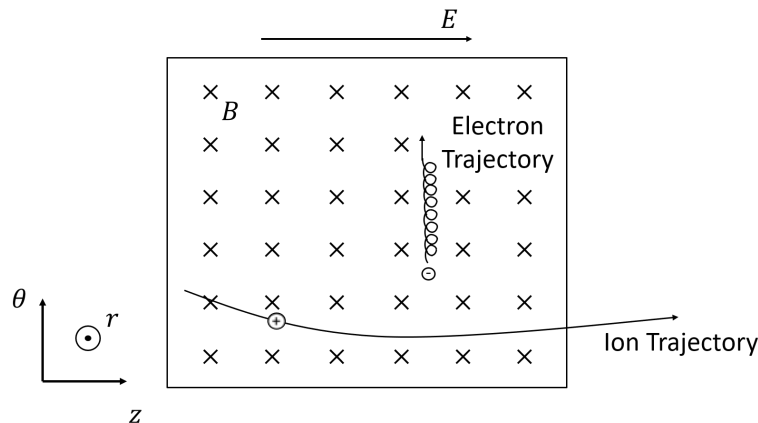


Figure 5. Particle paths in crossed E and B fields.

This diffusion across magnetic field lines, towards the anode, should occur at a rate given by the classical electron mobility as defined by Equation 30. However, measurements have shown this value to be orders of magnitude larger than classical diffusion, closer to the Bohm diffusion coefficient D_B [14] where

$$D_B = \frac{1}{16} \left(\frac{kT_e}{eB} \right) \quad (51)$$

This is often called *anomalous diffusion* and has been attributed to a single major plasma density fluctuation, a “spoke”, rotating azimuthally around the annular discharge region [9, 17, 10, 18] which has roots in the ionization process. Esipchuk et al. theorize that incomplete ionization of the propellant gas and immediate expulsion of the generated ions from the acceleration channel cause the spoke [19]. Oscillations or instability modes such as the spoke mode are often a result of the ionization and acceleration methods inherent to Hall thrusters [10]. If these oscillations are exacerbated, they can extinguish the discharge or affect plume divergence.

Neutral propellant atoms, injected from the anode, drift into the plasma and are ionized by collision with the electrons diffusing toward the anode or collision with electrons in the Hall current. The ionization mean free path (λ_i) describes how far into the plasma these neutrals drift before ionization occurs, and is given by:

$$\lambda_i = \frac{u_n}{n_e \langle \sigma_i u_e \rangle} \quad (52)$$

where u_n is the neutral velocity, σ_i is the ionization cross-section, and the bracketed term is the reaction rate coefficient. A high mass utilization efficiency is desired; any propellant gas left un-ionized is effectively wasted. To this end, the plasma thickness should exceed the ionization mean free path to ensure a high percentage of neutral propellant is ionized.

2.4.1 Hollow Cathodes.

Electrons are supplied to the Hall thruster by a cathode, often mounted externally, though experiments with internally-mounted cathodes have shown reduction in plume divergence and improvements in plume symmetry compared to typical external-mounted cathodes [20]. For this research, a hollow cathode was used. Part of the propellant gas flow is sent through the cathode for the generation of electrons; this gas flow exits the cathode outside of the acceleration channel and is thus not accelerated. A cathode efficiency, then, describes the amount of propellant injected through the anode compared to the total injected through the anode and cathode:

$$\eta_c = \frac{\dot{m}_a}{\dot{m}_p} = \frac{\dot{m}_a}{\dot{m}_a + \dot{m}_c} \quad (53)$$

where $\dot{m}_a$ is the propellant mass flow through the anode, available for ionization, and $\dot{m}_c$ is the propellant mass flow through the cathode, which is effectively lost.

The hollow cathode consists of four main components, which are visible in Figure 6. A hollow tube with an orifice plate at the exit is wrapped with a heater and contains a cylindrical insert at the exit end. This insert is the electron emitter, and emits electrons when raised to a certain temperature. The heater functions to bring the hollow cathode to this emissive temperature to begin electron discharge. It is typically turned off for operation of the cathode, as plasma bombardment from operation generates enough heat to keep the insert at temperature.

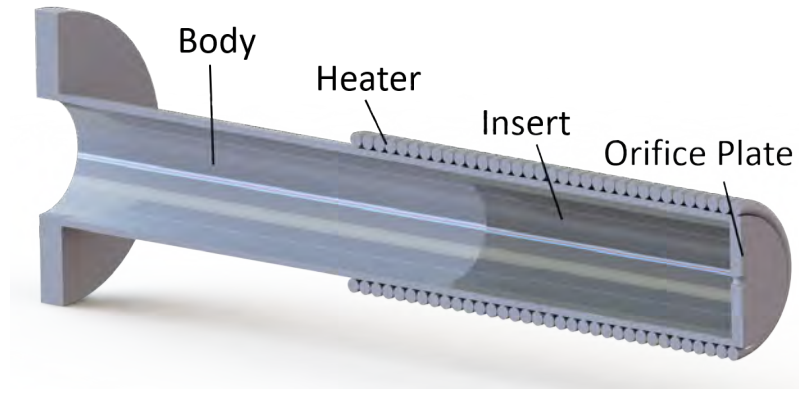


Figure 6. Hollow cathode geometry.

A fifth component, often included but not featured in Figure 6, is the cathode keeper. This is an electrode which typically encloses the cathode and has a positive bias. The keeper helps to start the cathode discharge, maintain temperature and operation in case of discharge interruption, and provide protection from sputtering, or erosive ion impact. Keeper erosion from ion impacts can be seen in Figure 7.

A discharge current (I_c) flows through the orifice, given by:

$$I_c = n_e e u_{e,d} A_o \quad (54)$$

where n_e is the plasma density, $u_{e,d}$ is the electron drift velocity, which is typically much lower than the thermal velocity, and A_o is the cross-sectional area of the orifice. The orifice cross-sectional area is determined by the amount of current necessary;

larger currents require larger cross-sectional areas. As seen in Figure 7, if the cathode orifice is not properly sized, significant erosion can occur in this region.



Figure 7. Hollow cathode erosion, with visible degradation of the orifice and cathode keeper.

The electron source for this current is the cathode insert, a thermionic electron emitter. Two main types of insert are used: barium oxide (BaO) dispensers or lanthanum hexaboride (LaB_6) [20]. Lanthanum hexaboride cathodes are more durable and at lower risk of poisoning compared to barium oxide dispenser cathodes, but barium oxide dispenser cathodes are more widespread as an electron source for Hall thrusters. The cathode used in this research is a BaO dispenser cathode. Such oxide cathodes can produce large emissions at low temperatures, but evaporate and are sensitive to damage from ion bombardment limiting life to tens of hours [14]. The dispenser cathode solves these issues, containing a reservoir of the oxide material and supplying the surface layer from this reservoir. The insert is often made of porous tungsten, which acts as a reservoir [20]. Barium is evaporated from the dispenser cathode insert, and ionizes easily, supplying the electrons required. The barium ions drift upstream due to gradients in the cathode, and deposit in cooler sections of the hollow cathode tube; these ions never leave the cathode.

2.4.2 Lifetimes.

The lifetime of the cathode is then primarily limited by the quantity of emissive material contained in the insert reservoir. Exposure to air or water, or impurities in the gas feed can also reduce cathode life. For BaO dispenser cathodes, complete oxygen poisoning can occur at oxygen partial pressures on the order of 10^{-7} torr, and complete water vapor poisoning can occur at partial pressures on the order of 10^{-6} torr. For this reason, thrusters using BaO dispenser cathodes for neutralization require propellant-grade purity of 99.9995% for propellant gases [14].

The lifetime of Hall thrusters is primarily determined by erosion of the channel and of the cathode. Thruster lifetimes of less than 10,000 hours are typical [14]. Charge exchange reactions (CEX) are one of the primary factors in the bombardment and erosion of thruster components and surfaces. CEX occurs when an un-ionized, neutral propellant particle in the exhaust plume collides with an energetic ion accelerated out of the thruster; charge exchange occurs, resulting in a low-energy or “cold” ion and an energetic neutral. The “cold” ion is then accelerated back towards the thruster and may impact components of the Hall thruster or the spacecraft, causing erosion. Most CEX occurs in the exhaust plume near the exit plane of the thruster [14, 5].

2.4.3 Plumes.

The plume of a Hall thruster contains primarily high-energy ions, accelerated and ejected from the Hall thruster. It also contains un-ionized propellant gas, low-energy ions and electrons, and sputtered thruster material. The low-energy ions present in the plume are the result of CEX, discussed earlier. In vacuum tests, some neutralized exhaust may re-circulate in the chamber and also be present in the plume.

The plume can have impacts on other components of a satellite on orbit, from erosion of components due to ion impact (called sputtering) to contamination of

components, such as deposits forming on imaging lenses. The exhaust plume exits the thruster with some divergence angle; satellite components within this cone will be impacted by the plume, mostly subject to ion impact or bombardment. These impacts provide momentum transfer from the ions to the spacecraft, given by:

$$p_i = n_i M_i u_i \quad (55)$$

where p_i is the ion momentum, n_i is the ion number density, u_i is the ion velocity, and M_i is the ion mass. Neutral propellant and other components of the plume have significantly lower velocity compared to the ions, so the plume momentum is effectively the ion momentum [14]. Further assuming that the ion velocity after impact is much less than the velocity before impact, the momentum transfer can be considered a complete transfer to the spacecraft. This momentum transfer per unit area can then be written as:

$$F_m = (2 - A_c)p_i \quad (56)$$

where F_m is the momentum transfer per unit area and A_c is the surface accommodation coefficient, which has a value close to unity [14]. It is important to account for potential impacts of the plume on a satellite's solar panels in the design process. In addition to effects of direct impacts by the plume on satellite components, the plume also causes phase shifts in microwave signals passing through, and causes weak optical emissions which could affect sensitive imaging instruments [14].

2.5 Laser Diagnostics

Laser diagnostic techniques are ideal for surveying gas flows due to the non-intrusive nature of the methods. Intrusive probes have been shown to induce in-

stability modes in Hall thrusters [21]. Unlike intrusive probes, non-intrusive methods do not disturb the flowfield and thus do not affect the measurements being taken. This is especially useful in Hall thruster exhaust plumes close to the exit plane of the thruster, where intrusive probes both affect the plume, and are quickly eroded by impact of the energetic ions in the exhaust. Limitations exist for these methods, including sampling frequency; laser diagnostic methods do not sample at high enough frequencies to provide analysis for Hall thruster instability modes which are often oscillatory at frequencies on the order of kHz or MHz [10]. To understand these methods, laser fundamentals are first discussed, followed by two popular laser diagnostic techniques: Laser Absorption Spectroscopy and Laser Induced Fluorescence.

2.5.1 Laser Fundamentals.

“LASER” is an acronym, standing for Light Amplification by Stimulated Emission of Radiation [22]. The term today, however, has taken on the meaning of a system or device that produces this light. The laser light is directed and of a high intensity, with emitted photons concentrated around a particular wavelength. These systems range in powers from pocket-sized devices used in presentations to military weapons.

There are three main components to a laser; these are the active medium, energy pump, and optical resonator [23, 22]. The active medium is the material which acts to amplify the electromagnetic (EM) wave. The energy pump provides energy to the active medium to increase the population density N of particles in the active medium at a higher energy state E_k . With enough pump power, $N(E_k)$ may exceed $N(E_i)$, the population density of particles at a lower energy state, and a population inversion occurs. Thus, the induced emission rate exceeds the absorption rate of the particles forming the active medium, and the active medium acts as an amplifier for EM waves passing through it. The optical resonator may be thought of as two opposite mirrors

[23], and provides feedback of emissions from the excited active medium. The optical resonator is typically chosen to limit this feedback to certain frequencies, and is used in tunable lasers to choose the laser's wavelength.

The absorption coefficient ($\alpha(\nu)$) of the active medium can be defined as:

$$\alpha(\nu) = \left[N_i - \frac{g_i}{g_k} N_k \right] \sigma_{ik}(\nu) \quad (57)$$

where N_i and N_k are the population densities of energy states i and k , g_i and g_k are the statistical weights, σ_{ik} is the absorption cross section, and ν is the frequency. From Equation 57, it is clear that when $N_k > \frac{g_i}{g_k} N_i$, $\alpha(\nu)$ becomes negative. This indicates that when the population density of the higher energy state exceeds the lower (when population inversion occurs), the active medium acts as an amplifier.

If two mirrors separated by a distance L_m have a reflectivity R then there is a loss in the reflected beam of $(1 - R)$. Other loss terms may be present, including absorption, diffraction, and scattering. These loss terms are accounted for by the loss coefficient, γ_r . The loss can therefore be described in terms of intensity, I :

$$I = I_i e^{-\gamma_r} \quad (58)$$

where I_i is the initial intensity. If an active medium is between these two mirrors, we can also account for amplification for one round-trip distance of $2L_m$:

$$I = I_i e^{-2\alpha(\nu)L_m - \gamma_r} \quad (59)$$

For amplification, then, the condition $-2L_m\alpha(\nu) > \gamma_r$ must be satisfied. This leads to the threshold condition [23]:

$$\Delta N = N_k \frac{g_i}{g_k} - N_i > \Delta N_{thr} = \frac{\gamma_r}{2\sigma(\nu)L_m} \quad (60)$$

where ΔN is the population difference and ΔN_{thr} is the population difference threshold for amplification. For the reflected wave's intensity to increase, the inverted population difference must exceed the threshold condition.

This system of reflectors and active medium can be termed the resonator cavity. For a laser resonator, the cavity must provide feedback for the desired modes and losses for the others. Depending on the active medium, a number of modes will naturally resonate, with gain exceeding loss. The active medium, therefore, is the determining factor in the range of possible modes attainable by a particular laser. To select a single mode, or wavelength, all other modes must be suppressed. Selective suppression can be attained through the use of a Littrow prism, or Littrow grating for modes in the infrared [23].

Lasers, like any radiation source, have an emission spectrum. Demtröder defines the emission spectrum as “the spectral distribution of the radiant flux from a source” [23]. Discrete emission spectra may also exist for some sources, where this flux reaches maxima at particular frequencies. Emissions are caused by transitions between higher and lower energy states:

$$h\nu_{ik} = E_k - E_i \tag{61}$$

where h is a photon, ν_{ik} is the frequency of the maxima, where the transition occurs, E_k is the high energy state, and E_i is the low energy state. The discrete emission spectra are also known as line spectra, and the frequencies at which they occur are known as eigenfrequencies. Particles at the low energy state may absorb energy at these same frequencies. For lasers, selective suppression is used to limit output to a single wavelength or frequency.

The resulting single-mode laser will not emit perfectly at a single wavelength; there will still be slight frequency fluctuations resulting in a linewidth. Demtröder describes

three sources of noise causing this linewidth: first, spontaneous emission from the excited active medium; second, amplitude fluctuations resulting from a distribution in the number of photons in the oscillating mode; third, phase fluctuations, as photon avalanches have a random distribution in phase [23].

Absorption, too, will not occur at simply one exact frequency or wavelength; rather, an absorption line or line profile for absorption describes the spectral distribution $I(\nu)$ of the absorbed intensity around the eigenfrequency, or line center. An example line profile is presented in Figure 8. The full-width at half maximum (FWHM) is also known as the linewidth, or halfwidth of the spectral line. It is bound by two frequencies around the line center at which the absorbed intensity is half that of the absorbed intensity at the line center, I_0 . The region inside the FWHM is the kernel of the line, and the regions outside the FWHM are the line wings [23].

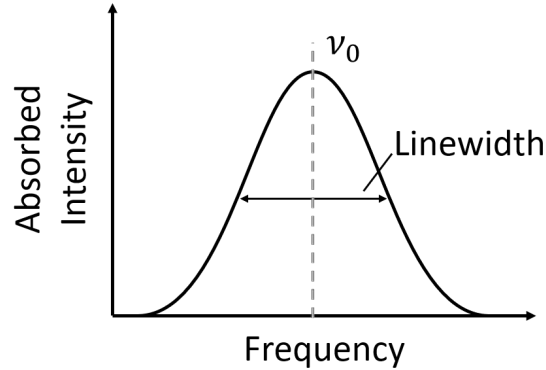


Figure 8. Illustration of an absorption line profile.

For gases at low pressures, such as those in a Hall thruster operating at vacuum, Doppler width is one of the major sources of spectral linewidth. This is caused by the thermal motion of absorbing particles. Absorption will actually occur at ν_a :

$$\nu_a = \nu_0 + \mathbf{k} \cdot \mathbf{u} \quad (62)$$

where ν_0 is the eigenfrequency or line center, $\mathbf{k}$ is the vector describing the imping-

ing light or radiation propagation, and $\mathbf{u}$ is the particle’s velocity vector [23]. The Doppler-broadened spectral lines can be represented as a combination of a Gaussian and Lorentzian profile, called a Voigt profile.

Many types of lasers exist, and this research makes use of a diode laser. The diode laser operates by passing a suitable current through a p-n semiconductor diode, which generates a significant emission of EM radiation, which in turn causes a population inversion and generation of the laser beam. This radiation is caused by spontaneous emission from electrons recombining with holes in the p-n junction. Choosing the proper semiconductor material allows variation and selection of the wavelength of this spontaneous emission [23].

2.5.2 Laser Absorption Spectroscopy.

Laser absorption spectroscopy (LAS) is a non-intrusive method of studying a fluid. LAS has numerous uses, traditionally used for the detection of trace gases and taking atmospheric measurements [24]. More modern applications have been applied at the University of Tokyo and include enthalpy measurements in plasma generator flows by Matsui *et al.* [2] in 2005, temperature and number density measurements in Xenon Hall thruster plumes by Yokota *et al.* [3], velocity measurements in the plume of a magnetoplasmadynamic (MPD) arcjet by Kinefuchi *et al.* [4] in 2006, and evaluations of background chamber Xenon on number density measurements in Xenon Hall thruster plumes by Matsui *et al.* [5] in 2007. A typical LAS setup is depicted in Figure 9.

LAS is useful in Hall thrusters to survey the region close to the exit plane of the thruster, where intrusive probes would both disturb the flowfield and be subject to erosion from ion bombardment. The basic principle of LAS uses a tunable laser, tuned to a certain wavelength matching an absorption line of a target molecule, and

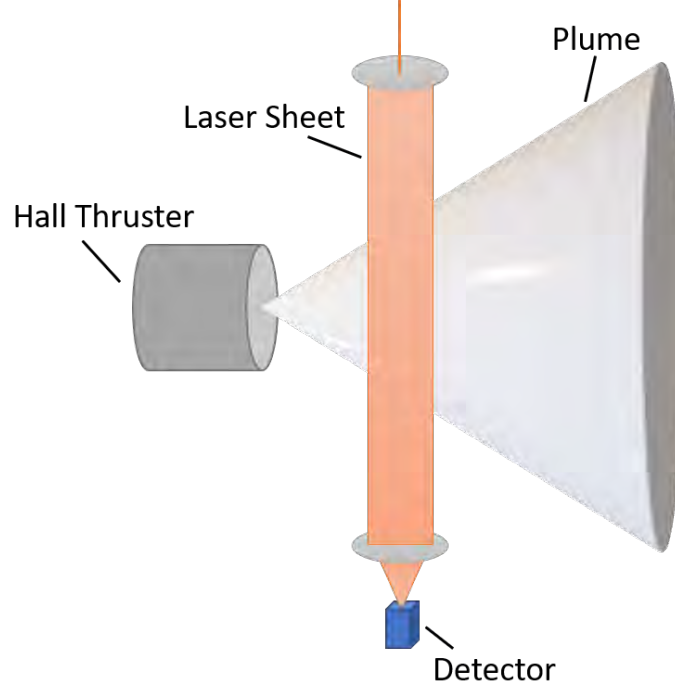


Figure 9. Illustration of a typical LAS setup.

a detector to determine absorption spectra [24]. The laser is aligned to pass through the region of interest and into the detector. Absorption can be defined by Beer's law [25]:

$$Ab = \epsilon l C \quad (63)$$

where Ab is the absorbance, ϵ is the absorptivity, l is the path length, and C is the concentration of the absorbing species.

If the species absorbing the laser light has a velocity component along the same axis as the laser, it absorbs light at a frequency different from that of stationary absorbers [26, 4]. The species velocity can therefore be determined by this Doppler shift, using:

$$\delta\nu_{12} = \nu_{12} \frac{u}{c} \quad (64)$$

where $\delta\nu_{12}$ is the Doppler frequency shift, ν_{12} is the stationary absorption line center, u is the species velocity, and c is the speed of light. To the author’s knowledge, LAS has never been used to obtain field velocity measurements. Thus, this system’s design to obtain three-component velocity field measurements of high resolution in Hall thruster plumes using LAS represents a novel concept.

2.5.3 Laser Induced Fluorescence.

Similar to LAS, Laser Induced Fluorescence (LIF) is also a non-intrusive method of studying a fluid, and is also useful in Hall thrusters to survey the region close to the exit plane of the thruster. However, instead of measuring absorbance, LIF excites certain transitions of the target species which fluoresce and emit photons [23]. The target species may fluoresce at the same frequency as the absorption, or at a different frequency, and some absorption lines may not fluoresce. LAS directly measures absorption, while LIF measures a secondary effect of the absorption; the fluorescence profile is determined by the detector instead of absorption spectra. The species velocity, in the same manner as LAS, can be determined using Equation 64. Here, instead of observing a Doppler shift from the stationary absorber line center, the Doppler shift is observed from the fluorescence profile line center for particles which have a velocity component along the same axis as the laser.

A number of papers have obtained point measurements in the plume of Hall thrusters using LIF methods, including Cedolin et al. (1997), Hargus and Cappelli (2001), Hargus and Charles (2008), and Lee (2010) [26, 27, 28, 6]. Further, LIF methods have been applied to subsonic and supersonic gas flows to obtain velocity-field measurements of low resolution (100×100) by Hiller *et al.* [29, 30, 31] using laser sheets and photodiode-array cameras such that each pixel measures fluorescence from a defined volume.

2.6 Prior Experiments at AFIT

Two prior AFIT theses relate to this work; Lt Duc Bui’s 2012 characterization of the Busek 600W Hall thruster, which provides nominal thruster performance data for comparison, and Capt Daniel Lee’s 2010 work profiling the plume of a Busek 200W Hall thruster using laser induced fluorescence.

2.6.1 Busek 600W Hall thruster characterization.

Lt Bui performed a full characterization of the Busek 600W Hall thruster in 2012 [32], using intrusive probes including a Faraday probe for determinations of ion current density and an $E \times B$ probe (Wein filter) for determining ionized Xenon charge distributions, as well as an inverted pendulum thrust stand. Results from the experiments described herein use the same thruster and BHC-1500 cathode, and compare data to Bui’s 2012 characterization. The thruster and cathode are described in detail in Chapter III. Bui’s experiments made use of the same facility as this research, also described in Chapter III, with the exception of the 2017 extension installed to increase the vacuum chamber’s volume.

Table 2 presents the nominal-condition advertised performance values compared to the experimentally determined values. Actual performance is clearly moderately worse than advertised. At these nominal conditions, Bui found ionized Xenon species fractions of 0.903 Xe^+ , 0.051 Xe^{2+} , and 0.041 Xe^{3+} . Bui also determined that the inner and outer magnet coils could use different currents, but any differences in varying current are minor, and both can be connected in series with no adverse effects. Further, discharge voltages on the lower end around 175V and voltages on the higher end of 350-400V should be avoided due to instabilities observed at these voltages.

Table 2. Nominal performance parameters for the Busek 600W Hall thruster [32].

	Busek Co.	Bui Experimental
Discharge Voltage	300V	300V
Discharge Current	2.0A	2.0A
Anode $\dot{m}$	2.5 mg/s	2.66 mg/s
Cathode $\dot{m}$	0.15 mg/s	0.146 mg/s
Magnet Current	1.75A	1.75A
Thrust	42 mN	34.8 ± 1.3 mN
I_{sp}	1585 s	$1,267 \pm 39$ s
Efficiency	55.4%	$36.2 \pm 2.4\%$

2.6.2 LIF analysis of Busek 200W Hall thruster plume.

Capt Lee’s work provided analysis of a different thruster than is used in this research, the Busek BHT-200. His research obtained point data from laser diagnostics, with an unsuccessful attempt at implementing LAS in addition to LIF due to a lack of equipment to perform both measurements simultaneously. He identified and attempted to use the absorption line of ionized Xenon at 834.72 nm, which this reasearch utilizes to obtain relative number density and velocimetry measurements. His experience with laser diagnostics in the same facility used in this research is relevant, particularly his issues with an overheating laser. Lee noted that the laser often overheated on warm days, shutting off, and that consistent operation was only obtained during the winter with 63-68°F ambient temperatures [6]. Ambient temperature is thus a point of concern during experimentation; however, improved cooling in the lab has been installed since Lee’s work, which should mitigate the issue.

III. Methodology

This research analyzed the plume of a Xenon Hall thruster using a LAS system designed to provide full-field, simultaneous Xenon ion number density and velocity profiles of a Hall thruster plume. To obtain first measurements and validate the full system, a single horizontal laser sheet was used to survey the plume. Measurements were performed at AFIT's SPASS lab vacuum chamber. Full-field measurements will eventually be taken in future research with the full system, which will allow time-resolved analysis of plume divergence, plume symmetry, low-frequency oscillatory modes, charge-exchange reactions and ion backflow, and basic thruster performance characteristics.

This analysis was performed to validate the LAS system against previously-obtained data from Bui's 2012 thesis [32], for future use at the SPASS lab as the first part of a larger system for full particle analysis of Hall thruster plumes. To this extent, Chapter III contains discussion of the laboratory setup and instrumentation. Section 3.1 discusses the vacuum chamber and its supporting equipment, Section 3.2 details specifics for the Xenon Hall thruster and its equipment, and Section 3.3 discusses the LAS system and its implementation. Section 3.4 details the test conditions used in firing the Hall thruster and collecting LAS data, and section 3.5 discusses the method used to quantify uncertainty in the collected data.

3.1 Laboratory

Experiments were performed at AFIT's SPASS lab, pictured in Figure 10. The vacuum chamber is 1.8 meters in diameter and 3.75 meters in length, formed from three sections. It can maintain pressures on the order of 10^{-5} torr during thruster operation and 10^{-7} torr without thruster operation. Section 3.1.1 contains details

for the pumping system and pressure gauges, Section 3.1.2 discusses the propellant supply system for the chamber, and Section 3.1.3 provides details for the internal traverse systems.



Figure 10. The SPASS Lab vacuum chamber.

3.1.1 Pumps and Pressure Gauges.

Pressures were maintained using a Oerlikon Leybold SP250 roughing pump to bring pressures in the chamber to approximately 70 millitorr. Four CVI Torrmaster TM500 cryopumps, one CVI Torrmaster TM250 cryopump, and one CVI Torrmaster CRG012 chilled plate were used to obtain operational pressures with a pumping speed of 20,900 l/s. The cryopumps and chilled plate attach to the top of the chamber as seen in Figure 10. The chamber extension completed in 2017 added two flanges for mounting two additional TM500 cryopumps, but these have yet to be populated.

Pressures were measured using a Kurt J. Lesker 300 Series pressure gauge for higher pressures, and an Extorr Residual Gas Analyzer (RGA) for low pressure analysis. The RGA allowed identification of gas species present within the chamber based on atomic mass unit (AMU) when combined with Extorr's VacuumPlus software, pictured in Figure 11, and can measure pressures down to 10^{-11} torr. This was used to monitor the chamber pressure during thruster operation, and verify that propellant gases were flowing; Xenon, with an AMU of 131, is visible due to an aliasing effect as the spikes between 61 and 71 AMU.

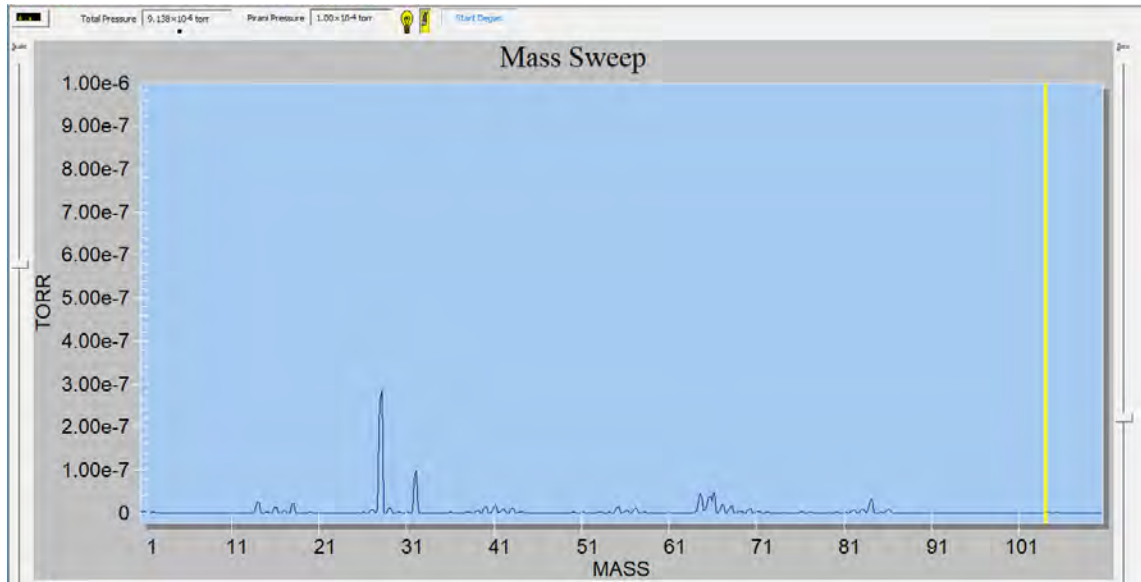


Figure 11. Extorr's VacuumPlus software, used with the Residual Gas Analyzer to monitor partial pressures inside the chamber.

The Lesker 300 Series pressure gauge can measure pressures down to 1×10^{-4} Torr and has a resolution of 0.1 mTorr from pressures of 1×10^{-4} to 1×10^{-3} Torr, an accuracy of $\pm 10\%$ from 1×10^{-3} to 400 Torr, and an accuracy of $\pm 2.5\%$ from 400 to 1,000 Torr [33]. The Extorr RGA contains a Pirani gauge for pressure measurements from 1×10^{-3} Torr to ambient as well as an ion gauge which operates below 1×10^{-2} Torr and a quadrupole which operates below 1×10^{-4} Torr for partial pressure determination. The Extorr RGA can determine partial pressure AMU determination

with a resolution of better than 0.5 AMU at 10% peak height [34]. These pressure gauges automatically control the crossover from the roughing pump to the cryopumps when the chamber is pumping down to vacuum.

3.1.2 Propellant Supply.

Propellant was fed into the chamber using a system of mass flow controllers, pictured in Figure 12; Krypton and Xenon tanks were attached to the system, though this research only made use of Xenon. The mass flow controllers used were MKS Model 180A controllers, pictured in Figure 12a and 12b; two of the four were used with one supplying the cathode and the other supplying the anode. The cathode mass flow controller was limited to a range from 0-10 standard cubic centimeters per minute (SCCM) while the mass flow controller used for the anode was limited to a range from 0-50 SCCM. The mass flow controllers are accurate to $\leq 0.01\%$ [6].

These mass flow controllers were monitored and set using a MKS Type 247 four-channel readout, allowing adjustment by percent of maximum flow rate. These propellant lines entered the chamber through a single flange, pictured in Figure 12c. Two of the four lines were split in two, allowing for the installation of two thrusters in the vacuum chamber simultaneously. This enabled experimentation on multiple thrusters without requiring venting, changing thrusters out, and pumping the chamber back to vacuum. The split lines were connected to mass flow controllers two and four; controller two was capable of up to 10 SCCM and controlled flow to the cathode, and controller four was capable of up to 50 SCCM and controlled flow to the anode. The MKS Type 247 readout had a broken potentiometer on channel two making adjustments impossible, so channels two and three were swapped on the readout. Mass flow was then controlled using channels three and four, where channel three corresponded to cathode mass flow and channel four corresponded to anode mass flow.

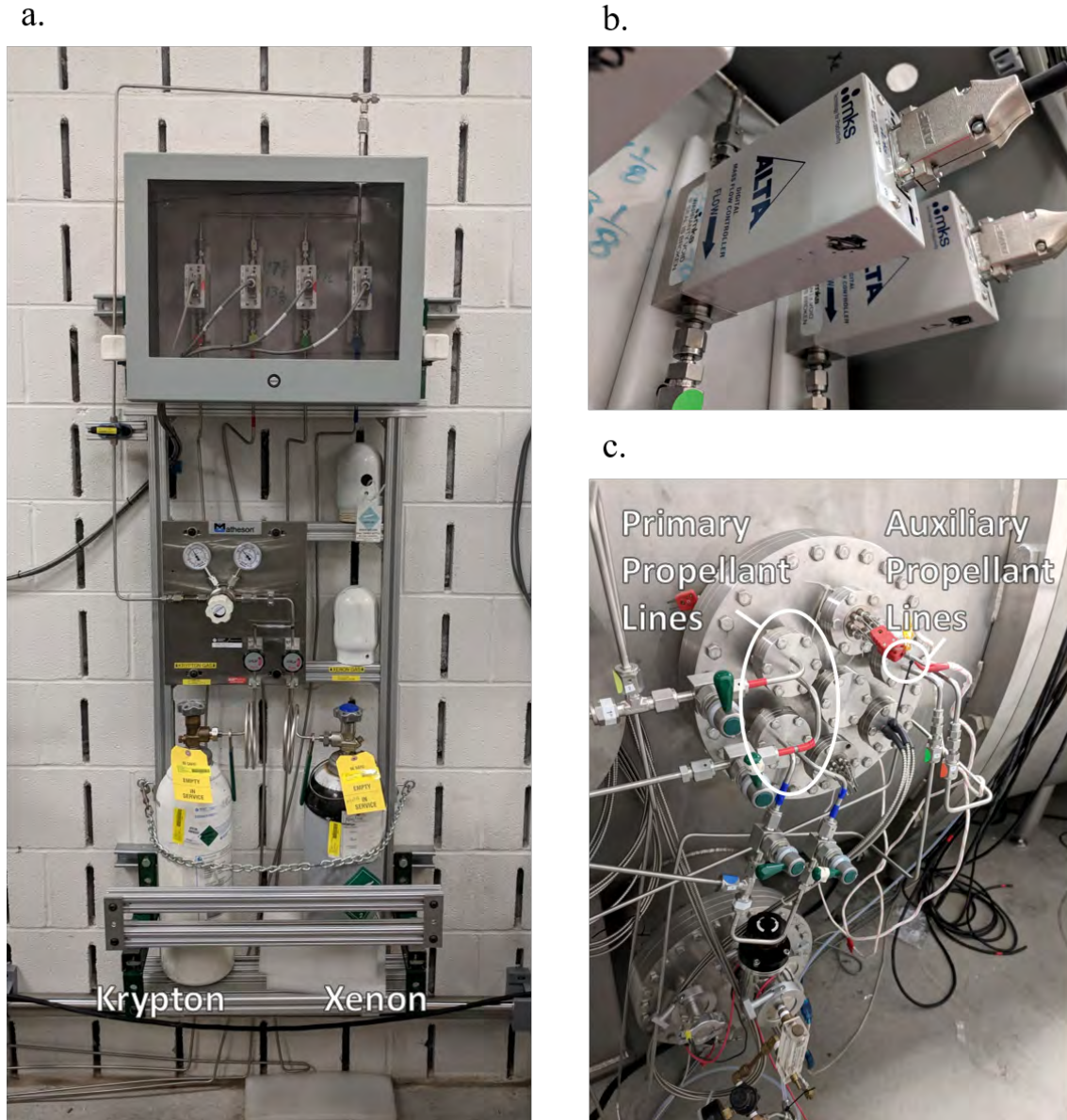


Figure 12. Propellant supply system. a. Propellant tanks and mass flow controllers, b. MKS 180A mass flow controllers, c. Attachment point for propellant lines to vacuum chamber. Valves on the primary propellant lines allowed use of the same controllers when two thrusters were simultaneously installed inside the chamber.

While Xenon was used in this research, Krypton was also used by other research ongoing in the SPASS Lab vacuum chamber. Care was taken to avoid back-flow of Krypton or Xenon into the opposite tanks; as both were connected to the same lines, whenever propellants were swapped, the lines were evacuated back to the tanks before opening the other propellant's valve.

3.1.3 Traverses.

The chamber includes two traverses, the larger of the two pictured in Figure 13. They were used to maneuver the thruster to adjust which portion of the plume was imaged, and for maneuvering intrusive probes through the plume if desired. The front traverse provided 20 cm of thruster maneuvering; its translational capabilities exceed this but only 20 cm of movement was required to image different sections of the plume. The rear traverse, for maneuvering probes, was not utilized in this research.

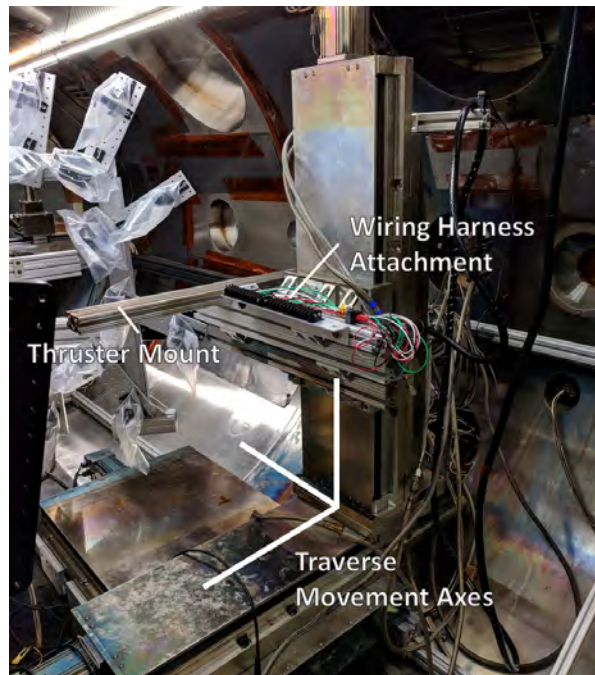


Figure 13. Front traverse, used for mounting the thruster and adjusting which portion of the plume is imaged.

3.2 Thruster

The thruster used in this research was the Busek BHT-600 Hall thruster. The overall configuration for the thruster with its power and propellant line connections can be seen in Figure 14. While one “power” line is shown running from the power processing unit, there were actually multiple lines for the heater, magnets, cathode, and anode. The locations relatively represent the component locations within the laboratory, but sizes and exact locations are not to scale.

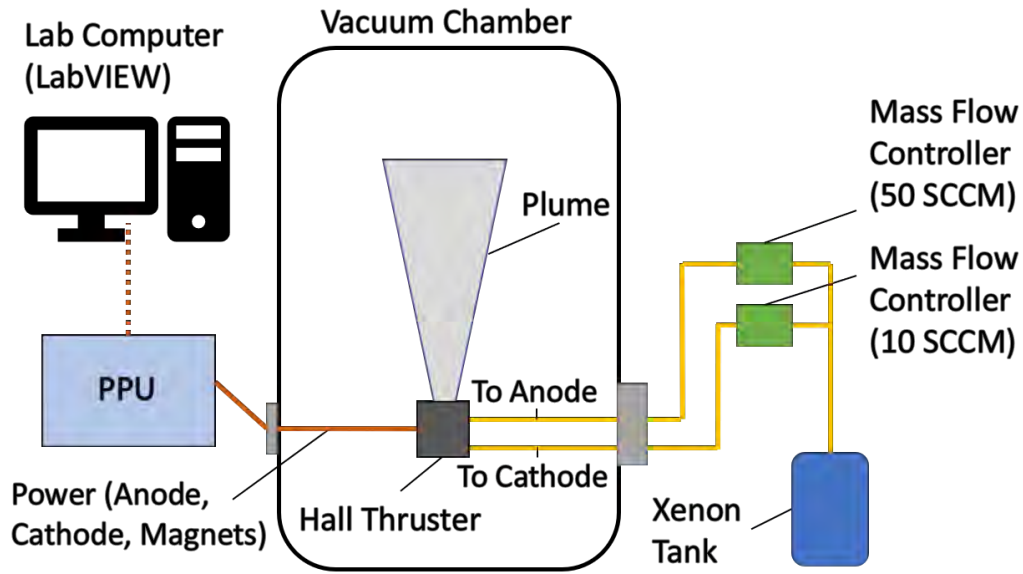


Figure 14. Thruster connection diagram. Component sizes and locations not to scale.

3.2.1 Busek BHT-600 Hall Thruster.

The Busek BHT-600 Hall thruster, pictured in Figure 15, can operate at a range of powers from 400 to 800 W and nominal design power of 600 W. The thruster had external dimensions of 11 x 11 x 8.6 cm and a mass of 2.5 kg [35]. The thruster was a standard Hall thruster, also known as a stationary plasma thruster or magnetic-layer thruster, and featured a dielectric insulating channel made of Boron Nitride. Con-

nections can be seen in Figure 15b on the rear of the thruster; it required connections to a propellant feed, and two electrical connections to the anode power supply and the magnet power supplies. The anode power supply power input was isolated to a separate, insulated cable in order to prevent high-voltage arcing. The magnets can be connected in series with no adverse effects, as advertised by Busek Co. [35] and verified by Duc Bui [32], and thus were connected in series. The magnetic field generated in the acceleration channel of the Busek BHT-600 is ≈ 0 near the anode and has a steep axial gradient [35]. Nominal performance characteristics for the thruster are provided in Table 2 assuming use of the BHC-1500 cathode.

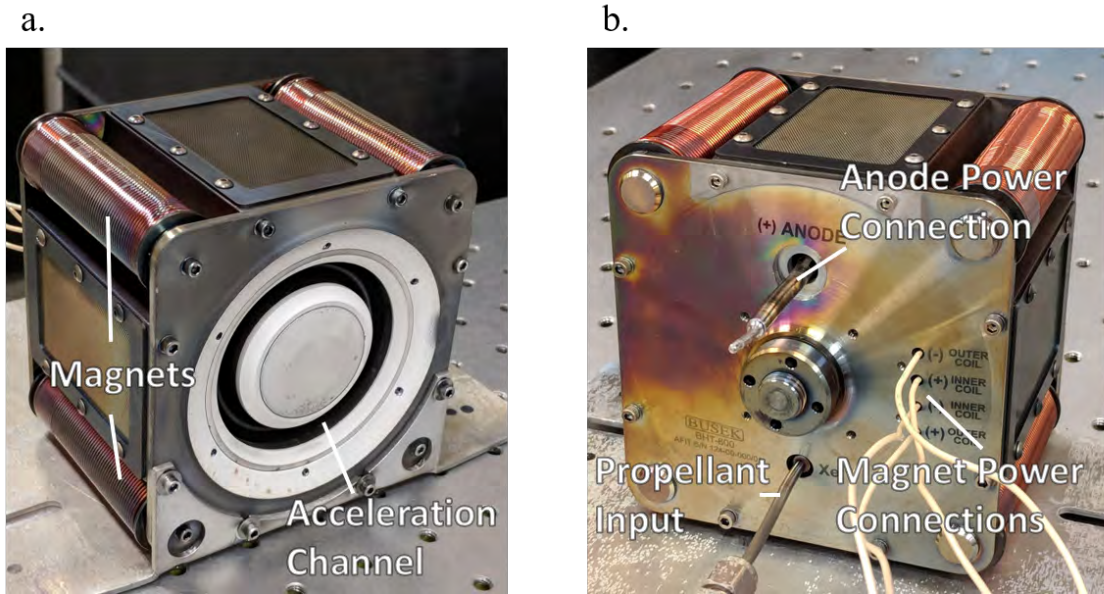


Figure 15. a. Front of the BHT-600 thruster, with visible acceleration channel and magnet coils, and noticeable erosion from ion bombardment, b. Rear of the BHT-600 thruster, showing propellant and electrical connections.

3.2.2 Power Processing Unit and Control System.

A Busek BPU-600 was used as the Power Processing Unit (PPU) for these experiments, and was rack-mounted. The BPU-600 provided a variety of voltages and currents for operation. It produced 8.0A at 12V for the cathode heater, 2.0A at 30V

for the cathode keeper, 3.0A at 12V for the first magnet, 3.0A at 12V for the second magnet, and up to 3.2A at up to 400V for discharge [35]. This thruster was controlled via Busek-provided LabVIEW® code, pictured in Figure 16. This LabVIEW® code operated the thruster by communicating with the BPU-600 over a RS-422 serial connection. This allowed for direct voltage and current control of all thruster components and provided a way to monitor actual discharge current while operating the thruster.

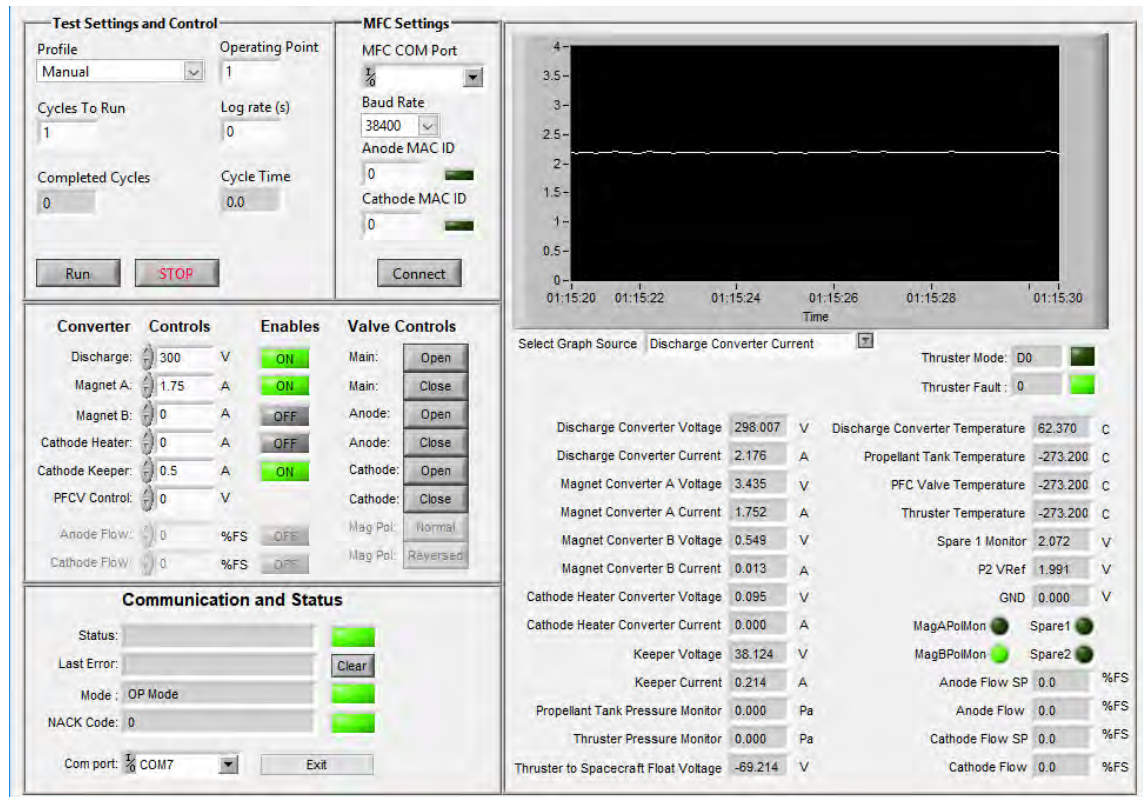


Figure 16. Busek BPU-600 control software, used to operate the thruster.

Issues with the BPU-600's cathode heater circuit caused a switch from the BPU-600 to a Sorensen DCS33-33E power supply, which can provide currents up to 33A at voltages up to 33V. It was used in constant-current mode to provide up to 7.5A for the cathode heater. This power supply was rack-mounted on the same rack as the BPU-600, allowing a simple change of source for the heater and ground terminals outside the chamber to switch over from the BPU-600 heater power supply.

Further issues with the BPU-600 led to its removal from use entirely, shortly before the collection of final data. Its various circuits were replaced with separate power supplies. The magnets were powered by an Agilent 6038A power supply, pictured in Figure 17, which can provide combinations of up to 60V at up to 10A with an overall limit of 200W. Both the anode discharge and cathode keeper were powered by two separate rack-mounted Matsusada REK650-2.5 power supplies, pictured in Figures 18 and 19 respectively, which can each provide up to 650V at up to 2.5A.

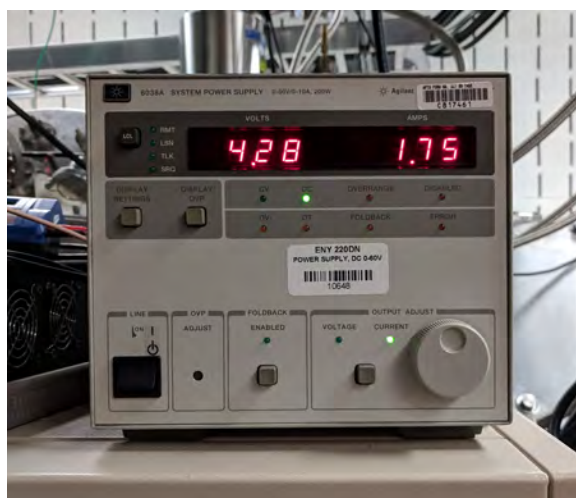


Figure 17. Agilent 6038A power supply, used to power the magnets.



Figure 18. Matsusada REK650-2.5 power supply, used to power the anode. Also pictured, MKS Type 247 mass flow readout.



Figure 19. Matsusada REK650-2.5 power supply, used to power the cathode keeper, and Sorensen DCS33-33E power supply, used to power the cathode heater.

3.2.3 Busek BHC-1500 Hollow Cathode.

A BHC-1500 hollow cathode was used as the electron source for these experiments and is pictured in Figure 20. This is a dispenser cathode, with a porous tungsten insert impregnated with a barium-calcium-aluminate mixture, and is capable of providing up to 3 amps of current [36]. The cathode has an ignition temperature of $\approx 1000-1200^{\circ}\text{C}$, which was reached initially using the heater and sustained throughout operation by ion bombardment. Typical mass flow rates are $\approx 6\%$ of the anode mass flow rate [35]; actual flow rates were sustained to match those used by Bui in his characterization of the thruster, presented in Table 2. As seen in Figure 20a, the cathode required three electrical connections. From left to right, these are the cathode body, heater, and keeper. The cathode body was connected to ground, and the heater and keeper were connected to their respective power supplies. The heater required up to 20V and up to 7.5A, and the keeper required 600V to start. When ignited, the keeper voltage dropped to approximately 40V.

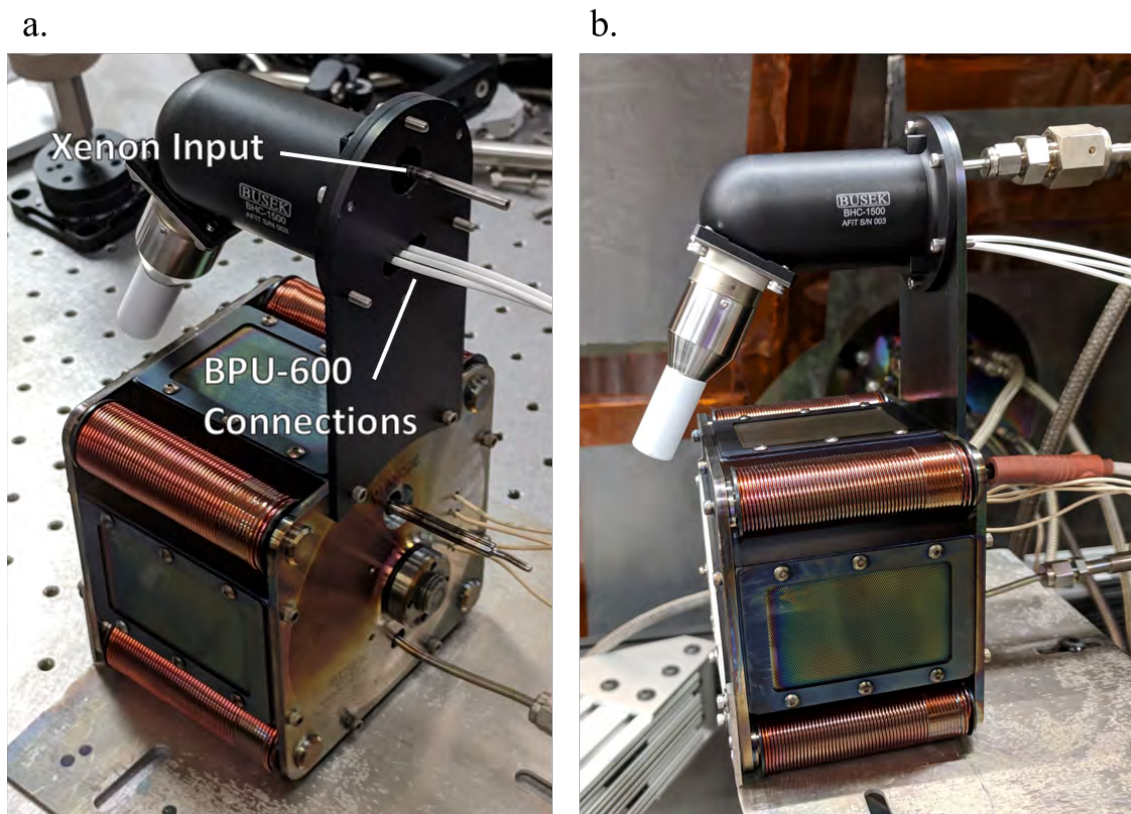


Figure 20. Busek BHT-1500 dispenser cathode, a. Rear view mounted to thruster, showing propellant and electrical connections, b. Side view mounted to thruster, installed in vacuum chamber.

3.2.4 Wiring Setup.

The thruster was wired as pictured in Figure 21. Terminals 1-5 were used by another student also operating in the SPASS Lab vacuum chamber. Terminals 6-11 were used by this research. Terminal 6 corresponded to the cathode body, which connected to ground; Terminal 7 was the cathode heater, Terminal 8 was the cathode keeper, Terminal 9 was the outer magnet coil negative, Terminal 10 was the inner magnet coil positive, and Terminal 11 connected the outer magnet coil positive and inner magnet coil negative to wire both magnets in series. The anode was connected separately with an insulated cable to avoid high voltage arcing.

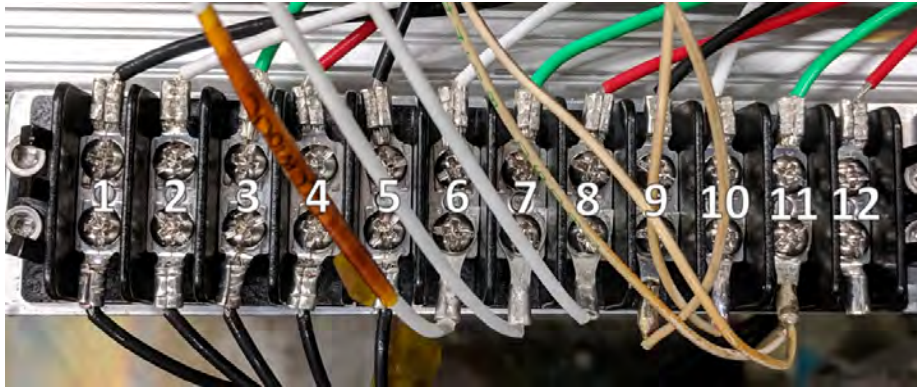


Figure 21. Interior wiring setup for Busek BHT-600 Hall thruster.

This terminal board matched exactly to a terminal board rack-mounted outside of the chamber, which made connections using banana plugs. The rack-mounted board allowed simple switching between various terminals and connected power supplies, making on-the-fly switching possible while the chamber was at vacuum. This also enabled connecting multiple thrusters at once inside the chamber; to switch between thrusters, the power supplies only required switching on the external terminal board.

3.3 Instrumentation

The primary instrumentation used was a Spectral Energies LAS system, a schematic of which is pictured in Figure 22. The system consisted of a Sacher Lasertechnik diode laser with output power $> 1\text{W}$, split into seven beams; by design, one beam is used for stationary absorber measurements while the other six are expanded into laser sheets and used to examine the Hall thruster plume. In the figure, only one beam path is shown for simplicity; this represents the six beams used to analyze the plume. For this research, only a single one of the six sheets was used, aligned horizontally to obtain a vertical distribution of axial velocities throughout the plume. This sheet was mounted 17° from perpendicular to the plume centerline. The beam was fiber-coupled to the optic ring, at which point it was expanded to a free-beam sheet through the plume, and directed as a free-beam to the camera. The laser controllers and lab computer used for collecting data from the cameras are not pictured.

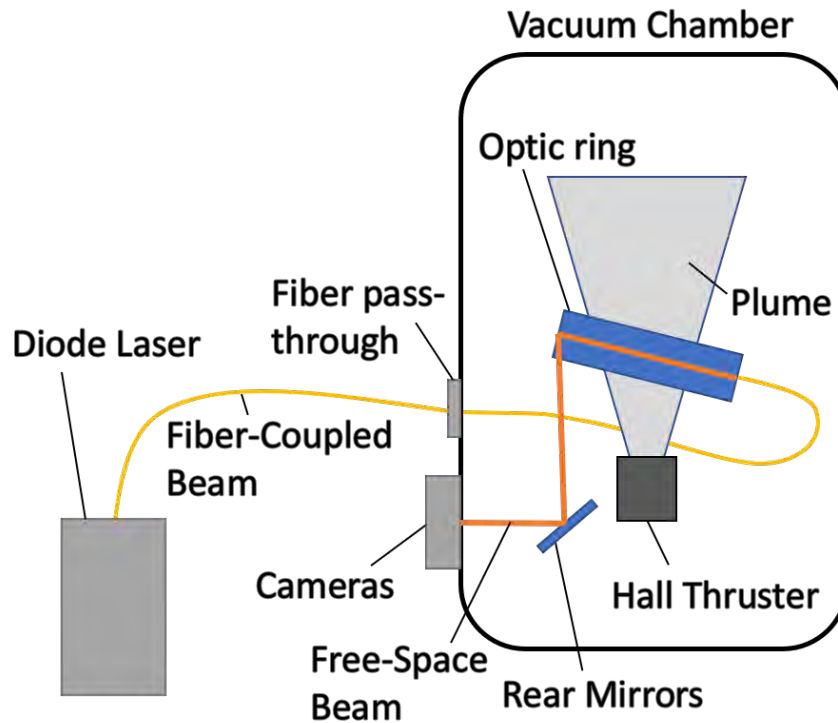


Figure 22. Simplified LAS system schematic.

3.3.1 Diode Laser and Laser Controllers.

The Sacher Lasertechnik Tec 420 diode laser was tuned to an absorption line of ionized Xenon at 834.724 nm, and was powered by two Sacher Lasertechnik laser controllers, a PilotPC 4000 and a PilotPZ 500, pictured in Figure 23. The laser itself consists of two primary components, the master diode laser and an amplifier, pictured in Figure 24. The beam path in the laser forms a U-shape using two mirrors to turn the beam. Wavelength tuning was performed using the tuning screw on the master diode laser. The PilotPC 4000 laser controller provided the main power to the amplifier, while the PilotPZ 500 laser controller offered a sweeping function as presented in Figure 25 to scan the master diode laser around the center frequency using a specified waveform.

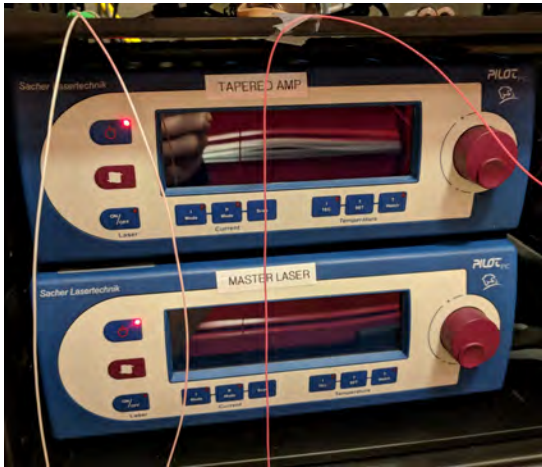


Figure 23. Sacher Lasertechnik laser amplifiers.

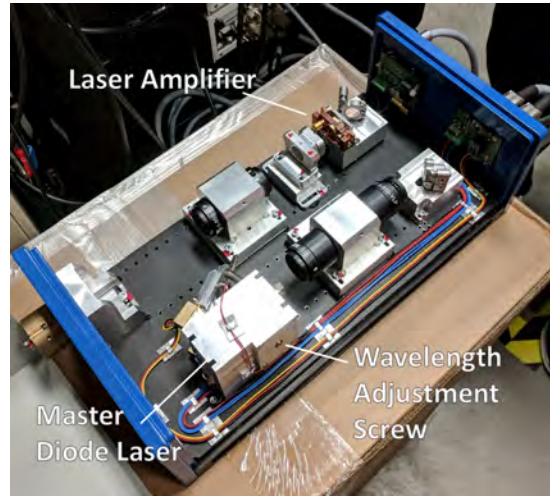


Figure 24. Sacher Lasertechnik Tec 420 diode laser with cover removed.

A triangle waveform was chosen so that each frame of camera data, taken at equal timesteps, corresponded to an equal change in wavelength. The PilotPZ 500 laser controller could oscillate the frequency for scan rates of up to 1 kHz. For the 834.724 nm tune, the laser scanned from 834.714 to 834.754 nm. This enabled the laser's use for LAS velocimetry; to see the absorption of Doppler-shifted Xenon ions

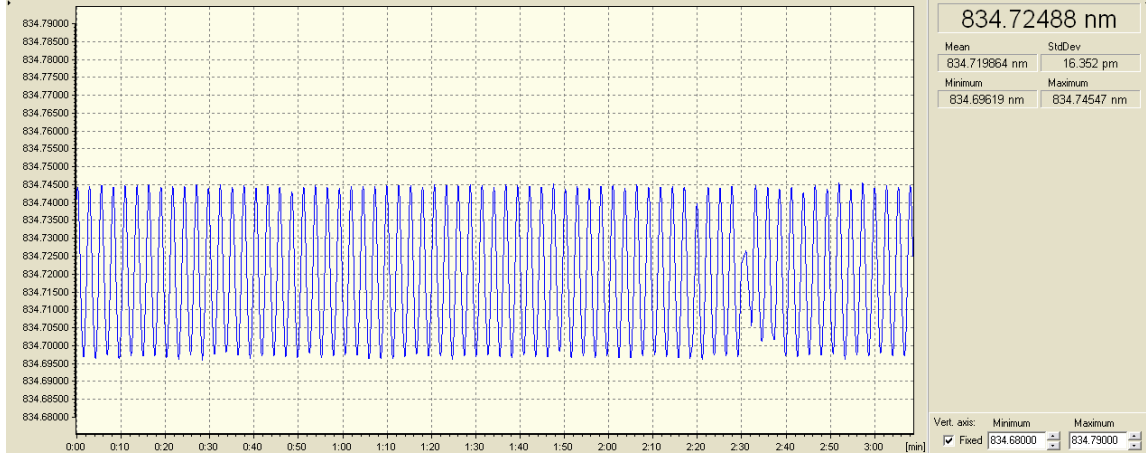


Figure 25. Scan profile for the Sacher Lasertechnik diode laser tuned to 834.72 nm

travelling at high velocities, the laser needed to scan over the potential wavelengths at which the Doppler-shifted Xenon ions would absorb. The PilotPZ 500 laser controller sent a signal to the lab computer at the start of each scan, allowing synchronization of data collection with the wavelength scan.

3.3.2 Fiber Coupling and Free-Beam Path.

The laser was fiber-coupled into the chamber through a custom flange with seven fiber passthroughs, pictured in Figure 26. The desired wavelength and single-mode fiber required creation of the custom flange as commercial parts were unavailable. A custom strain relief was 3D-printed to support the bare fiber; the exterior relief bolted on to the flange while the interior relief, pictured in Figure 27, friction-fit into the flange. The interior relief was printed in solid Ultem to avoid outgassing.

A system of optics directed the beam through the plume of the Hall thruster and out of the chamber into one of six cameras. One of the six assemblies is pictured in Figure 28. The laser was collimated with a Thorlabs F240APC-850 collimator, mounted in a Thorlabs KAD12F kinematic pitch/yaw adapter and visible in the foreground. The beam next passed through a $f = -3.9\text{mm}$ acylindrical lens, mounted on



Figure 26. Custom fiber pass-through, exterior, with strain relief.

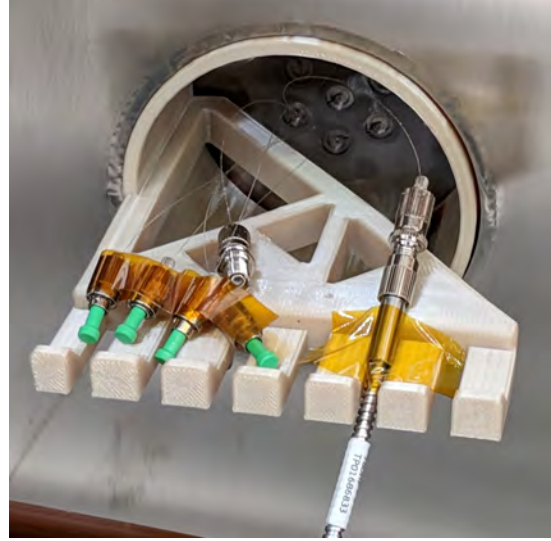


Figure 27. Interior strain relief.

a Thorlabs KM100C kinematic mount, expanding the beam into a sheet. The sheet then passed through a $f = 200\text{mm}$ acylindrical lens to form an approximately 10cm sheet through the imaging plane, mounted on a Thorlabs KM200B kinematic mount. This sheet then was directed to the cameras using protected silver mirrors, mounted in Thorlabs KM100S kinematic mounts. The sheet was focused on the camera using a $f = 60\text{mm}$, 75mm outer diameter aspherical lens. Each of the six assemblies used the same components. A seventh beam was directed through a Photron hollow cathode lamp into a Thorlabs PDA10A photodiode to provide a stationary reference absorption profile and the reference intensity. This stationary reference was necessary for determining the Doppler shift. All fiber splitting was performed using Thorlabs TW850 wideband fiber optic couplers with a variety of split percentages.

To support the optics, the chamber was modified with installation of 80/20 rails along the sides and a cross-beam through the middle of the chamber, to which the optical assembly was attached, as pictured in Figure 29. A second, shorter cross-beam attached to one of the side beams and the main cross-beam allowed adjustment of the imaging plane angle by sliding the second cross-beam along its two attachment

points. The optical ring assembly was constructed of Thorlabs 66mm dovetail rails, and attached to the 80/20 assembly using optical plate. Another 80/20 assembly was installed behind the thruster towards the end of the chamber for further mirror mounting, pictured in Figure 30. The collimators and lenses in the optical ring assembly, as pictured in Figure 29, would pass the laser sheets through the plume, then direct the sheets to the mirrors in the rear of the chamber as pictured in Figure 30. These mirrors were in the same plane as the two viewing ports and would direct the laser sheets out of the chamber to the six sets of aspherical lenses and cameras, which were flange-mounted outside the chamber to the viewing ports circled in red. For this research, only one of the six sets was utilized.

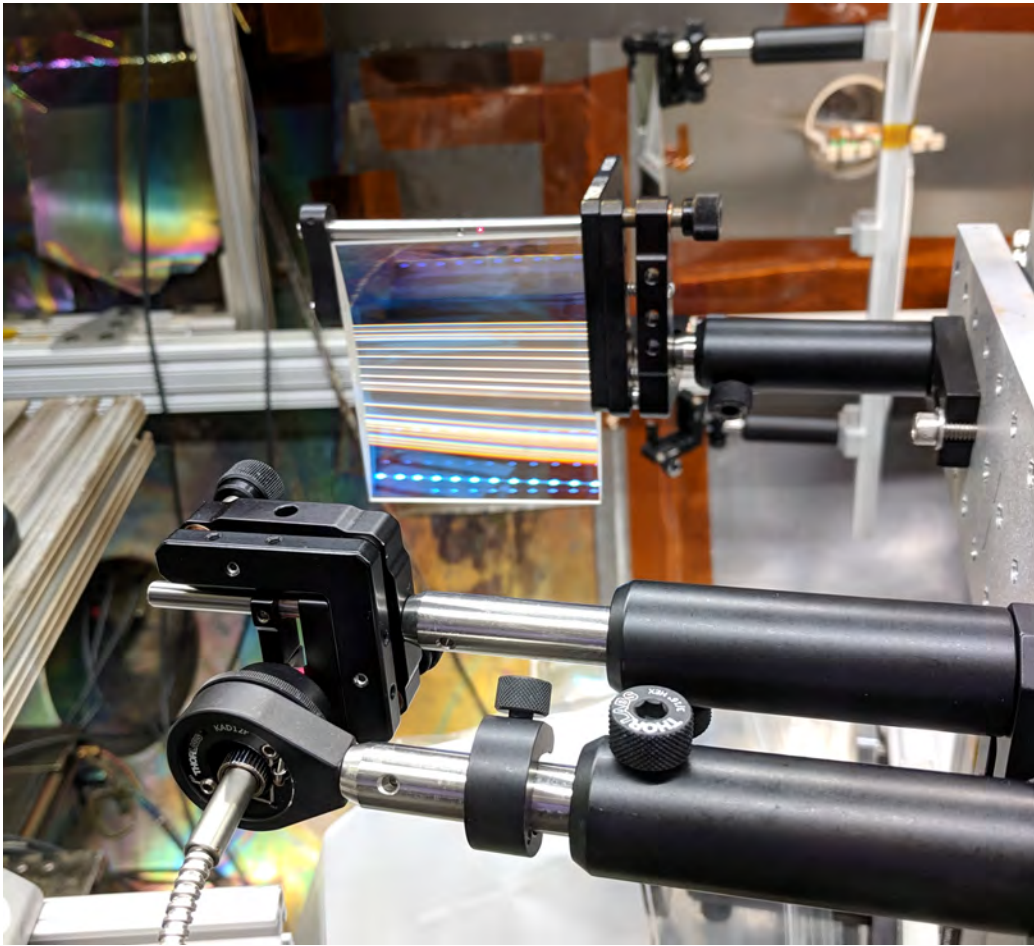


Figure 28. One of the six optic assemblies.

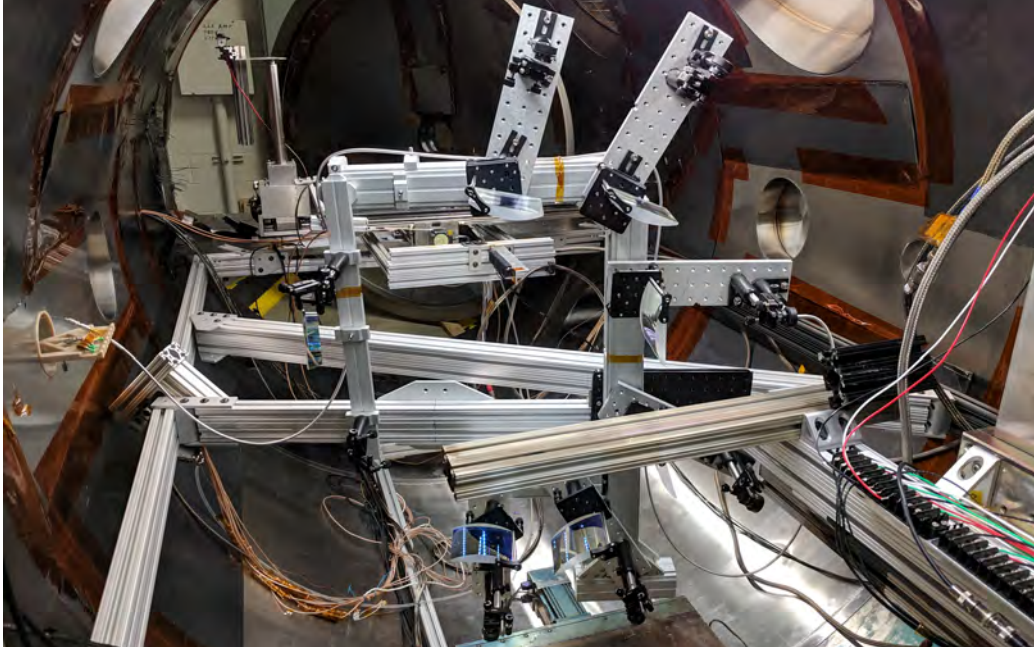


Figure 29. Ring of six collimator and lens assemblies mounted to Thorlabs 66mm dovetail rails, surrounding the imaging plane.

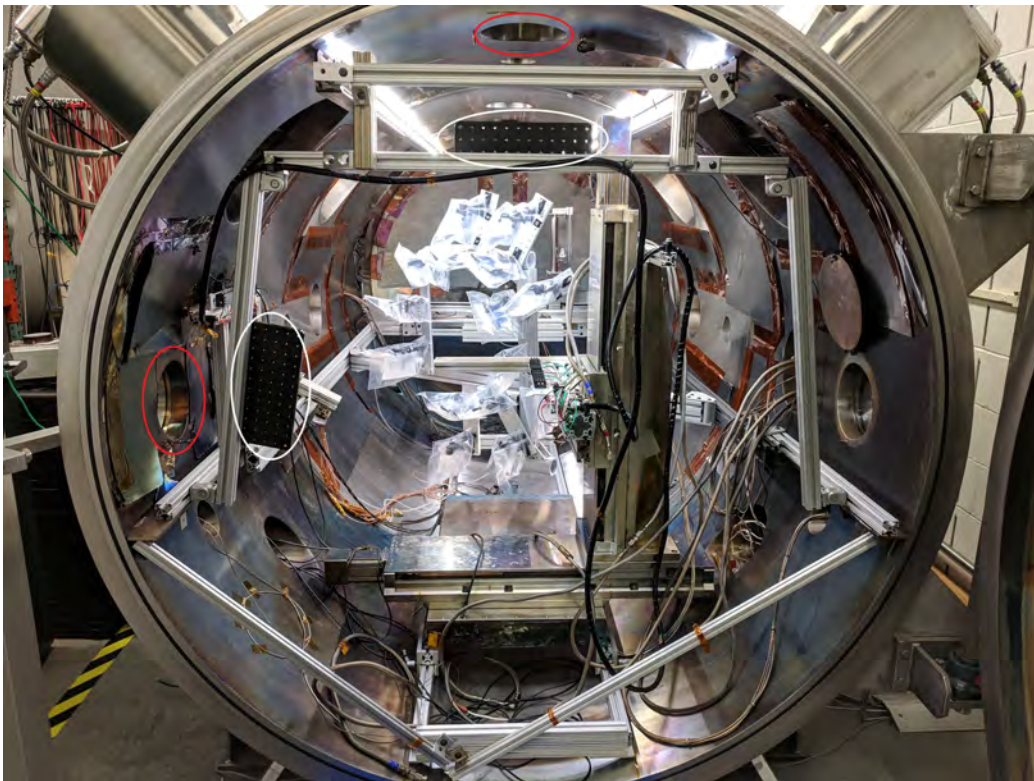


Figure 30. 80/20 mounting ring with optical plate for mounting mirrors, circled in white, which direct the beam through the red-circled viewing ports.

For this research, only the horizontal sheet was considered to gather a histogram of axial velocities throughout the plume. This sheet is highlighted in Figure 31 along its path from the second acylindrical lens to the rear of the chamber. While a translucent orange is used to represent the sheet in this image, the actual sheet was not visible to the naked eye and required an IR viewer or detector sheets to view. To avoid contamination of the other five sets of optics for the LAS system, they were removed before experimentation began, leaving only the horizontal set. This preserved the lifetime of the optics by preventing accumulation of sputtered aluminum, steel and carbon, visible on many surfaces of the chamber as a rainbow-colored coating, and also by preventing erosion by ion bombardment.

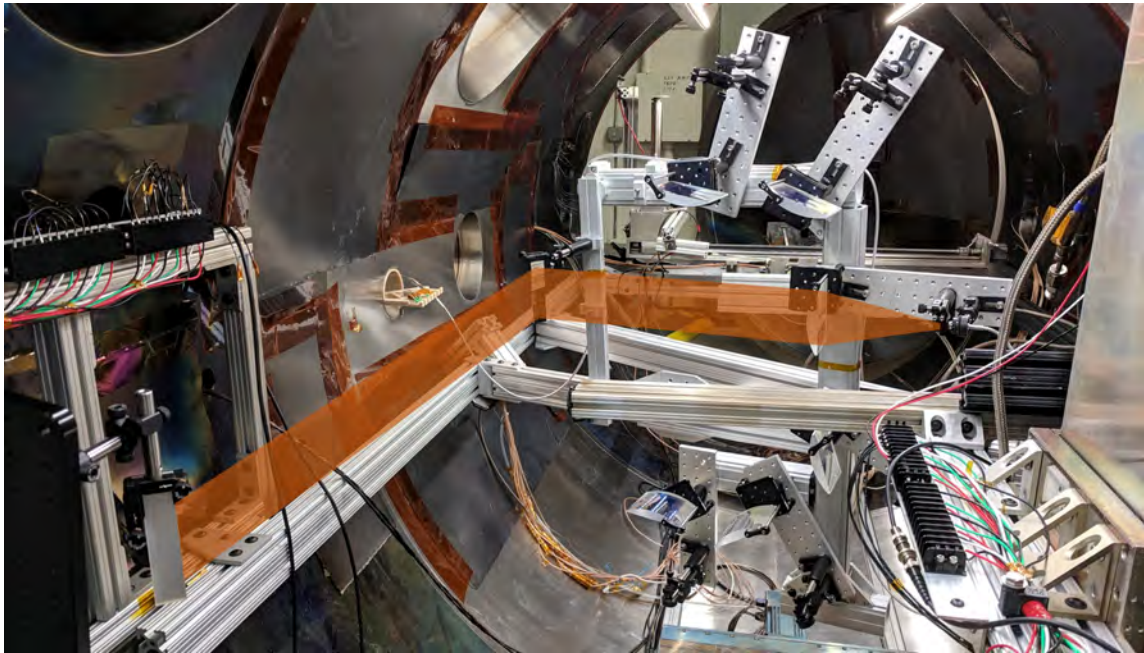


Figure 31. Sheet path for the horizontal laser sheet used to probe the Hall thruster plume.

3.3.3 Cameras.

Six Basler raL2048-80km cameras with a linear resolution of 2048 pixels were mounted to two viewing ports, on the side and top of the chamber, using a custom 3D-printed assembly pictured in Figure 32. The side-mounted assembly was shrouded with a cage (not pictured) to avoid disturbing the cameras and misaligning the system. At 834.72 nm, the Basler cameras had a quantum efficiency of approximately 0.46 electrons per photon in 12-bit depth mode. Exposure time was variable from $2.0 \mu\text{s}$ to 3.6 ms, with a maximum frame rate of 80,000 Hz [37]. The cameras were mounted to 3-axis translation stages, but had no rotational control. This posed an issue as the laser sheet may not be vertical coming into the camera. For this research, as only one sheet was considered, a temporary fix removed the lower camera on the side assembly and used its mount with a variety of posts and clamps, and is visible in Figure 33. This allowed rotational control of the camera in addition to 3-axis translation.

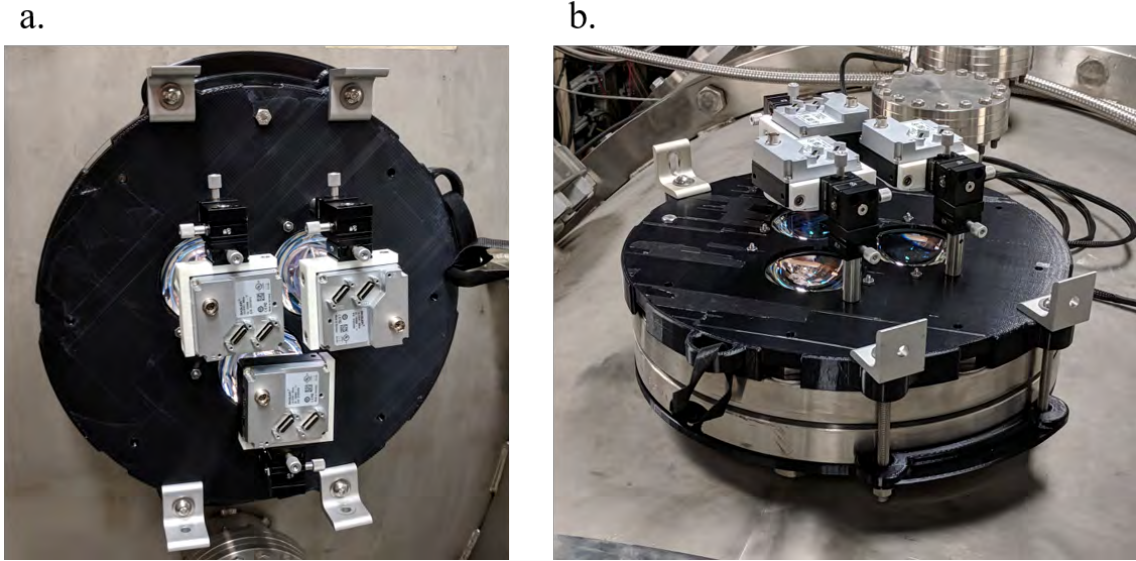


Figure 32. Camera assemblies, consisting of a custom 3D-printed housing for three lenses and Basler raL2048-80km cameras, a. Side-mounted camera assembly, b. Top-mounted camera assembly.

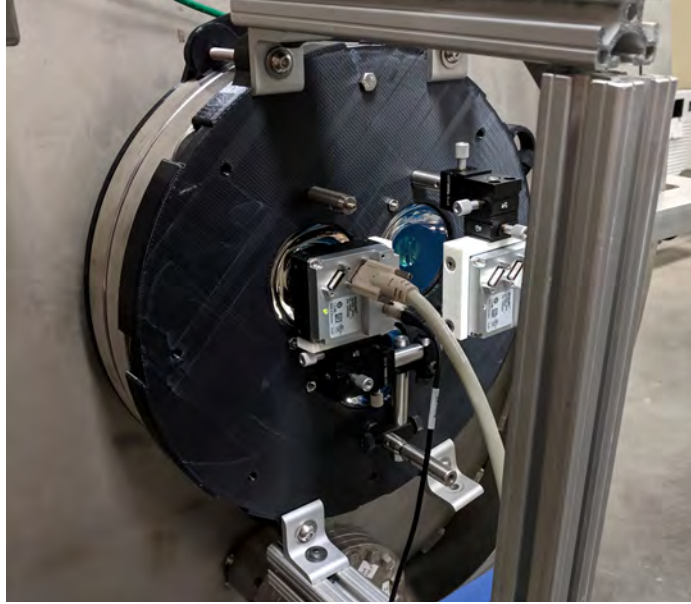


Figure 33. Modified camera assembly, with lower camera removed and cage attached.

3.3.4 Beam Path Alignment.

Alignment of the beam path was performed using a variety of methods. The laser collimator, each acylindrical lens, and each mirror had two adjustment points. The camera was mounted to a 3-axis translational stage. This resulted in 11 points of adjustment for the laser sheet. The collimator and acylindrical lens adjustments affected the position of the sheet, but also had a large role in determining the shape of the sheet. Initial alignment efforts used a fiber-coupled red 30 mW laser to roughly position the sheet through each optic and onto the sensor. This resulted in a slightly curved sheet, which was noticeable on the sensor but not to the eye, detected by the sensor either showing saturation in the center but not fully the edges, or with saturation towards the edges and a defined sheet edge, but loss of power in the center. Adjusting the alignment of the collimator and the second acylindrical lens removed this curvature. Once alignment with the visible laser was finalized, the source was swapped to the Sacher Lasertechnik IR laser. This required minor adjustments to optimize the sheet onto the sensor.

3.3.5 Surveyed Region.

It was desirable to take measurements closer to the exit plane of the thruster to avoid significant plume divergence, reducing the total number of measurements required to capture the entire plume, and protecting the lenses and mirrors from contamination and erosion by ion bombardment. However, it was also desirable to take these measurements beyond the effects of the radial magnetic field generated by the thruster's electromagnets. To determine placement of the thruster relative to the laser sheet, the thruster's magnetic field was surveyed using a LakeShore Model 410 Gaussmeter with a Model MSA-410 axial probe, as pictured in Figure 34. This determined that the peak of the magnetic field was located adjacent to the exit plane of the thruster, and that the magnetic field completely dropped back to nominal values at an axial distance of 7 cm from the exit plane.

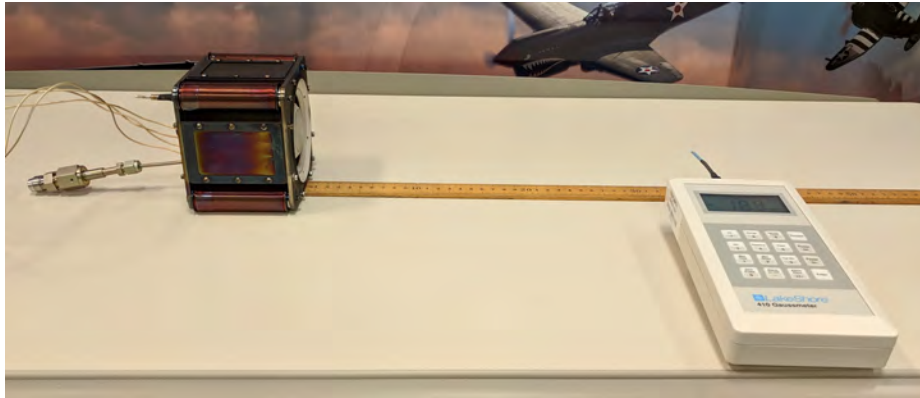


Figure 34. Tabletop setup for magnetic field measurements.

Therefore, the single horizontal sheet was used to survey the plume approximately 10 cm from the exit plane of the thruster, to obtain data slightly downstream from the region affected by the thruster's magnetic field. The mounted thruster, and horizontal optic assembly, are visible in Figures 35 and 36. In order to obtain velocity components along the axis of the laser, the laser sheet cannot be perpendicular to the flow. While this would be ideal to obtain a perfect cross-section of the plume,

the majority of the ionized Xenon would have negligible components of velocity along the axis of the laser sheet, so little to no Doppler shift would be recorded and thus no velocities would be determined. The laser sheet was therefore set to a 17° angle from the plume centerline. This thruster has a plume divergence angle of 30° [32]. Thus, the 10 cm positioning of the thruster behind the laser sheet represents the axial distance between the thruster and the point where the laser sheet first impinges upon the plume, assuming that the divergence begins from the edge of the acceleration channel of the thruster. At 10 cm from the laser sheet, with a 30° divergence angle, the plume diverged 5.7 cm. The thruster was therefore positioned so that the edge of the acceleration channel was approximately 6 cm below the centerline of the laser sheet.



Figure 35. Mounted Busek BHT-600 Hall thruster with BHC-1500 hollow cathode, and LAS system optic assembly, viewed from the rear.

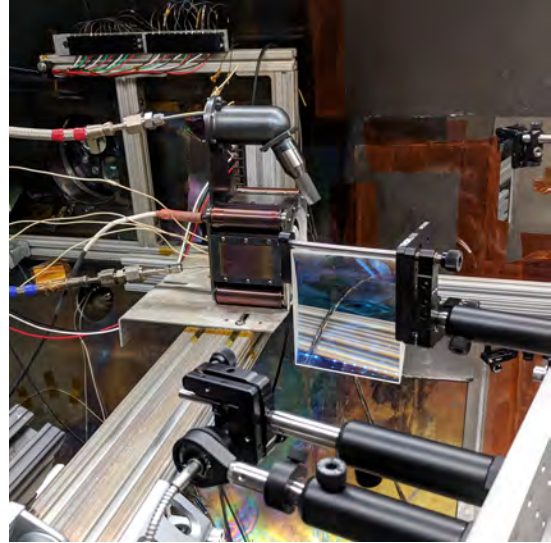


Figure 36. Mounted Busek BHT-600 Hall thruster with BHC-1500 hollow cathode, and LAS system optic assembly, viewed from the side.

The visible portion of the laser sheet was then determined by placing an object of known size into the sheet path. A single piece of 80/20, measured at 1.5 inches in width, was mounted to the rear traverse and placed into the path of the laser sheet. The resulting camera data was captured, along with a set of camera data with no

objects in the beam path. These two pictures were analyzed using MATLAB® to determine, first, the pixel-to-inch and pixel-to-mm mappings by finding where the intensity dipped as a result of the object in the path of the beam. This determined correlations of 882.67 pixels per inch, or 34.75 pixels per millimeter. The second picture, of only the laser intensity with no objects in the sheet path, was then used to determine the visible portion of the beam. The cameras record 8-bit images with integer data at each pixel from 1 to 256. A threshold of 50 was set as the minimum representing the visible portion. Analysis of which portions of the picture met that cutoff gave a total visible portion of 1,393 pixels, or 40.08 mm.

At the determined thruster distance of 10 cm from the laser sheet, the plume spread approximately 19 cm. Thus, to fully capture the plume, with some excess to the sides, six sets of data were collected. Using the front translation stage, the thruster was translated vertically in 40 mm increments to capture the entire plume.

3.3.6 Data Collection.

LabVIEW® code provided by Spectral Energies took in data from the six cameras and the reference photodiode and saved this to separate files for post-processing. Only a single one of the six cameras was enabled for this research, so the only files written were for the first camera and the photodiode. The laser scanned across wavelengths of interest at a constant rate using a triangle wave. The cameras recorded the intensity as a function of time as 8-bit depth images with integer values from 1 to 256, and the photodiode recorded intensity as a voltage. The PilotPZ 500 laser controller sent a high (5V) transistor-transistor logic (TTL) signal to the computer each time the laser crossed the center wavelength scanning to the highest wavelength, and a low (0V) TTL signal when crossing the center scanning to the lowest wavelength. This allowed correlation of the laser's wavelength with camera data, and triggered the

capture of data with the high TTL pulse. By correlating this intensity data with the wavelength scanning data from the laser controller, the wavelength at each instant in time was determined. Observed intensity decreased where the singly-ionized Xenon (Xe II) absorbed the laser. This occurred at each pixel in the camera; each pixel corresponded to a different spatial component of the plume.

The wavelengths across which the laser was scanned were measured with a wavemeter and separately recorded each time data was collected. This was performed using a borrowed Bristol Model 621 wavemeter, pictured in Figure 37. The wavemeter was fiber-coupled into the system. While this in theory would allow observation of the wavelength during testing, data was collected with the PilotPZ 500 scanning at 500 Hz, and observing the highest and lowest wavelengths between which the laser swept required using the lowest scan frequency possible of 0.1 Hz. Real-time determination of the wavelength was therefore not possible.



Figure 37. Bristol Model 621 wavemeter, fiber-coupled to the LAS system.

3.4 Test Conditions

To validate the LAS system against prior experimental data obtained by Bui [32], the Busek BHT-600 Hall thruster was operated at the same operating conditions. These conditions are presented in Table 3 with derived currents and powers presented in Table 4. Propulsion-grade Xenon gas was used as the propellant. For Case 1, the discharge voltage and magnet current were chosen to match the nominal values as suggested by the manufacturer as well as the prior experiment at AFIT, while the anode and cathode mass flow rates differ from manufacturer nominal values to match the prior experiment. Cases 2 and 3 adjust the discharge voltage while keeping all other conditions constant, matching other conditions tested by Bui.

Table 3. Set thruster operating conditions.

	Case 1	Case 2	Case 3
Discharge Voltage	300V	200V	100V
Anode $\dot{m}$	2.66 mg/s	2.66 mg/s	2.66 mg/s
Cathode $\dot{m}$	0.146 mg/s	0.146 mg/s	0.146 mg/s
Magnet Current	1.75A	1.75A	1.75A

Table 4. Determined thruster operating conditions.

	Case 1	Case 2	Case 3
Discharge Voltage	300V	200V	100V
Discharge Current	2.0A	2.05A	2.45A
Discharge Power	600W	410W	245W

Issues with the chamber on the date of testing, discussed in Chapter IV, led to actual testing for only Case 1 conditions with a reduced anode mass flow rate of 1.23 mg/s Xe. All other conditions were maintained the same. The results are still comparable to those obtained by Bui, as the same discharge voltages were used. This indicates that similar exit velocities will be obtained, as the exit velocity is not a

function of the mass flow rate. For all test cases, the master laser controller and amplifier controller were set to the same settings as presented in Table 5. The laser was allowed to come to steady-state conditions by running for one hour before data was taken. Temperatures were maintained through built-in thermoelectric cooling.

Table 5. Laser operating conditions.

Discharge Current	1835 mA
Offset	4.100 V
Amplitude	8.201 V
Max Wavelength	834.754 nm
Min Wavelength	834.714 nm
Sweep Frequency	500 Hz

These operating conditions, and the mounted angle of the laser sheet with respect to the plume centerline, result in performance characteristics for the LAS system as presented in Table 6. These performance characteristics are subject to change if the amplitude, offset, or center tuned frequency of the laser are modified. The wide range of velocities with which the system can resolve indicates that it can also be tuned for use with other forms of Xenon-based electric propulsion systems with higher exhaust velocities than Hall thrusters. The maximum and minimum axial velocity are based on the wavelengths between which the laser sweeps, the resulting potential calculated Doppler shifts and therefore velocities along the laser axis, and the corresponding axial velocities based on the 17° angle of the laser sheet. When operated at the maximum frame rate, the limit on the camera's buffer of 32,000 frames limits the continuous sampling time to 0.4 seconds. This corresponds to 200,000 samples from the data acquisition device (DAQ), which records the photodiode and TTL signals.

Table 6. LAS system performance characteristics.

Maximum Axial Velocity	36.85 km/s
Minimum Axial Velocity	-12.28 km/s
Resolution	0.622 km/s
Velocity Bins	80
Camera Frame Rate	80,000 Hz
Maximum Camera Frames	32,000
DAQ Sample Rate	500,000 Hz
Sample Time	0.4 s

3.5 Uncertainty Analysis

To quantify the uncertainty of these measurements, the resulting data is analyzed to estimate the standard deviation. Data was collected over 0.4 seconds, with the cameras operating at 80,000 Hz. They captured 32,000 frames per set of data, which equates to 200 full sweeps of the laser. The last sweep was lost in the data analysis process to align the various sets of data, so 199 sweeps were available for analysis. To define a 68% confidence interval for the absorbance and velocity calculations, the results were analyzed to determine the standard deviation.

For velocities, this was performed by taking the mean of the absorbance-weighted average velocity at each sweep. This mean was then subtracted from the value calculated for each sweep. This amount was then squared, and summed; the square root was taken of the result, determining the standard deviation for velocity for that pixel. This was performed at each pixel, for each set of data. Similarly, for absorbance, the same calculations were performed at each pixel to determine the standard deviation of absorbance. The results were used to add error bars representing the 68% confidence interval on relevant data discussed in Chapter IV. The calculated standard deviations for both absorbance and velocity are also presented and discussed in Chapter IV.

IV. Results

This research provided the first measurements on a LAS system designed to obtain full-field, simultaneous Xenon ion number density and velocity profiles in the plume of a Busek BHT-600 Hall thruster. These first measurements were obtained using a single one of the six laser sheets, aligned horizontally to capture a vertical histogram of axial ion velocities. This data was then used to validate the full system against data from Bui's 2012 Master's thesis [32], taken using intrusive probes.

Numerous difficulties led to results only being taken for the first test case, in terms of discharge voltage, and at an anode mass flow rate lower than the nominal mass flow rate as used by Bui. When data was taken, at the highest thruster position, part of the traverse or propellant line setup impacted the optical plate holding the laser collimator and acylindrical lenses. This pulled the plate downward and misaligned the laser sheet upward off the first mirror. Fixing this issue would require bringing the chamber back to atmosphere and re-aligning the system. Thus, no further data was captured due to time constraints. Anode mass flow rates were reduced compared to the nominal conditions to 12.5 SCCM or 1.23 mg/s Xe, as higher mass flow rates caused the chamber pressure to exceed the operating limits. This issue may have been a result of the air conditioning freezing up due to a strong snowstorm, leading to soaring temperatures in the lab of over 100° F the week of testing. It also may have been a result of increasing the chamber volume without increasing the pump capacity, or other issues with the pumps; however, the thruster was previously operated at conditions matching those determined by Bui without issue so the excessive ambient temperature was likely the primary cause.

4.1 Post-processing

The LabVIEW® code provided by Spectral Energies merely wrote out the camera data and DAQ data to files; all absorption analysis and data processing occurred in post-processing using MATLAB®. The Spectral Energies LabVIEW® code saved out a pseudo-tab-delimited file which required modification for input into MATLAB®, so an assortment of functions and scripts were created to pull in all relevant data files and fix them into a form that can be read.

The camera data and DAQ data were saved as two separate files for each scan. The camera data included an array of intensities for each pixel for each frame captured by the cameras, for a total of 32,000 frames. This was the maximum which could be taken at a single time by the cameras before requiring a write to disk, and corresponds to 160 frames per laser sweep. Data was captured for all 2,048 pixels; the camera was orientated such that higher-index pixels correspond to physically higher physical locations in the plume, such that pixel 2048 corresponds to the top-most physical location and pixel 1 corresponds to the bottom-most physical location surveyed in a given set. The DAQ data had three columns; the first was the TTL pulse where the second was the photodiode output. The third input was not connected and recorded noise; this was unused and thrown out. The DAQ data saved out at 1000 samples per laser sweep; however, this took data immediately, while the cameras did not start taking data until after the first high (+5V) TTL pulse. Thus, the files did not line up directly. To fix this issue, the inflection point where the TTL pulse went high for the first time was determined, and every data point before this was discarded. The remaining data was split into individual complete laser sweeps based on the TTL pulse changepoints, and matched to the corresponding camera data.

Reference data was taken for both the camera and the photodiode, to account for the effects of the path length, lenses, mirrors, and chamber glass on the laser sheet,

and the effects of the glass and neutral Xenon in the hollow cathode lamp on the reference intensity beam, respectively. For each, the data was split into individual complete laser sweeps, and averaged to determine the average complete sweep, visible in Figures 38 and 39. When the raw data was read, the averaged references were re-sized to match the size of the raw data, and then subtracted from it, for both the camera and photodiode.

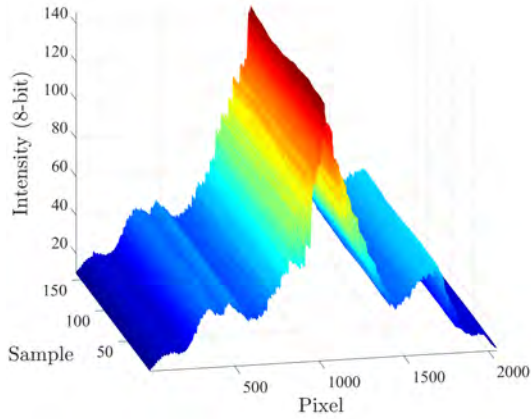


Figure 38. Averaged camera reference over one sweep of 160 camera frames.

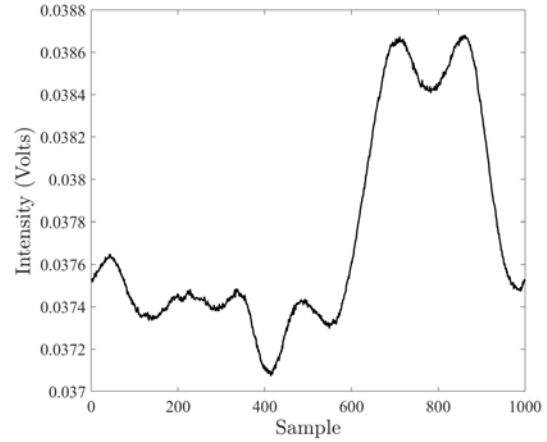


Figure 39. Averaged photodiode reference over one sweep of 1000 DAQ frames.

The observed sheet profile in Figure 38 appears similar to a Bessel function of the first kind, visible in Figure 40. The actual beam profile would appear as a Bessel series, but is well represented by the Bessel function of the first kind. The acylindrical lenses expand this profile into a sheet, maintaining the same shape but stretching it. In theory, the wings should have the same magnitude of intensity; however, the rightmost wing appears to have a higher intensity than the leftmost. This is likely a result of the various lenses and mirrors used to direct the laser from the collimator to the camera; the mirrors could have differentials in reflectivity, or the lenses or viewport could have differentials in transmissivity.

Absorption data was then determined by choosing what indices in the photodiode data correspond to the maximum and minimum wavelength which the laser sweeps

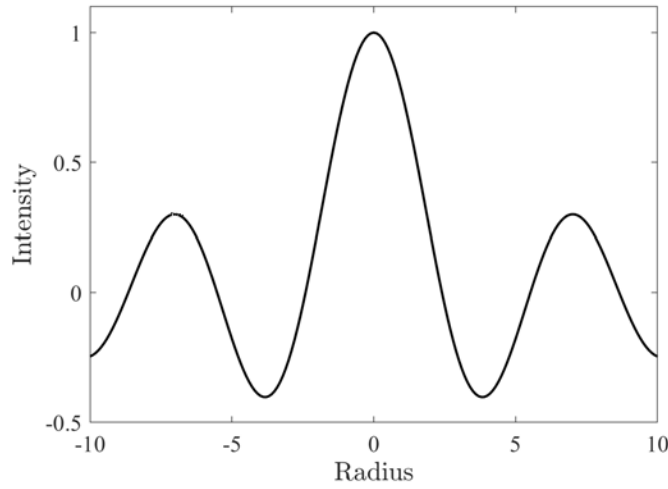


Figure 40. Bessel function of the first kind, generated using *besselj* with MATLAB®.

between. Each full sweep, as determined by the TTL signals, contained a single full max-to-min wavelength sweep. These points were then used to find the corresponding indices from the camera data, which were separated and indexed by sweep number. Relative absorption data was then determined for each, sweep by sweep, by interpolating to find the index of the photodiode data from the DAQ which corresponded to the camera frame, and taking the natural log of the quotient of the intensity data from the camera and the reference intensity from the photodiode.

This, however, led to issues. The resulting absorbance data was calculated using the same reference intensity, as determined by the photodiode. This caused a few problems; first, the reference intensity was passing through a hollow cathode lamp which could affect the observed intensity by the photodiode in ways which the beam sent through the chamber was not affected. Second, this divided an entire row of data by a single value. Observation of the reference data from the cameras, as seen in Figure 38, shows that the reference intensity at each pixel is not constant, as the optics expand the incident beam pattern into a sheet. Thus, the proper reference intensity should be based on the camera data, to reflect this distribution.

To determine the reference intensity in this manner would require construction of a mapping of photodiode intensity to camera reference, so the proper camera reference frame could be used as the reference intensity against a given data point. However, analysis of the photodiode intensity data taken during experimentation, visible in Figure 41 revealed that a sinusoidal noise overpowered the laser. The signal is presented as an average over all sweeps from Set 1, as a function of the time since the start of the sweep, which corresponds to 1000 samples from the DAQ sampling at 500,000 Hz. The intensity of this signal was more than double the signal from the laser; the only difference between the two cases was the hollow cathode lamp being turned on to 5 mA. It was determined that the most likely cause was that small visible emissions from the hollow cathode lamp overpowered the signal from the laser, given the relatively low intensities observed.

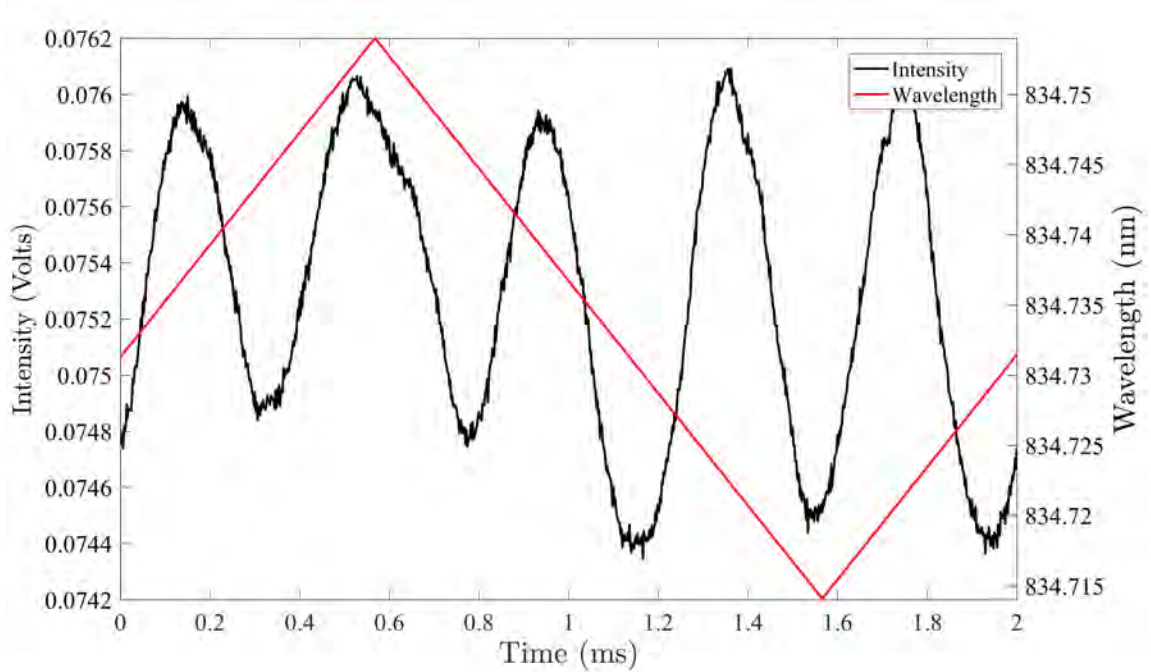


Figure 41. Averaged photodiode signal from data set 1 over one sweep of 1000 DAQ frames, presented as photodiode intensity as a function of the time since the start of the sweep, with the laser wavelength also as a function of time since the start of the sweep.

Due to all photodiode data taken effectively only containing noise, other than the reference, a new technique was developed to obtain usable results from the existing data. Because the cameras were triggered by the high TTL pulse, all camera data could still be aligned with the reference photodiode data. This was used to find which portions of camera data corresponded to the full wavelength sweep from 834.754 to 834.714 nm. These portions of the raw camera data were then used as the observed intensity data, with the raw camera reference used as the reference intensity, to determine relative absorption. All sets of camera data were taken with the same total number of samples, so the actual data was compared directly to the reference without averaging. Because the same data source was used, no scaling was required and relative absorption (α_{rel}) was determined using:

$$\alpha_{rel} = 1 - \frac{I}{I_i} \quad (65)$$

However, the resulting absorption data had an undesirable low-frequency noise component at approximately 124 Hz. This was removed through a bandstop filter instead of a highpass filter to preserve the DC offset of the signal. A lowpass filter was also applied to remove high-frequency noise above 12,500 Hz. Both standard filtering and zero-phase filtering were implemented; however, as presented in Figure 42, the phase shift caused by the standard filtering appeared to have negligible effects on the resulting data. Thus, data filtered only by the standard bandstop and lowpass filters were used. Both the raw camera data and the reference camera data were filtered.

For the camera, 80 points were taken over the full sweep. With the laser sheet mounted at a 17° angle to the plume centerline, this resulted in a velocity distribution of -12.28 km/s to 36.85 km/s in increments of 0.622 km/s, calculated by the Doppler shift from the Xe II absorption line at 834.724 nm given the maximum and minimum wavelengths. Each camera frame was taken at a constant frame rate, corresponding

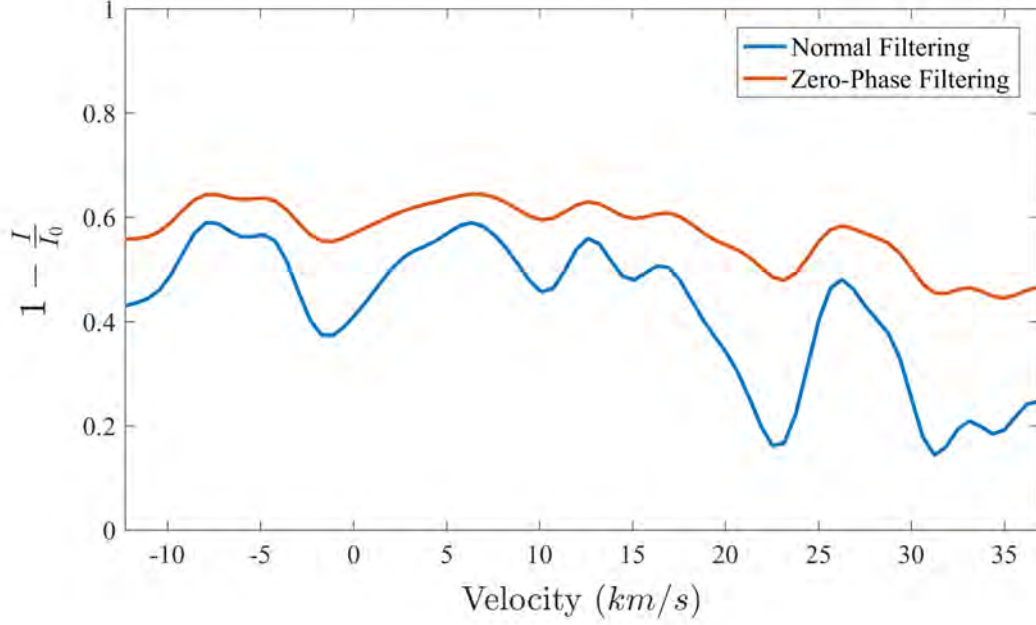


Figure 42. Relative absorption as a function of estimated axial velocity, calculated from Doppler-shifted wavelength, using data processed with both normal filtering methods and zero-phase filtering.

to a constant wavelength sweep of the laser, and thus a constant velocity increment. Therefore, the 80 points in each full sweep were directly correlated to this velocity distribution.

Given the operating conditions of the thruster with a discharge voltage of 300V, and given that this sensor was only capable of picking up Doppler-shifted singly-ionized Xenon based on known absorption lines of higher-order ionized Xenon, the theoretical maximum exit velocity was 20.9 km/s. However, that value is only obtained if the ionized Xenon particle is accelerated by the entire electric field. In reality, and in Figure 42, a distribution of velocities were observed between zero and the theoretical maximum. However, there was also absorption at wavelengths corresponding to Doppler-shifted velocities outside this range. Bui took ion current density data at various axial distances and angles relative to the plume centerline [32], which shows that at the given plume divergence angle of 30°, depending on axial location, the

current density is approximately five to six percent of the maximum current density. His measurements were obtained at the same operating conditions as this research, but at a higher anode mass flow rate, and are presented in Figure 43.

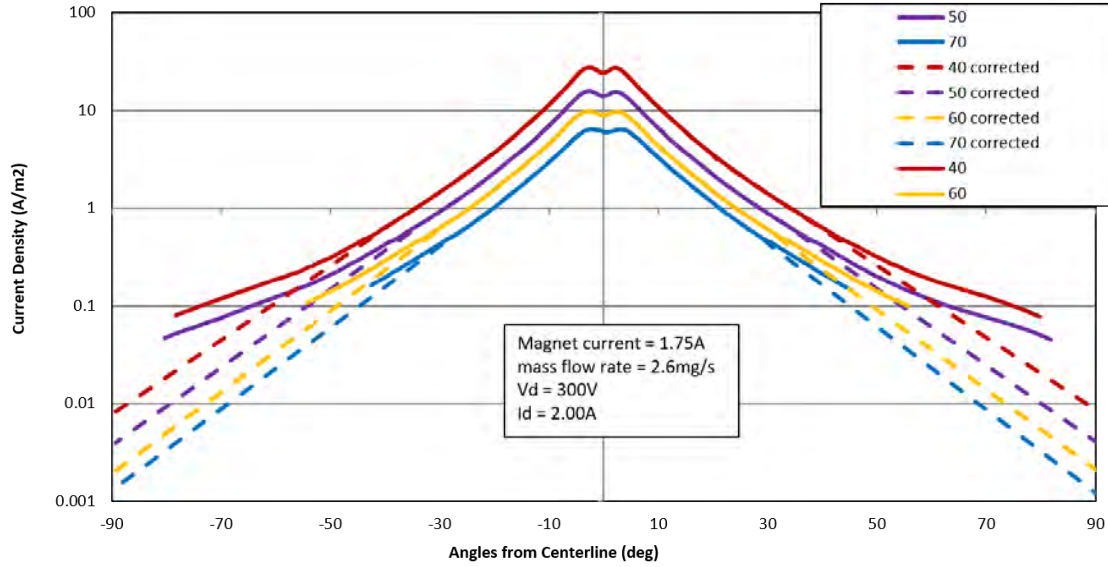


Figure 43. Ion current density in the plume of a Busek BHT-600 Hall thruster, operating on Xenon propellant at 300V discharge, at various axial locations (cm), from Bui [32].

Bui's data shows that significant amounts of Xenon ions in the plume diverge from the centerline, with all positions reporting values dropping an order of magnitude at approximately 25° . Any particle with a velocity vector which is not aligned with the centerline will cause a misrepresentation of velocity, as the codes processing the raw data assume that the 17° offset between the plume centerline and the laser axis applies to every particle. To consider the possible velocities that may be seen, cases were considered where the particle was accelerated over the full electric field to a velocity of 20 km/s at the widest divergence angles, 30° with respect to the thruster centerline in the direction of the laser, and 30° with respect to the thruster centerline in the opposite direction of the laser (toward the laser collimator). Considering the first case, this particle would have a laser-axis velocity component of 14.6 km/s, which the code would convert to an axial velocity of 50.0 km/s. Considering the second case,

this particle would have a laser-axis velocity component of -4.5 km/s, resulting in a reported axial velocity of -15.4 km/s. There are some ions which diverge even further than the reported divergence angles; these could potentially cause reported values even further beyond these extremes. Even considering a smaller region represented by a cone with a divergence angle of 10° , with a bulk of the ion density present, would result in a range of potential velocities from 8.3 km/s to 31.1 km/s, again assuming that the particles exit at 20 km/s.

Bui also captured velocity data along the centerline using an $E \times B$ probe positioned 60 cm from the exit plane of the thruster. Bui determined three major peaks, representing Xe II (singly-ionized Xenon), Xe III (doubly-ionized), and Xe IV (triply-ionized) [32]. His results for nominal thruster conditions, with a discharge voltage of 300 V, are presented in Figure 44. Because this LAS system was tuned to a wavelength representing an absorption line of Xe II, and no absorption lines for Xe III or Xe IV exist in the wavelength range which was swept [38], nor any absorption lines for nitrogen or oxygen [39], only Xe II would absorb. Xe II was detected at plate voltages ranging from 18 to 24 V. Using Equation 66, where B is the magnetic field for the sensor's permanent magnets given as 0.7 T, d is the distance between the probe's plates given as 0.003 m, and V_{probe} is the plate voltage, ion velocities can be determined from Bui's data. This shows that the majority of axial Xenon ions travel at velocities of 10.0 km/s, represented by the peak at 21 V, with a range from 8.57 km/s to 11.4 km/s represented by the edges at 18 and 24 V.

$$V_{probe} = Bdu_i \quad (66)$$

These values, as determined by the given calibration settings for the probe and measured voltages, indicate a lower velocity than was expected. Bui reported the I_{sp} as 1267 seconds for this thruster; this corresponds to an exit velocity of 12.42 km/s.

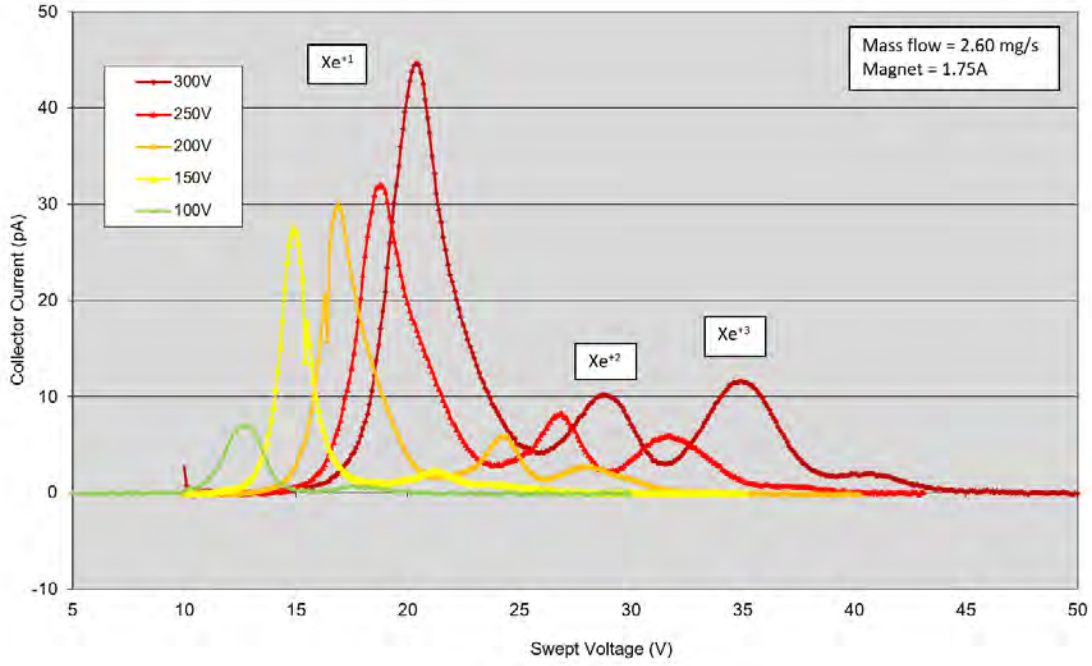


Figure 44. $E \times B$ probe collector current with visible peaks relating to Xenon ion species in the plume of a Busek BHT-600 Hall thruster, operating on Xenon propellant at 300V discharge, from Bui [32].

However, this value includes both the effects of the ionized, accelerated Xenon as well as the un-ionized, neutral Xenon. Based on an assumed mass utilization efficiency of 0.75, the exit velocity for the ions would be 16.56 km/s. As such, we expect the given magnetic field for the probe is incorrect; a magnetic field strength of approximately half the reported value would give velocities close to those expected for this thruster. This is contributed to by each ionized state of Xenon, from single ionization to quadruple ionization. Bui reported ionization fractions as given in Table 7. From this data, we can determine that the singly-ionized Xenon exits at approximately 15.04 km/s.

This exit velocity, combined with the various divergence angles, was used to determine the possible axial velocities which may be reported by the LAS system. By examining the ion current density at each divergence angle, a relationship between the ion current density and these reported axial velocities was determined, and is

Table 7. Ionized Xenon species fractions for discharge at 300V, from Bui [32].

Species	Fraction
Xe II	0.825
Xe III	0.093
Xe IV	0.072
Xe V	0.01

presented in Figure 45. Here, the ion current density is analogous to absorption, so the shape of this data can be compared to that in Figure 42. This data appears to correlate with the absorption plateau between approximately 5 and 20 km/s.

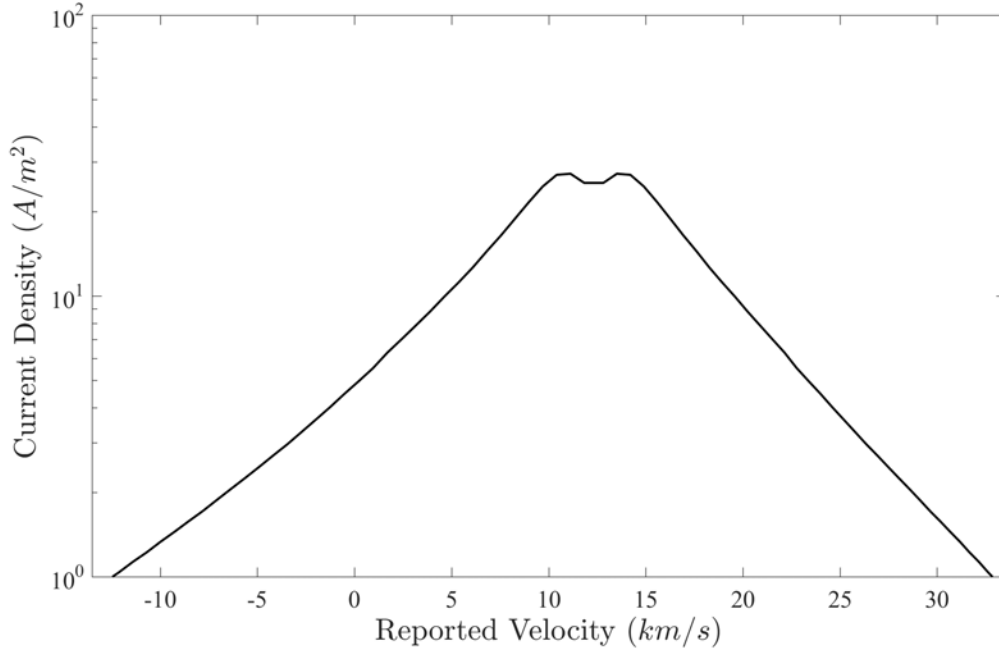


Figure 45. Ion current density (values from Bui [32]), which is analogous to absorption, as a function of reported axial velocity.

Absorption observed at wavelengths corresponding to negative reported velocities, however, is not well explained through this analysis. While this data shows that there is some ion current density at those reported velocities, it does not show an equivalent amount of current density in those regions compared to the region between 5 and 20

km/s, while we observe approximately equal absorption between the two regions. Absorption observed at wavelengths corresponding to approximately 26 km/s was likely partially due to the long path length of the laser, and the presence of neutral Xenon in the chamber. Neutral Xenon has an absorption line at 834.745 nm, the wavelength which would correspond to a Doppler shift for singly-ionized Xenon at approximately 26 km/s. Absorbance in this region could also have resulted from ionized Xenon with large divergence angles in the direction of the laser sheet. Similar features were observed in other sweeps. Therefore, further analysis considered only the data points corresponding to velocities between 0 and 20.9 km/s, which could only be a result of the ionized Xenon. The resulting data has arbitrary units, useful for comparison to one another so long as scaling is maintained constant. This stems from the camera intensity output having unitless integer values. Thus, relative absorption and velocity distributions were determined.

4.2 Velocity Distributions

Overall velocity distributions were taken by averaging data across each full sweep. With 160 frames per sweep from the cameras, and a total of 32,000 captured frames, a total of 200 full sweeps were obtained at each position in the plume. These were captured at 500 Hz, taken over 0.4 seconds. These distributions were obtained by taking the dot product of the vector of velocity distributions with the corresponding vector of absorbance at each velocity bin, and dividing by the sum of the absorbances, to find the average velocity as weighted by absorbance. This was obtained for each sweep. The average absorbance was also obtained for each sweep. This resulted in two vectors at each pixel for each data set, containing the average result of each sweep, spanning 0.4 seconds of thruster operation. Mesh plots of the velocity distributions for each data set are presented in Figure 46, and mesh plots of the absorption

distributions are presented in Figure 47. Taking the dot product of these two vectors and dividing by the sum of the average absorbances then obtained the average exhaust velocity, again weighted by the average absorbance of the sweep. Averaging the absorbance across all sweeps at each pixel obtained the overall average absorbance. Figure 48 depicts the average absorbance and velocity at each pixel for each data set.

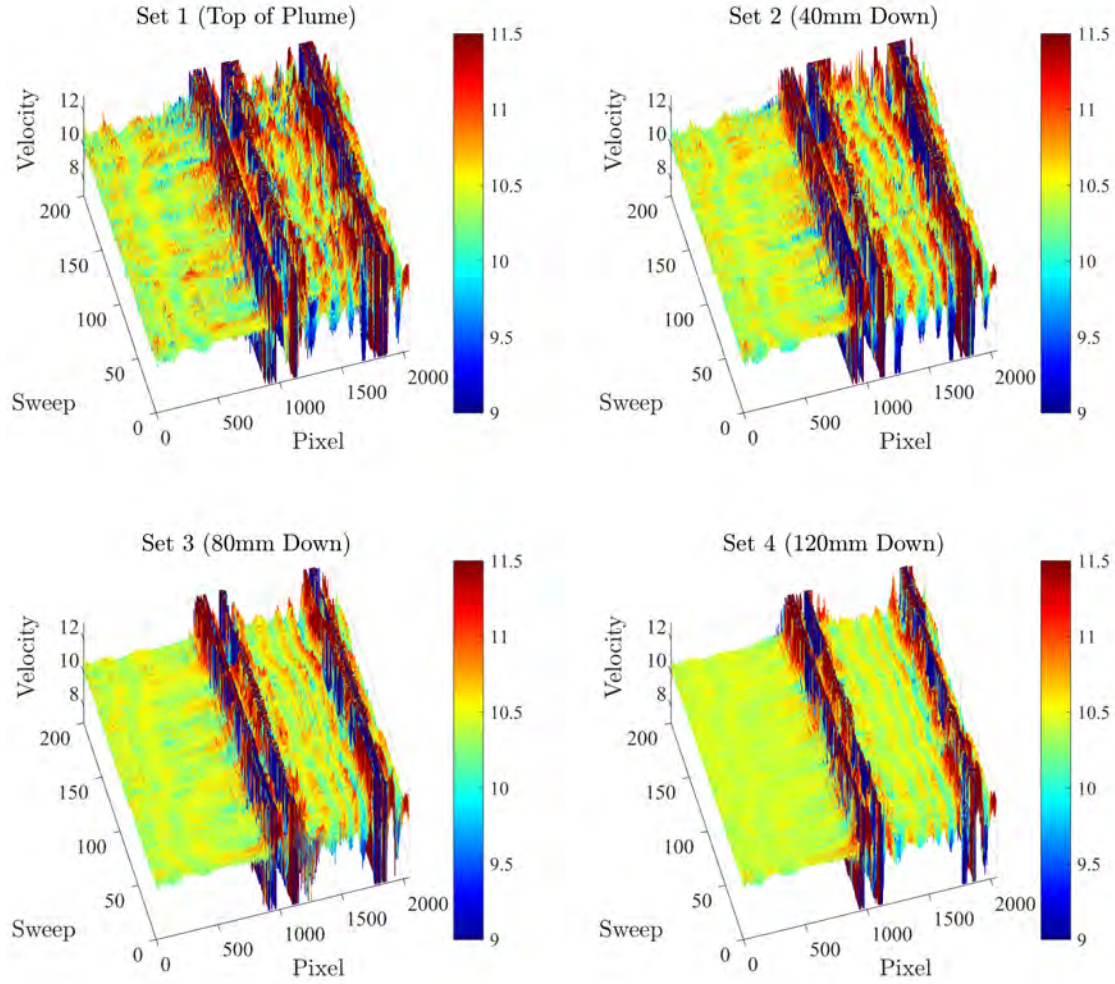


Figure 46. Average velocity at each sweep weighted by relative absorbance, for each pixel, for data sets 1 through 4, from the top of the plume down 120mm.

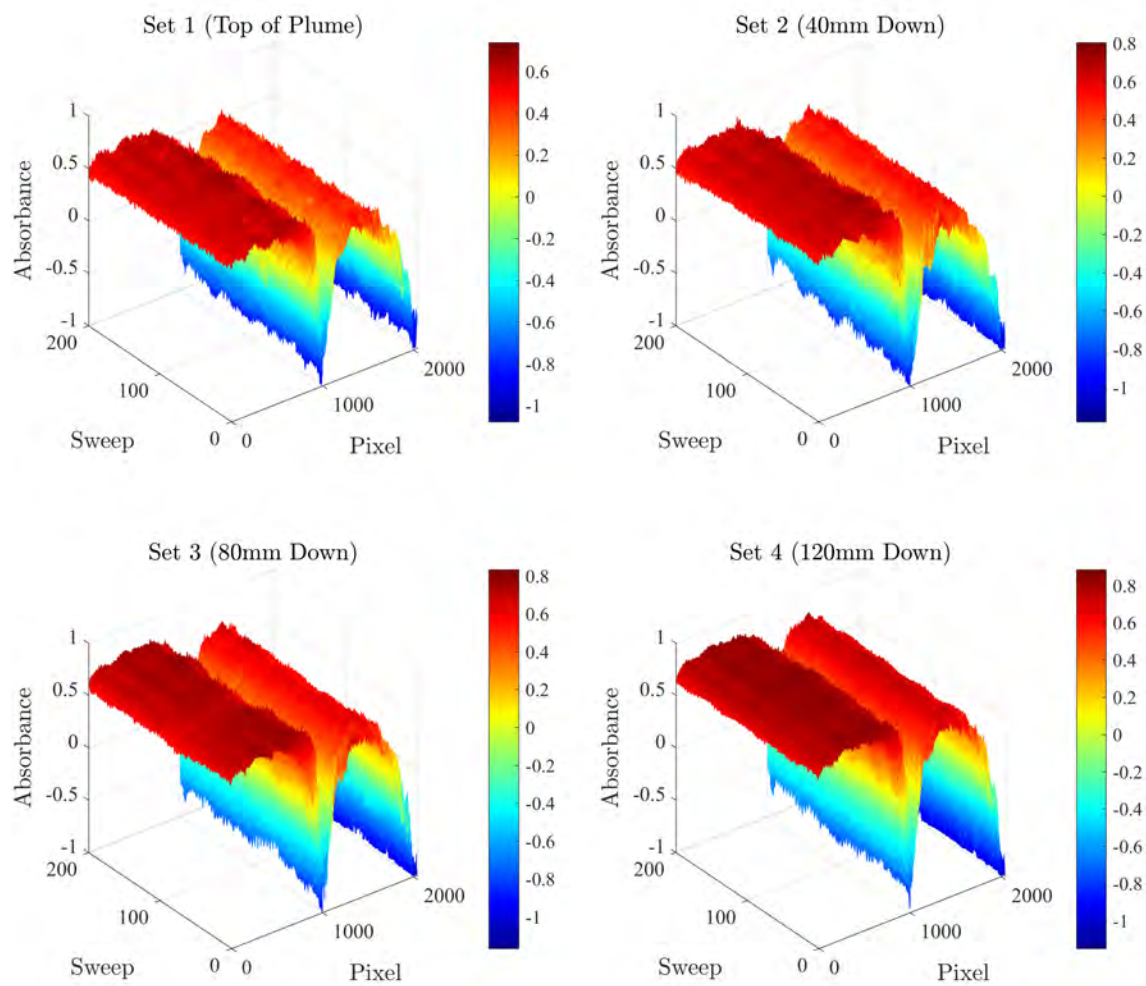


Figure 47. Average relative absorbance at each sweep, for each pixel, for data sets 1 through 4, from the top of the plume down 120mm.

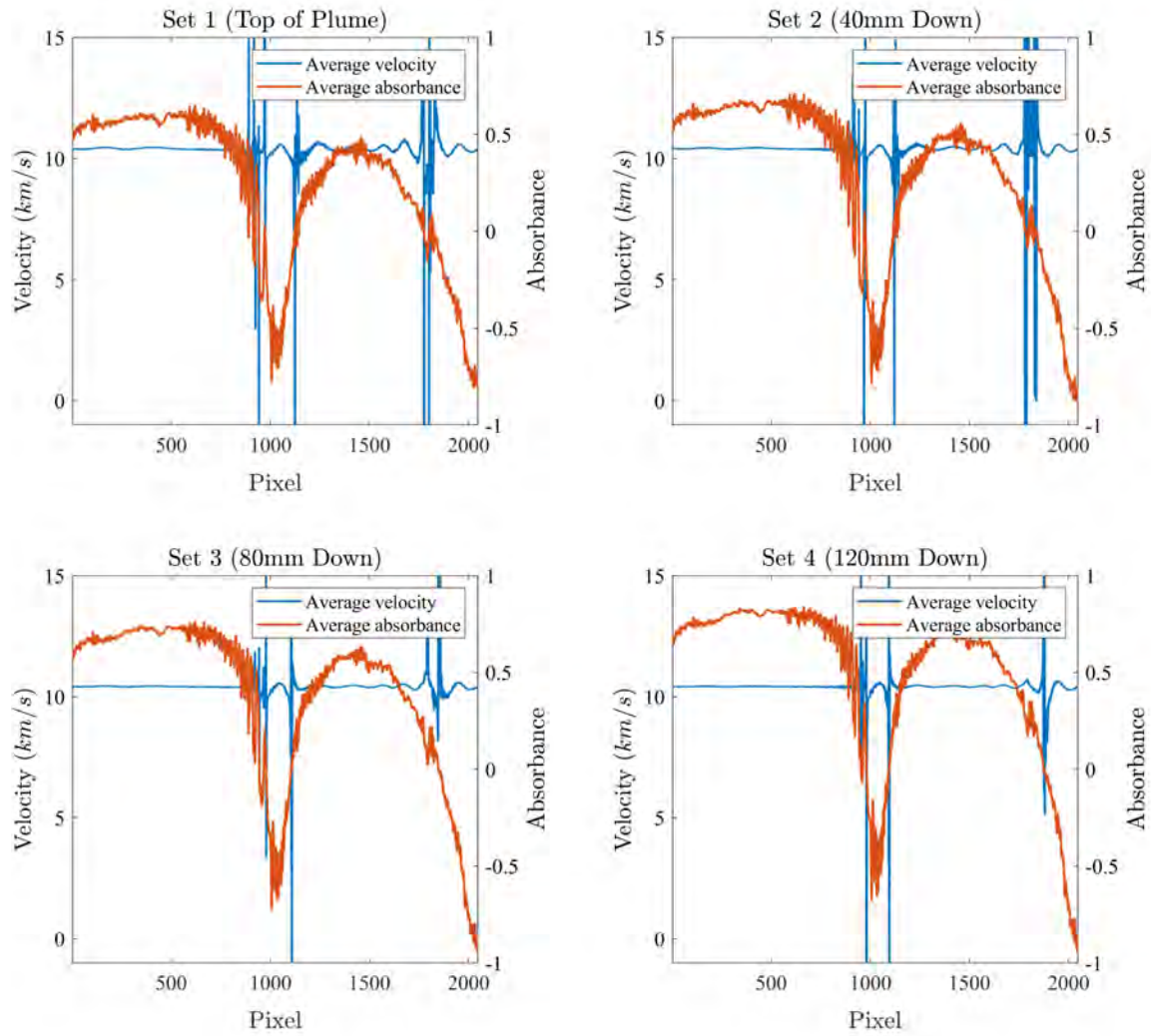


Figure 48. Average relative absorbance and average velocity, for each pixel, for data sets 1 through 4, from the top of the plume down 120mm.

Clearly, the high intensity in the center has negatively affected the results. Towards the center, velocity discontinuities are visible in Figure 46, a result of the negative absorbances which are visible in Figure 47. These discontinuities arise when the weighted average is taken and the absorbances have both positive and negative values, resulting in a sum close to zero. Negative absorbances indicate that the observed intensity was larger than the reference intensity. This may be due to the laser improperly warming up before the reference intensity was taken; perhaps the laser actually required a longer warm-up period. The laser was warmed up for one hour before the reference data was taken, and was kept on while the thruster was ignited and prepared for operation, with a total time of three hours between the collection of the reference and the first data point. The negative absorbance close to pixel 2000 was expected; this is a region which had little to no intensity from the laser during alignment and focusing. A similar region was expected on the opposite side, in the first few pixels, but is not visible. The trends are better observed by averaging the sweeps, as was performed to generate Figure 48. Here, the effects of the high intensity in the line center are visible. Further, increasing absorbance is observed as the laser sheet surveys regions closer to the center of the plume. Comparing the curves directly, averaged absorbance increases closer to the plume centerline, visualized in Figure 49. Averaged velocities, however, trend toward the same velocity in regions away from those affected by negative absorbances. The trend toward 10.5 km/s indicates even distributions of velocity in the visible range from 0-20.9 km/s.

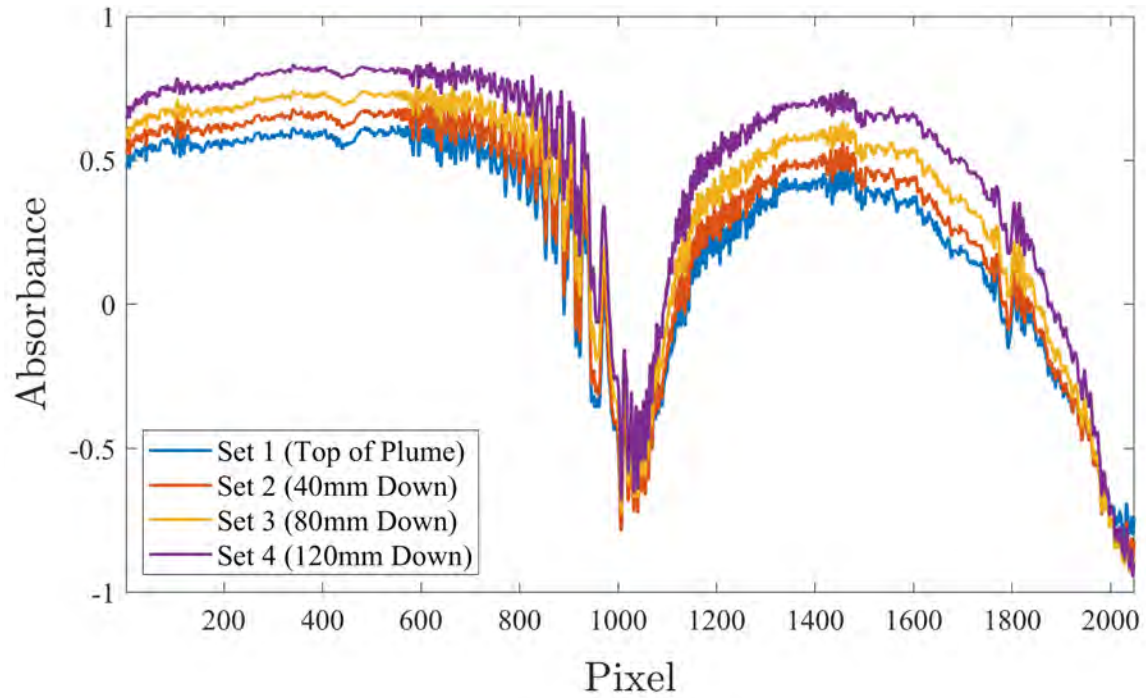


Figure 49. Average relative absorbance at each pixel, for data sets 1 through 4, from the top of the plume down 120mm.

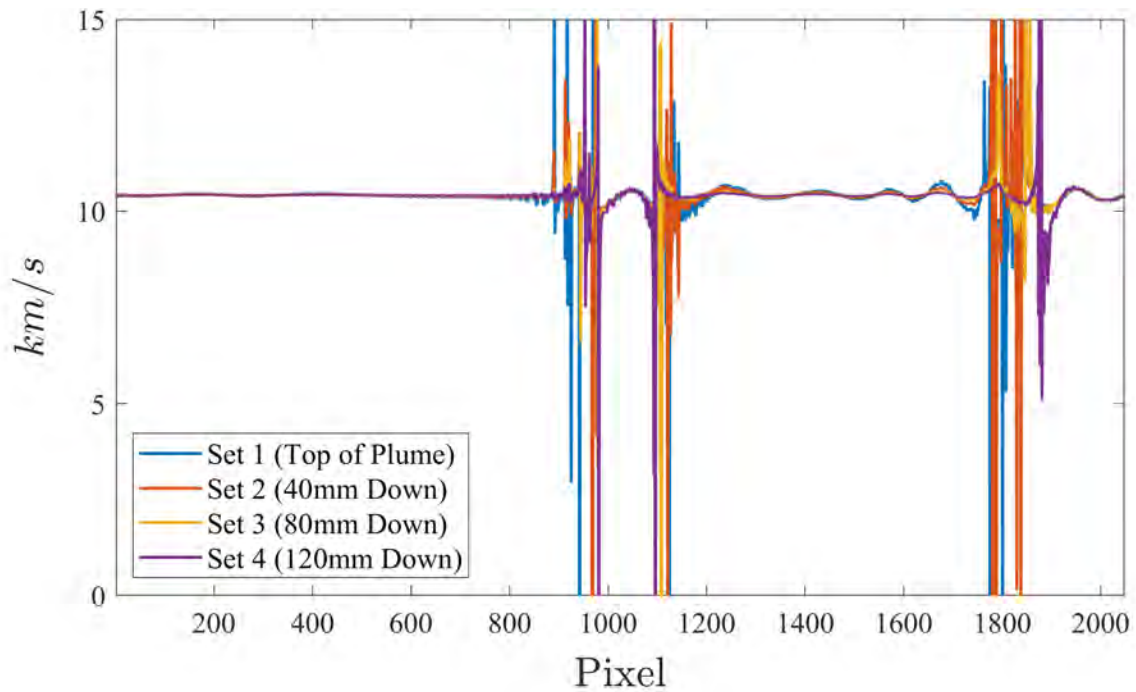


Figure 50. Average velocity at each pixel, for data sets 1 through 4, from the top of the plume down 120mm.

The standard deviation for these velocity averages was calculated and is presented in Figure 51. The values presented were obtained from the averaged velocity obtained at each of the 199 recorded sweeps. Large deviations are observed towards the center and right-hand side where velocity discontinuities were observed. While the presented standard deviations are limited to up to 1 km/s , those observed in regions of velocity discontinuity reach values as high as 3,561 km/s for Set 1, 811 km/s for Set 2, 4,270 km/s for Set 3, and 867 km/s for Set 4.

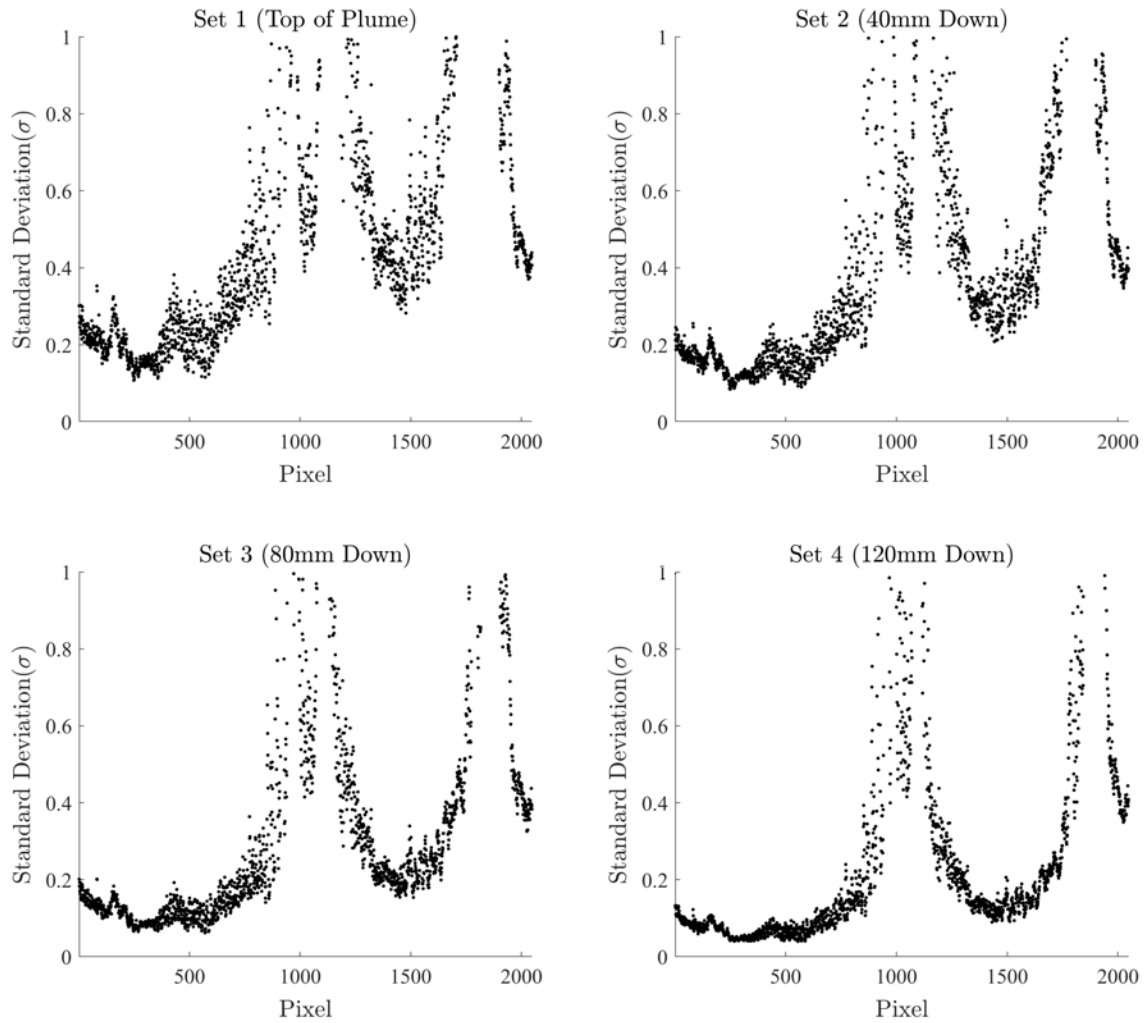


Figure 51. Standard deviation in km/s of weighted average velocity at each pixel, for data sets 1 through 4, from the top of the plume down 120mm. Standard deviations above 1 km/s are not displayed; values of up to 4,270 km/s were observed.

To further examine this trend, the velocity bins were averaged across all sweeps to determine an average velocity distribution at certain points of interest in all data sets. These are presented in Figure 52, and reveal that the ion velocities have little variation at each location. Error bars indicate the 68% confidence interval for each value. Similar to the methods used to determine the standard deviation of the weighted average velocity at each pixel, the standard deviation was also determined for the absorbance. This is presented in Figure 53 at each pixel for each set of data.

Velocity concentrations were expected to arise around 15 km/s, however this was not observed, and rather a plateau of absorbance was observed in the range from 0 to 20 km/s. The exact cause is unknown, but is likely a result of the methods used to determine the velocity. As discussed earlier, the velocities were determined by calculating the Doppler-shifted velocity based on absorbance, which resulted in the particle velocity along the laser axis. It was then assumed that particles exited mostly axially, so simple trigonometry was used to calculate the axial velocity from the measured component of velocity, using the incident angle between the laser sheet and the plume centerline. However, in reality, these Xenon ions exit the thruster with varying velocities and angles. Not all are accelerated by the entire electric field; these variations combine to produce measured axial velocities anywhere from -15.4 to 50 km/s at the extremes. Many of the lower velocities measured were likely a combination of particles with lower exit velocities and exit angles closer to perpendicular with the laser sheet, resulting in lower laser-axis velocity components.

The thruster was translated in 40 mm increments because the physical visible region was determined to have a width of approximately 40 mm. As such, the edges of this region were determined as approximately pixels 340 through 1730. To this extent, the absorbance measured around pixel 340 in Set 1 should approximately be the same as absorbance measured around pixel 1730 in Set 2, and so forth, as

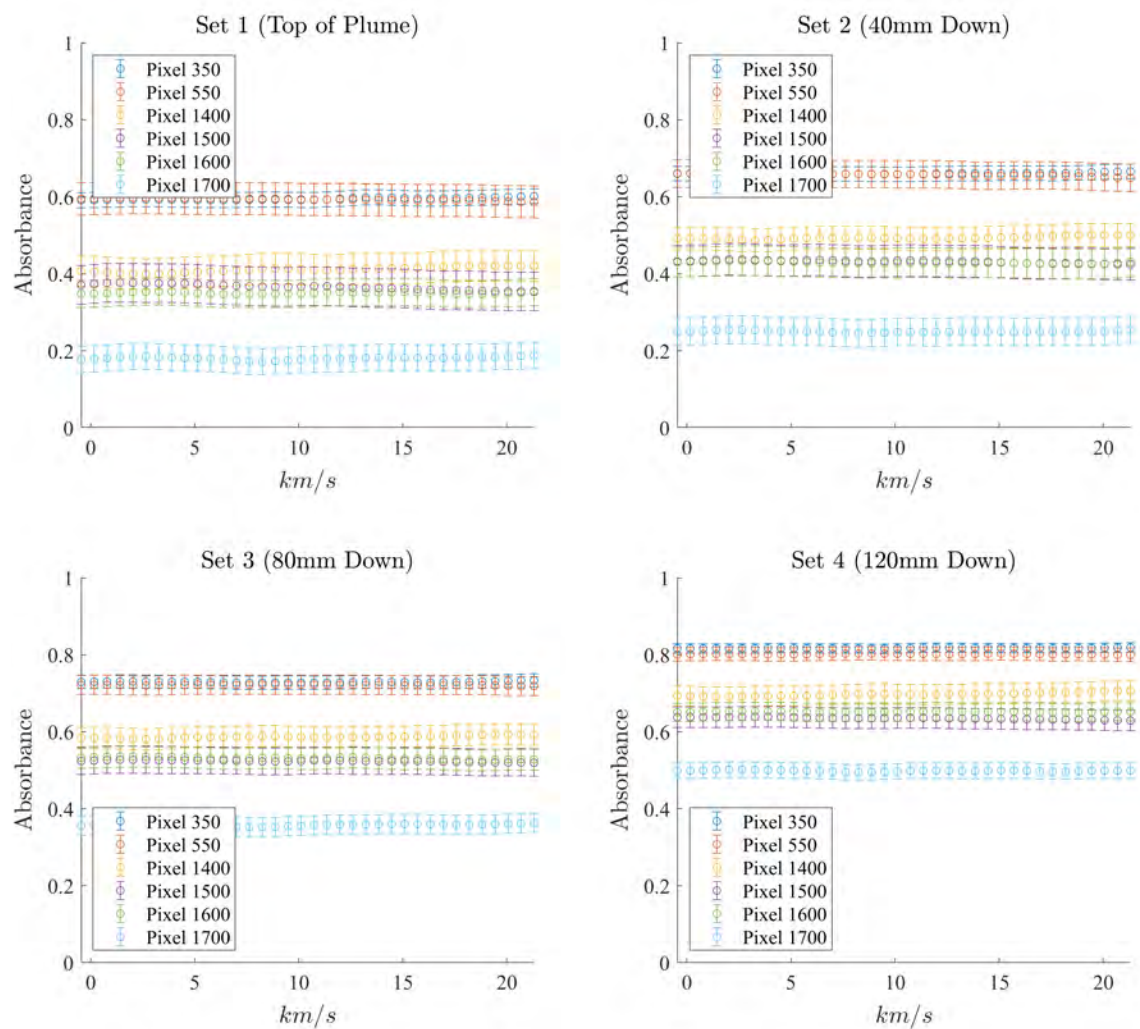


Figure 52. Time-averaged absorbance at each velocity bin, selected pixels, for data sets 1 through 4, from the top of the plume down 120mm.

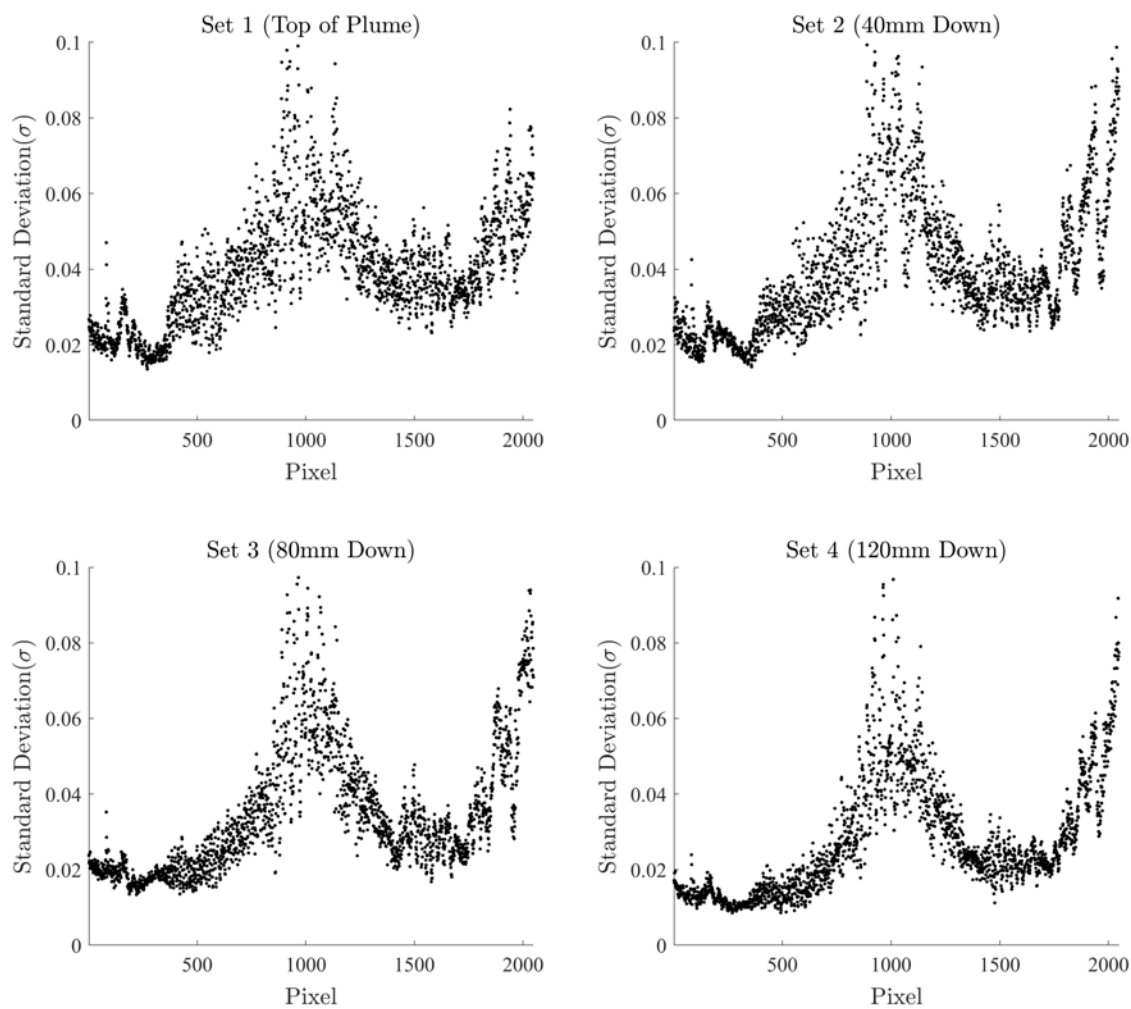


Figure 53. Standard deviation of relative absorbance at each pixel, for data sets 1 through 4, from the top of the plume down 120mm.

these represent the same physical location. However, this is not observed. Indeed, as absorbance at pixels 1500 and 1600 appears similar, the absorbance at pixel 1700 should not significantly diverge from that measured at pixels 1500 and 1600, but is measured significantly lower across all sets of data. The same distribution is observed across all sets. This indicates a spatially dependent phenomena, which appears to be caused by the equipment. The exact cause is unknown, but may be related to the issues observed towards the center, where the effects of the increasing laser intensity with time were most prevalent. As the laser warmed up further, its profile may have changed from the reference that was taken and presented in Figure 38; the wings may have come to a steady state condition with the same intensity. This could result in the observed differences in absorption measurements at the same physical location in different data sets. If an extra photodiode was added to obtain a true reference intensity from the laser, it could be used to normalize the camera reference. When this normalized reference is later used with the actual intensity recorded during each set from this extra photodiode, a proper reference intensity would be generated, fully accounting for the laser's intensity variations.

V. Conclusions

This research obtained a vertical distribution of velocity throughout the plume of a Busek BHT-600 Hall thruster, providing proof of concept for a LAS system designed to obtain full-field, simultaneous Xenon ion number density and velocity profiles. While issues with the chamber maintaining pressure forced operation at lower mass flow rates than Bui used, the ions were accelerated across the same discharge voltages, and similar velocities were obtained.

5.1 Significance

This work laid the foundation for future research using this Spectral Energies LAS system. The full system will eventually allow full-field measurements of singly-ionized Xenon velocity and relative number density distributions. These measurements can then be used to determine the performance characteristics of the thruster, in a non-intrusive manner, eliminating error induced by taking measurements using intrusive probes.

Limitations for the system were discovered, including the usable visible width of a single sheet of approximately 40 mm. This indicates that when the full system is implemented, each sensor will pick up no more than 40 mm of the plume. When these line measurements are converted to 2D fields through methods such as filtered back-projections, the edges of each signal are lost, so the total area covered by a single scan would approximately be 35×35 mm or smaller. Further limitations of the system include its use for single-sheet analysis of the plume. This method, without other viewpoints for full velocity vector determination, cannot distinguish between particles moving with angles off the centerline. This resulted in a wide variety of axial velocities being reported, including some negative, as some ions had divergence

angles large enough in the direction of the laser source to Doppler-shift in the opposite direction.

To pick up the Doppler-shifted Xenon ions, the laser used a function on the PilotPZ 500 laser controller which allowed oscillation of the laser's wavelength around its center wavelength. At its maximum amplitude, and given the angle of the laser mounted with respect to the plume centerline, it was determined that the laser could hit a range of wavelengths corresponding to velocities from 0 to 100 km/s. This indicates that the system can also be used for Xenon-based ion thrusters with higher exit velocities than Hall thrusters. Adjustment of the laser sheet incidence angle with respect to the plume centerline could allow an even larger range of velocities.

Limitations of the vacuum chamber were also discovered; mass flow rates matching even the lowest mass flow rate used by Bui could not be maintained without increasing chamber pressure above the 5×10^{-5} torr limit for thruster operation. This forced operation at lower mass flow rates than Bui. This issue may have been a result of the air conditioning freezing up due to a strong snowstorm, leading to soaring temperatures in the lab of over 100° F the week of testing. It also may have been a result of increasing the chamber volume without increasing the pump capacity. If that is the case, it should be mitigated by the installation of two additional cryopumps on the new center chamber section, planned for the upcoming year.

5.2 Future Work

To reach this LAS system's full operational capabilities and the original goals intended for the system, the remaining five sets of optics must be installed and aligned such that each sheet passes through the exact same plane. This will require significant time and effort. The camera mounts will also require re-work to include rotational capabilities for each camera, as the other five laser sheets will not necessarily be

perpendicular to the ground but will be rotated somewhat. It may be worthwhile to look into different mounting hardware, as well, which would better allow locking the camera to a specific position and resist misalignment caused by the vibrations of the vacuum chamber in operation. Further, this will require improvements to the fiber pass-through. In its current state, only one of the seven pass-throughs was able to be freed; the others are seized and thus inoperable. This should be resolved with a custom Lesker pass-through, which is on order, and has hard attachment points for the fibers instead of damage-prone flexible fibers as is currently implemented.

Improvements to the thruster mounting setup could also be made, to attach the thruster on a sting allowing its positioning closer to the imaging plane. This would reduce the risk of impacting any of the LAS system hardware with the traverse, and reduce the total size of the plume which must be surveyed. In turn, this would also mean that none of the LAS system hardware would be subject to ion impacts and erosion from the plume. As currently set-up, it was observed that as the thruster reached its final position to capture the bottom-most portion of the plume, the majority of the plume was directly centered on the upper dovetail beam. This indicates that any LAS system hardware, including the two acylindrical lenses and laser collimator which would be attached there, would be subject to ion bombardment, surely limiting the lifetime of the optics either due to erosion or contamination of the lenses.

Equipment should also be acquired, which was borrowed for this research but essential to the operation of the system. This includes a fiber-coupled visible red laser that can be used for alignment purposes, and a wavemeter accurate to at least one thousandth of a nanometer. The wavemeter is essential for the purposes of determining the wavelengths between which the laser sweeps, and its center wavelength, and should be capable of displaying the wavelength with respect to time in order to properly determine the maximum and minimum of the sweep.

Additional equipment should be acquired which was not available for use for these experiments, to provide a reliable reference intensity for use in absorption calculations. The initial plan of using the photodiode behind the hollow cathode lamp would not work, as this photodiode only records the laser intensity after it has been affected by the glass of the lamp as well as the energized Xenon within. Rather, a separate photodiode is needed, which would directly receive a component of the beam. This would allow a direct mapping of the laser's intensity as it drifts in time with the reference intensity for each camera, or a normalization of the reference intensity with respect to the raw photodiode intensity. Then, during experimentation, this mapping would allow determination of a proper reference for each camera, based on the recorded intensity of this separate photodiode during the experiment. While a seventh camera with a seventh set of lenses and mirrors could provide a similar reference, the beam incident on the camera may not be the same as the other six. Due to a variety of factors, from imperfections in the assorted lenses and mirrors, to the actual alignment and focus of the beam, this method would not provide an exact match for reference as well as pre-recording normalized references for each of the six cameras. The best option is thus simply installing a separate photodiode, which can be used to normalize the reference for each camera, and can be attached to the empty third DAQ input.

With this extra photodiode installed to determine proper reference intensities, single-sheet analysis could again be performed to validate this as a potential solution to the observed issues from this research. Future research could also investigate the observed absorbance at negative reported axial velocities; the cause of which was not definitively determined from this research. It should also check for hysteresis in the system to see if repeated consecutive data collection affected the results, which was not examined in this research.

Improvements to resolution could be made through a number of methods. Currently, the laser scans at 500 Hz. This is adjustable up to 1 kHz if faster data collection is desired; however, at the maximum sampling rate for the cameras (80 kHz), this would cause reduced resolution. Slowing the scan speed can increase the number of samples which the cameras take during the sweep, thus decreasing the “size” of the velocity bins and improving resolution. Further, adjusting the range of wavelengths between which the laser scans could improve the resolution by scanning a smaller range. Currently, the scan range exceeds the desired range of 0-20 km/s up to 36.85 km/s and down to -12.28 km/s. Decreasing that range would further decrease the “size” of each velocity bin by having fewer bins exceed the desired range.

Other than the hardware changes which need implementation, software must be developed to handle three-component velocity measurements. Before their use with existing and widely available MATLAB® toolboxes such as *iradon* to perform a filtered back-projection, the scalar measurements must be split into their components of velocity. This cannot be accomplished simply through the use of unit vectors describing the orientation of each laser sheet, as the resulting matrix is rank-deficient. Each measurement must also account for the fact that the plume is not contained within the survey region, but that each laser sheet passes through significantly larger portions of the plume. Engineers at Spectral Energies are working on a solution, but are far from its implementation. Implementation of this method will resolve the issues described earlier with extreme ranges in axial velocity determination caused by a single sheet’s inability to distinguish the divergence angles of the particles in the sheet, by solving for unique three-component velocities at each pixel.

Appendix A. Thruster Operation Guide

This appendix contains a general guide for operating the vacuum chamber and Busek BHT-600 Hall thruster paired with a Busek BHC-1500 hollow cathode. For other thrusters and cathodes, some exact steps may have different numbers (e.g. discharge voltages), but much of the general information will be valid.

1. Install thruster and cathode

- (a) Cathode should have wires labeled, from left to right, C H K
 - i. C = cathode body, connect to ground
 - ii. H = heater, connect to heater +
 - iii. K = keeper, connect to keeper +
- (b) Thruster should have magnets, inner and outer, + and -
 - i. You should wire these in series, so inner – to outer + or vice versa. Doesn't matter what order, as long as they're in series, and you connect them out properly
- (c) Both need propellant line connections. Put a new VCR washer on the lines (each can only be used once, throw out when you disconnect)
 - i. As of this writing, mass flow controller 2 is for the cathode (10 SCCM max) and mass flow controller 4 is for the anode (50 SCCM max).
 - A. Mass flow controller 2 is hooked up to readout slot 3 due to an issue with slot 2 on the readout. Mass flow controller 4 is hooked up to readout slot 4

2. Make sure exterior connections match up to interior, and record what goes where (especially useful if you're testing at the same time as someone else in the chamber)

3. Pump the chamber down

- (a) Once everything is installed inside the chamber, and you know what electrical connections are what, close both doors (CHECK FOR WIRES HANGING OUT THE DOORS FIRST) and tighten. Then, press the big green “start” button on the panel on the wall.
- (b) You can monitor this using the Lesker 300-series pressure gauge, to the right of the computer, on the chamber.
- (c) The cryopumps will turn on once the chamber pressure hits about 70 millitorr, measured by the Lesker gauge. Below the millitorr level, you’ll need to use the residual gas analyzer to get the pressure. This software is on the lab computer.
- (d) This process takes 16-24 hours depending on ambient conditions and whether you’ve added new things into the chamber which may outgas somewhat.

4. Purge the propellant lines

- (a) This should be done every time you expose the lines to air, or change out propellant canisters. Flow about 2 SCCM through the cathode and anode lines for 30 minutes.

5. Condition the cathode

- (a) Once the propellant lines are purged (AFTER, not at the same time) start conditioning the cathode by turning the heater to 2.0A. Leave it heating at 2.0A for 90 minutes.
- (b) After 90 minutes, turn the heater to 4.0A. Leave it for another 90 minutes.
- (c) After 90 minutes, turn the heater to 6.0A. Leave it for 30 minutes.

- (d) After 30 minutes, the cathode is conditioned. This process only needs to be performed after the cathode is exposed to air. So, if you're testing, stop testing, and leave the chamber at vacuum, you can heat up the cathode faster (same amperage steps, but only 20 minutes at each step)

6. Light the cathode

- (a) The heater should be at 6.0A. Start a mass flow of Xenon or Krypton through the cathode at approximately 1-2 SCCM.
- (b) Increase the heater to 6.5A and leave it for 5 minutes.
- (c) After 5 minutes, set the keeper to 0.5A (if you're still using the BPU, the voltage should be set; if you're using an external supply, the keeper voltage should be 600V or so.)
- (d) The cathode should immediately light, which is noticeable by a keeper voltage drop from 600V to about 40V and a current of about 0.2A
- (e) If the cathode does not light, try a few things:
 - i. Increase the mass flow rate through the cathode, max 5 SCCM
 - ii. Increase the heater in increments of 0.5 A to a maximum of 7.5 A (do not exceed)
 - iii. Don't just jump right to 5 SCCM and 7.5 A, go in steps, and at each step let the cathode sit for about 10 minutes.
- (f) If the cathode still has not lit after this, it's probably been poisoned, or you have a grounding issue. Check with a voltmeter to make sure that you're getting the voltage you think you should see (around 600 V) across the keeper and cathode body terminals.
- (g) When the cathode lights turn the heater down to 4.0A. This can be adjusted to more current if the thruster goes out, but 4.0A should be enough

to keep it going until the thruster is lit.

7. Start flow to the anode at approximately 20 percent
8. Turn on the thruster discharge
 - (a) Don't turn on the magnets yet. Start the thruster discharge at 50-100 V. This should stabilize into the current limit mode, which will max out the PPU current at 3.2A or so and whatever voltage is necessary to maintain that.
 - (b) Once lit, turn the cathode heater down to 2.0A.
 - (c) Turn the magnets on low (0.25A). Slowly bump up the magnets in 0.25A increments until the voltage limit mode is reached (should max out at the set voltage, and some discharge current to meet that voltage)
 - (d) If stable, keep upping the magnet current and discharge voltage until the desired operational discharge voltage is reached.
 - i. If not stable or the discharge goes out, turn off the magnets or turn down the current to the magnets. Wait for a bit at the lower magnet current for the thruster to stabilize, 10 minutes. Increase the mass flow slightly.
 - (e) Once the desired discharge voltage is obtained, adjust the anode and cathode mass flow rates to the desired testing mass flow rates. The cathode heater can be turned off at this point, it will maintain temperature through operation.
 - (f) The thruster will take 1-2 hours to reach equilibrium. Wait at least one hour at nominal operating conditions before taking data.
9. Take measurements

10. When finished with experimentation, turn off the thruster and cathode
 - (a) Turn off the magnets first
 - (b) Then the discharge
 - (c) Then the cathode keeper (and the cathode heater if you still had that on)
 - (d) Then turn off the mass flow controllers and mass flow controller readout
 - (e) Close all valves
11. Before venting vacuum chamber to atmosphere, wait 45 minutes for the cathode and thruster to cool
12. Vent chamber by opening nitrogen valve on the side of the chamber near the propellant line input.
 - (a) Make sure to take off all door clamps, leaving only one closed but loosened to keep the door from swinging wide open in an uncontrolled manner.
 - (b) When the chamber pops open, IMMEDIATELY remove the cathode and place into the dry box. The thruster should also go in the dry box, but is less susceptible to damage from the air. DO NOT LEAVE CATHODES OUT IN THE AMBIENT AIR.
 - (c) Also turn off the nitrogen when the chamber opens.
 - (d) This takes about 1.5 to 2 hours to open.

Troubleshooting.

1. Often grounding issues cause things to not work. Check that what you think is grounded is actually grounded. Check that things which should not be grounded are not shorting out somewhere. There's a lot of wires between the switchboard outside the chamber and the board inside the chamber.

2. Use a voltmeter to check that the keeper, discharge, etc are actually at the proper voltages. There may be a power supply issue.
3. If the chamber is not pumping down, check the nitrogen tank. One of the gate valves requires that system be pressurized to operate. If the building nitrogen system loses pressure, it won't work!

Appendix B. LAS System Operation Guide

This appendix contains a general guide for operating the Spectral Energies LAS system as it is installed at AFIT's SPASS lab vacuum chamber.

1. Before operation, turn on the laser safety warning lights and ensure all personnel in the lab have the proper eye protection.
2. Turn on both the PilotPC 4000 (labeled "MASTER LASER") and PilotPC 500 (labeled "TAPERED AMP") by pressing the power buttons. When off, this button has a LED which glows red.
3. Verify that the settings for the PilotPC 4000 (laser amplifier controller) and PilotPZ 500 (master laser controller) are as desired.
 - (a) The main setting on the PilotPC 4000 is the laser current. Do not start with the laser at a high current setting. Ideally, set the current to approximately 1,000 mA before turning on the amplifier.
 - (b) The main settings on the PilotPZ 500 are the scan settings.
 - i. "Piezo" controls whether the scanning function is enabled or disabled.
 - ii. "Offset" controls the offset of the center frequency. This helps if you set the center frequency of the laser manually using the tuning screw to a specific wavelength, and then want to offset your scan slightly above or below that without physically changing the tuned frequency.
 - iii. "F-Generator" allows selection of the desired waveform. Typically, a triangle wave is desired.
 - iv. "Frequency" chooses the scan rate.

- v. “F-Amplitude” chooses the amplitude of the scan. Adjusting this changes the wavelengths between which the laser scans. These wavelengths must be determined externally using a wavemeter.
- 4. Turn on the PilotPZ 500 by pressing the “ON/OFF” button above the label “Laser”. The button will now glow red to indicate that the laser is on.
- 5. Turn on the PilotPC 4000 by pressing the “ON/OFF” button above the label “Laser”. The button will now glow red to indicate that the laser is on.
- 6. The laser requires >1 hour to warm up and reach steady state conditions. The exact time required is unknown. In my research, 1 hour was not enough- by the time actual data was taken (3 hours later) the laser’s intensity had significantly increased from the 1 hour point.
- 7. Turn on the Spectral Energies computer in the same rack as the laser controllers.
- 8. Open the Spectral Energies LabVIEW® code.
- 9. With the laser on, make sure you can see data coming from the DAQ. The photodiode behind the hollow cathode lamp (also mounted in the rack) connects to a DAQ which then connects to the computer. Adjust the laser’s power, or the collimator directing the laser through the hollow cathode lamp, so that the voltages being read in are approximately 3-4 V.
- 10. When ready to take data...
 - (a) Make sure the “write” button has been clicked and choose a path where you want the data saved.
 - (b) Turn on the hollow cathode lamp power supply and adjust the current to 5 mA.

11. Click “run” to take data. The button will change to show “stop”. Do not click the button. It will change back to “run” when it has written all of the data to disk.
12. If taking more data, but it will be some time before the data is taken, turn off the hollow cathode lamp to preserve its lifetime. Remember to turn it back on before taking more data.
13. Based on the name chosen, each additional data file will be saved in a directory named “chosen name” + 1, 2, 3... etc based on the existing directories already present in the write directory. So, you do not need to re-name between runs.
14. When finished collecting data, turn off the hollow cathode lamp and lasers. Before turning off the lasers, you may want to re-check the laser’s center wavelength and the range of the scan using the wavemeter.

Alignment.

There are various components of the laser which can become misaligned. This includes the two mirrors located inside the laser housing, which direct the beam in a u-shape into the fiber coupler. This is the component sticking out of the laser housing to which the fibers attach. The fiber coupler also can become misaligned.

From here, the other points of misalignment exist in the various sets of optics inside the chamber and the cameras external to the chamber. Most small misalignments come from bumping the cameras and can be fixed externally. However, there are points of adjustment on the laser collimator, the first and second acylindrical lenses, and the mirrors.

The mirrors mostly adjust the positioning of the laser and have little effect on the shape of the beam. The lenses and collimator can affect the shape. If you notice that

the beam has a slightly curved shape (i.e. when aligning with the cameras, you can only get saturation on the edges and not the center or only the center), check that...

1. The last mirror is directing the sheet into the final aspherical focusing lens perpendicularly.
2. The second acylindrical lens has not been bumped.

The adjustment of how the beam hits the aspherical lens, and the alignment of the second acylindrical lens seem to have the most effect on the shape of the beam. That being said, they are not the only sources of error and if adjusting these does not fix your issue, check into the first acylindrical lens and the laser collimator.

Alignment is easiest when performed first with a visible (red) laser to coarse-align the entire system, then switching to the Sacher Lasertechnik IR laser for fine alignment. Coarse alignment usually aligns the system almost perfectly; the only adjustments that tend to be required with the IR laser are to the camera positioning.

It is recommended to check that all sheets are properly aligned onto their respective cameras before taking any data.

Appendix C. MATLAB Code

3.1 multiprocessingdata.m

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MULTIPROCESSDATA.M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % This handles processing for the raw data files and saves out .mat files
6  % with the data.
7  %
8  % This version will take in as many sets of data as you tell it and save
9  % them out numerically (data1.mat, data2.mat, etc) which is useful when
10 % you've got a bunch of data sets (ex. to capture the entire plume)
11 %
12 % It'll pull in reference data all at once with the actual data
13 %
14 % To only pull in new data (after ref is processed, for example) change
15 % readInRef to true
16 %
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18 clc; clear; close all;
19
20 %% SETTINGS
21
22 % How many data sets?
23 numdata = 6;
24
25 % Choose which pixels to use
26 pixmin = 1;
27 pixmax = 2048;
28
29 readInRef = false; % SET FALSE TO PROCESS REFERENCE ALONG WITH NORMAL DATA
30
31 % Set data file names based on number of scans we're taking in
32 dfnames = [];
33 for i = 1:numdata
34     dfname{i} = sprintf('data%s.mat', num2str(i));
35 end
36
```

```

37 %% IMPORT
38
39 % Choose files to read in
40 if ~readInRef
41     reffnames = selectfiles('ref');
42     otherreffnames = selectfiles('refother');
43 end
44 for i = 1:numel(dfname)
45     fprintf('Choose data frames and other frames for set %s\n', num2str(i))
46     datafnames{i} = selectfiles();
47     otherfnames{i} = selectfiles('other');
48 end
49
50 % Fix reference intensity file for reading
51 if ~readInRef
52     fprintf('Converting reference files\n')
53     reffiles = filefixer('ref',reffnames);
54 end
55
56 % Read in reference data file
57 if ~readInRef; refdata = importdata(reffiles ,pixmin ,pixmax); end
58
59 % Read in the "other" reference files too
60 if ~readInRef
61     fprintf('Processing reference DAQ files\n')
62     [refabscell ,refttlsig] = importtoother(otherreffnames);
63 end
64
65 % Save processed data to MAT files
66 if ~readInRef; save('ref.mat','refdata','refabscell','refttlsig'); end
67
68 % Now do all that for all the data files
69 for i = 1:numel(dfname)
70     % Fix data file for reading by tdfread()
71     fprintf('Converting data files for set %s\n', num2str(i))
72     datafiles = filefixer('',datafnames{i});
73
74     % Read in data file
75     data = importdata(datafiles ,pixmin ,pixmax);
76

```

```

77     % Get other data
78     [abscell, ttlsig] = importother(otherfnames{i});
79
80     % Save processed data to MAT files
81     save(dfname{i}, 'abscell', 'ttlsig', 'data');
82     clear abscell ttlsig data
83 end
84 fprintf('Processing complete.\n\n')

```

3.2 analyzedata.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ANALYZEDATA.M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % This actually messes with data.
6  % run "multiprocessdata.m" to save data to .MAT files from the raw data.
7  % This reads in those .MAT files then processes.
8  % Various flags can choose the type of processing. v5 was used to obtain
9  % the data presented.
10 % The filtering can take a while, depending on the size of data. Expect
11 % each case (~6 data sets) to run overnight.
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 clc; clear; close all;
14
15 %% SETTINGS
16 % How many scans were taken to encompass the full plume?
17 % This assumes that the first is the top scan.
18 % Note that left to right is bottom to top (leftmost is the bottommost)
19 scans = 4;
20
21 % Choose methods of calculating absorbance
22 % v1: uses DAQ as I0, -ln(I/I0)
23 % v2: uses avg. ref as I0, -ln(I/I0)
24 % v3: uses avg. ref as I0, 1 - I/I0
25 % v4: takes full ref as I0
26 % v5: takes full ref as I0, both filtered with bandstop and lowpass
27 % v6: same as v5 but filtered using filtfilt() to do zero-phase filter
28
29 v1 = false;

```

```

30 v2 = false;
31 v3 = false;
32 v4 = false;
33 v5 = true;
34 v6 = true;
35
36 % Number of points in one frame
37 % IF CHANGED, SET THESE
38 framesize = 160;
39 totalframesize = 160; %there are actually 160 points, but 5 are dropped
40 totalothersize = 1000;
41
42 % Calculate other data points per camera data point
43 pointRatio = totalothersize/totalframesize;
44
45 % Points from the "other" data that marks the start and end of the usable
46 % laser sweep
47 % SET THESE
48 %two lines below from testing 14 JAN 2019
49 point1 = 284;
50 point2 = 283; %note that this is actually point 500+value of the overall
51
52 % 1000-point sweep, but since they're divided into 500 points based on the
53 % TTL pulse data, it's given as the actual point number
54 totalsize = 500;
55 camerasweep = totalsize/totalothersize*totalframesize;
56
57 % Set the max wavelength (of the scan, nm)
58 % SET THESE
59 % These values are for the sample data
60 % maxwl = 834.74593;
61 % minwl = 834.66326;
62 % These values are from testing 14 JAN 2019
63 maxwl = 834.754;
64 minwl = 834.714;
65
66 %distribution based on the 500-pt vals
67 wldist1 = linspace(maxwl,minwl,totalsize);
68 %distribution based on the 80-pt vals
69 wldist2 = linspace(maxwl,minwl,camerasweep);

```

```

70
71 % Feature wavelengths
72 XeI = 834.68217;
73 XeII = 834.724;
74
75 % View plane offset angle.
76 viewangle = 17;
77
78 % Calculate velocities for each bin (using wldist2)
79 c = 299792458/1000; %km/s
80 veldist_laseraxis = (wldist2 - XeII)./XeII.*c;
81 veldist = veldist_laseraxis./cosd(90-viewangle);
82
83 % Same but calc using frequencies
84 c = 299792458; %m/s
85 maxf = c/(maxwl*1e-9); %wl in nm, convert to m, f in Hz
86 minf = c/(minwl*1e-9);
87 fdist = linspace(minf,maxf,camerasweep);
88 XeII_f = c/(XeII*1e-9);
89
90 c = 299792458/1000; %km/s
91 veldist_laseraxis_f = -(fdist - XeII_f)./XeII_f.*c;
92 veldist_f = veldist_laseraxis_f./cosd(90-viewangle);
93
94
95 % Affects peak finding for stationary reference signal
96 peakthreshold = 0.02;
97
98 % Pixels visible, determined on 13 Jan 2019 calibration
99 lowvis = 346;
100 highvis = 1623;
101
102 % DC offset (used to make the camera data non-negative)
103 offset = 190;
104
105 % Scale factor (used to scale the camera data to the voltage I0)
106 scalefactor = (1/256)*(1/200);
107
108 % FILTERING
109 % Settings

```

```

110 fs = 80000;
111 lowfilter = 1000;
112 highfilter = 12500;
113 bandstop1 = 0.01;
114 bandstop2 = 500;
115 % Set up filters for v6 calcs
116 % These will need to be updated if the desired frequencies change
117 % They were obtained using bandstop() to stop between 0.01 and 500 Hz and
118 % lowpass() to filter out above 12500 Hz
119 % Bandstop
120 bsfilter = designfilt('bandstopfir', 'PassbandFrequency1', 0.01, ...
121     'StopbandFrequency1', 39.2592, 'StopbandFrequency2', 460.7508, ...
122     'PassbandFrequency2', 500, 'PassbandRipple1', 0.1, ...
123     'StopbandAttenuation', 60, 'PassbandRipple2', 1, ...
124     'SampleRate', 80000, 'DesignMethod', 'kaiserwin', 'MinOrder', 'Even');
125 % Lowpass
126 lpfilter = designfilt('lowpassfir', 'PassbandFrequency', 12500, ...
127     'StopbandFrequency', 16817.5, 'PassbandRipple', 0.1, ...
128     'StopbandAttenuation', 60, 'SampleRate', 80000, ...
129     'DesignMethod', 'kaiserwin', 'MinOrder', 'Even');
130
131
132 %% DATA MANIPULATION
133 load ref.mat
134 for scannum = 1:scans
135     datafile = sprintf('data%s.mat', num2str(scannum));
136     dataarr{scannum} = load(datafile);
137 end
138
139 % Refdata
140 % This is taken from the cameras, and used to account for the beam path
141 % when no Xenon is flowing (the mirrors, optics, window, etc)
142 %
143 % This is subtracted from the actual camera data before comparing to I0
144 %
145 % The reference data is split into the different laser scans and averaged,
146 % then rebuilt to the proper size for any real input data from the cameras
147
148 [rrow, rcol] = size(refdata);
149 numRefSweep = round(rrow/framesize);

```

```

150 avgRef = zeros(framesize,rcol);
151
152 for sweep = 1:numRefSweep % Separate into the individual sweeps
153     refCell{sweep} = refdata((framesize*(sweep-1)+1):(framesize*sweep),:);
154     avgRef = avgRef + refCell{sweep};
155 end
156 avgRef = avgRef./numRefSweep;
157
158 % Filter the reference, if v5 or v6
159 if v5
160     % filter things, normal bandstop and lowpass
161     filteredRefDataV5 = zeros(rrow,rcol);
162     for i = 1:rcol
163         filteredRefDataV5(:,i) = bandstop(refdata(:,i), ...
164             [bandstop1 bandstop2], fs);
165         filteredRefDataV5(:,i) = lowpass(filteredRefDataV5(:,i), ...
166             highfilter, fs);
167     end
168 end
169 if v6
170     % filter things, zero-phase bandstop and lowpass
171     filteredRefDataV6 = zeros(rrow,rcol);
172     for i = 1:rcol
173         filteredRefDataV6(:,i) = filtfilt(bsfilter, refdata(:,i));
174         filteredRefDataV6(:,i) = filtfilt(lpfilter, filteredRefDataV6(:,i));
175     end
176 end
177
178 % Align the absorption cell reference (taken when the hollow cathode lamp
179 % is off) with the TTL pulses indicating the laser movement
180 % This is done the same way as is done for the normal data
181 %
182 % This variation finds the "high" fine but bad for "lows"
183 % Though, this doesn't matter very much because we only care where it goes
184 % high
185 ipt = findchangepts(refttlsig, 'MinThreshold',100, ...
186     'MinDistance',10, 'Statistic', 'rms');
187 % want the first to be high (5V)
188 if refttlsig(ipt(1)+1) < 1; ipt = ipt(2:end); end
189 numsect = numel(ipt)-1; % throws out the last one as this gets cut off

```

```

190 if mod(numsect,2) == 1; numsect = numsect - 1; end
191 avgLampRef = zeros(ipt(3) - ipt(1),1);
192 % get rid of the intermediate ones, keep full frames together
193 ipt = ipt(1:2:end);
194 chgpts = ipt(1);
195 for idx = 1:numsect/2
196     chgpts = [chgpts chgpts(idx)+totalothersize];
197 end
198 for idx = 1:numsect/2
199     %lampRef{idx} = refabscell(ipt(idx):ipt(idx+1)-1);
200     lampRef{idx} = refabscell(chgpts(idx):chgpts(idx+1)-1);
201     avgLampRef = avgLampRef + lampRef{idx};
202 end
203 avgLampRef = avgLampRef./numsect;
204
205 % Loop through all the data scans
206 % Each "scan" is a full set of data at a single thruster position/plume
207 % position
208 % First scan is the topmost
209 % Thruster translated upward throughout testing to get the plume from
210 % top to bottom
211 for scannum = 1:scans
212     % Size the reference to match the data
213     [row, col] = size(dataarr{scannum}.data);
214     numSweep = round(row/framesize);
215     ref = [];
216     for sweep = 1:numSweep
217         ref = [ref; avgRef];
218     end
219
220     % Remove the ref from the data (accounting for beam path, optics, glass
221     % windows, etc)
222     % Save raw data separately for other analysis
223     dataarr{scannum}.rawdata = dataarr{scannum}.data;
224     dataarr{scannum}.data = dataarr{scannum}.data - ref;
225
226     % Determine absorbance for full (v4, v5, v6)
227     if v4; dataarr{scannum}.fullAbs = ...
228         -log(dataarr{scannum}.rawdata./refdata); end
229     if v5

```

```

230 % filter things, normal bandstop and lowpass
231 dataarr{scannum}.filteredRawDataV5 = zeros(row,col);
232 for i = 1:col
233     dataarr{scannum}.filteredRawDataV5(:,i) = ...
234         bandstop(dataarr{scannum}.rawdata(:,i), ...
235             [bandstop1 bandstop2], fs);
236     dataarr{scannum}.filteredRawDataV5(:,i) = ...
237         lowpass(dataarr{scannum}.filteredRawDataV5(:,i), ...
238             highfilter, fs);
239 end
240 % this done per hartsfield's way instead of -log
241 dataarr{scannum}.fullAbsV5 = 1 - ...
242     (dataarr{scannum}.filteredRawDataV5./filteredRefDataV5);
243 end
244 if v6
245     % filter things, zero-phase bandstop and lowpass
246     dataarr{scannum}.filteredRawDataV6 = zeros(row,col);
247     for i = 1:col
248         dataarr{scannum}.filteredRawDataV6(:,i) = ...
249             filtfilt(bsfilter, dataarr{scannum}.rawdata(:,i));
250         dataarr{scannum}.filteredRawDataV6(:,i) = ...
251             filtfilt(lpfilter, ...
252                 dataarr{scannum}.filteredRawDataV6(:,i));
253     end
254     %this done per hartsfield's way instead of -log
255     dataarr{scannum}.fullAbsV6 = 1 - ...
256         (dataarr{scannum}.filteredRawDataV6./filteredRefDataV6);
257 end
258
259 % Find idx for start of TTL pulse
260 % Throws out stuff before the first changepoint and after the last
261 % This should match up with the start of the "data" frames (from
262 % cameras)
263 % ipt = findchangepts(dataarr{scannum}.ttlsig, ...
264 %     'MinThreshold',4,'MinDistance',3);
265 ipt = findchangepts(dataarr{scannum}.ttlsig, 'MinThreshold',100, ...
266     'MinDistance',10,'Statistic','linear');
267 % want the first to be high (5V)
268 if dataarr{scannum}.ttlsig(ipt(1)+4) < 1; ipt = ipt(2:end); end
269 numsect = numel(ipt)-1;

```

```

270     if mod(numsect,2) == 1; numsect = numsect - 1; end
271     % throw out intermediate (for full scans, up and down)
272     ipt = ipt(1:2:end);
273     chgpts = ipt(1);
274     for idx = 1:numsect/2
275         chgpts = [chgpts chgpts(idx)+totalothersize];
276     end
277     for idx = 1:numsect/2
278         % Finds the full scan (ttl pulse up until next ttl pulse up)
279         % And subtracts out the reference (from the lamp-off measurement)
280         % To account for the glass, neutral Xenon, etc
281         %     dataarr{scannum}.statref{1,idx} = ...
282         %         dataarr{scannum}.abscell(ipt(idx):ipt(idx+1)-1) - avgLampRef;
283         dataarr{scannum}.statref{1,idx} = ...
284             dataarr{scannum}.abscell(chgpts(idx):chgpts(idx+1)-1) - ...
285             avgLampRef;
286     end
287
288     % Find the number of usable sweeps (last is always dropped)
289     [row, col] = size(dataarr{scannum}.data);
290     numsweeps = row/framesize - 1;
291     cameraSweepPoints = [];
292     for i = 1:numsweeps
293         % Round to find the first point for the 155-point sweep
294         % If this causes issues with accuracy, may need to account for the
295         % about 3-pt (of 1000) difference the rounding causes
296         % Assume this first point is the minimum, and 80 points later is
297         % the maximum wavelength
298         firstpt = round(point1/pointRatio) + framesize*(i-1);
299         lastpt = firstpt + (camerasweep - 1);
300         cameraSweepPoints = [cameraSweepPoints firstpt lastpt];
301
302         % Find the points from the DAQ data that correspond to these
303         targetidx = firstpt;
304         %refIntensity = [];
305         % Set up arrays to store the absorbance data
306         % v1 calc: use the DAQ as I0 per Keith's suggestion
307         % v2 calc: use averaged reference as I0
308         % v3 calc: same as v2 but 1 - data/I0 instead of -log(data/I0)
309         if v1; dataarr{scannum}.sweepAbsV1{i} = zeros(camerasweep, col); end

```

```

310     if v2; dataarr{scannum}.sweepAbsV2{i} = zeros(camerasweep, col); end
311     if v3; dataarr{scannum}.sweepAbsV3{i} = zeros(camerasweep, col); end
312     if v4; dataarr{scannum}.sweepAbsV4{i} = zeros(camerasweep, col); end
313     if v5; dataarr{scannum}.sweepAbsV5{i} = zeros(camerasweep, col); end
314     if v6; dataarr{scannum}.sweepAbsV6{i} = zeros(camerasweep, col); end
315     for j = 1:camerasweep
316         DAQidx = 1 + round((j - 1)*pointRatio);
317
318         % Get the sweep's raw data
319         dataarr{scannum}.sweepRaw{i}(j,:) = ...
320             dataarr{scannum}.rawdata(targetidx, :);
321
322         % Calculate absorption based on this reference intensity for
323         % all points at that instant in time
324         if v1; dataarr{scannum}.sweepAbsV1{i}(j,:) = ...
325             -log((dataarr{scannum}.data(targetidx, :) + ...
326                 offset).*scalefactor ./ ...
327                 dataarr{scannum}.statref{i}(DAQidx)); end
328         if v2; dataarr{scannum}.sweepAbsV2{i}(j,:) = ...
329             -log(dataarr{scannum}.rawdata(targetidx, :)./ ...
330                 ref(targetidx, :)); end
331         if v3; dataarr{scannum}.sweepAbsV3{i}(j,:) = 1 - ...
332             dataarr{scannum}.rawdata(targetidx, :)./ ...
333             ref(targetidx, :); end
334         if v4; dataarr{scannum}.sweepAbsV4{i}(j,:) = ...
335             dataarr{scannum}.fullAbs(targetidx, :); end
336         if v5; dataarr{scannum}.sweepAbsV5{i}(j,:) = ...
337             dataarr{scannum}.fullAbsV5(targetidx, :); end
338         if v6; dataarr{scannum}.sweepAbsV6{i}(j,:) = ...
339             dataarr{scannum}.fullAbsV6(targetidx, :); end
340         targetidx = targetidx + 1;
341     end
342 end
343
344 % Now average all the sweeps
345 if v1; dataarr{scannum}.avgAbsV1 = zeros(camerasweep, col); end
346 if v2; dataarr{scannum}.avgAbsV2 = zeros(camerasweep, col); end
347 if v3; dataarr{scannum}.avgAbsV3 = zeros(camerasweep, col); end
348 if v4; dataarr{scannum}.avgAbsV4 = zeros(camerasweep, col); end
349 if v5; dataarr{scannum}.avgAbsV5 = zeros(camerasweep, col); end

```

```

350     if v6; dataarr{scannum}.avgAbsV6 = zeros(cameraSweep,col); end
351     for i = 1:numsweeps
352         if v1; dataarr{scannum}.avgAbsV1 = dataarr{scannum}.avgAbsV1 + ...
353             dataarr{scannum}.sweepAbsV1{i}; end
354         if v2; dataarr{scannum}.avgAbsV2 = dataarr{scannum}.avgAbsV2 + ...
355             dataarr{scannum}.sweepAbsV2{i}; end
356         if v3; dataarr{scannum}.avgAbsV3 = dataarr{scannum}.avgAbsV3 + ...
357             dataarr{scannum}.sweepAbsV3{i}; end
358         if v4; dataarr{scannum}.avgAbsV4 = dataarr{scannum}.avgAbsV4 + ...
359             dataarr{scannum}.sweepAbsV4{i}; end
360         if v5; dataarr{scannum}.avgAbsV5 = dataarr{scannum}.avgAbsV5 + ...
361             dataarr{scannum}.sweepAbsV5{i}; end
362         if v6; dataarr{scannum}.avgAbsV6 = dataarr{scannum}.avgAbsV6 + ...
363             dataarr{scannum}.sweepAbsV6{i}; end
364     end
365     if v1; dataarr{scannum}.avgAbsV1 = ...
366         dataarr{scannum}.avgAbsV1./numsweeps; end
367     if v2; dataarr{scannum}.avgAbsV2 = ...
368         dataarr{scannum}.avgAbsV2./numsweeps; end
369     if v3; dataarr{scannum}.avgAbsV3 = ...
370         dataarr{scannum}.avgAbsV3./numsweeps; end
371     if v4; dataarr{scannum}.avgAbsV4 = ...
372         dataarr{scannum}.avgAbsV4./numsweeps; end
373     if v5; dataarr{scannum}.avgAbsV5 = ...
374         dataarr{scannum}.avgAbsV5./numsweeps; end
375     if v6; dataarr{scannum}.avgAbsV6 = ...
376         dataarr{scannum}.avgAbsV6./numsweeps; end
377 end

```

3.3 filefixer.m

```

1 function procfilenames = filefixer(ftype,filename)
2 % Function to fix up the files output by LabView so they can be read using
3 % tdfread(). This appends a header which basically just has the index for
4 % each pixel (1-2048) and removes garbage chars that somehow appear in the
5 % first line. Assuming nothing changes with how the LabView code outputs
6 % these files, the first 4 indices of the first line are garbage chars so
7 % it takes the first line as 5:end. Adjust as necessary.
8 %
9 % This can handle multiple files if you simply call filefixer(). It will

```

```

10 % bring up a system dialog where you can choose all the files. It will
11 % process them and leave the processed files in the directory from which
12 % your script runs (or if run from the command window, your current
13 % directory)
14 %
15 % To append "ref" to the files , call filefixer('ref'), do this when
16 % processing the I_0 (reference intensity) files to keep them separate.
17 %
18 % If you really want to process specific files , you can specify them as the
19 % second argument. In doing so, if they're reference files the first
20 % argument should be 'ref' and if they're data files , the first argument
21 % should be '' (the comparison with the first argument, ftype, only looks
22 % for 'ref' to do anything).
23
24 if nargin == 0
25     filename = '';
26     ftype = '';
27 elseif nargin == 1
28     if ~isempty(ftype)
29         filename = '';
30     end
31 end
32
33 if isempty(filename)
34
35     %Prompt the user for the file
36     if isempty(ftype)
37         [filename,pathname]= ...
38             uigetfile({'*.*','Input File (*.*)'},...
39                 'Choose a frame to input','MultiSelect','on');
40     else
41         [filename,pathname]= ...
42             uigetfile({'*.*','Reference File (*.*)'},...
43                 'Choose a reference','MultiSelect','on');
44     end
45     filename=fullfile(pathname,filename);
46 end
47 if strcmp(ftype,'ref')
48     processedframe = 'processed_reference';
49 else

```

```

50     processedframe = 'processed_frame';
51 end
52
53 if iscell(filename)
54     %For a list of files
55     thisfile=filename;
56 else
57     thisfile=cellstr(filename);
58 end
59
60 for fnum=1:numel(thisfile)
61     longfname=thisfile{fnum};
62     [pathstr,name,ext]=fileparts(longfname);
63     shortfname=sprintf('%s%s',name,ext);
64
65     fprintf('Converting '%s'...\n',shortfname)
66     fid=fopen(longfname,'rt+');
67
68     if isempty(ftype)
69         processedfname = sprintf('processed-%s',shortfname);
70     else
71         processedfname = sprintf('processed_ref-%s',shortfname);
72     end
73     procfilenames{fnum} = processedfname;
74
75     %this should drop the processed file wherever the script runs
76     fidout = fopen(processedfname,'w');
77
78     header = '';
79     for i = 1:2048
80         header = [header num2str(i) '\t'];
81     end
82
83     header = [header '\n'];
84     fprintf(fidout,header);
85
86     count = 1;
87     while true
88         thisline = fgets(fid);
89         %removes the random stuff in the first line

```

```

90         if count==1; thisline = thisline(5:end); end
91         if ~ischar(thisline); break; end
92         fwrite(fidout,thisline);
93         count = count + 1;
94     end
95
96     fclose(fid);
97     fclose(fidout);
98 end
99 end

```

3.4 selectfiles.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SELECTFILES.M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % Used to get what files the user wants to process. Prompts UI selection.
6  %
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8
9  function filename = selectfiles(ftype)
10
11  if nargin == 0
12      ftype = '';
13  end
14
15  %Prompt the user for the file
16  if strcmp(ftype,'ref')
17      [filename,pathname]=uigetfile({'*.*','Reference File (*.*)'}, ...
18          'Choose a reference','MultiSelect','on');
19  elseif strcmp(ftype,'other')
20      [filename,pathname]=uigetfile({'*.*','Input Other File (*.*)'}, ...
21          'Choose other frame to input','MultiSelect','on');
22  elseif strcmp(ftype,'refother')
23      [filename,pathname]=uigetfile({'*.*', ...
24          'Input Reference Other File (*.*)'}, ...
25          'Choose reference other frame to input','MultiSelect','on');
26  else
27      [filename,pathname]=uigetfile({'*.*','Input File (*.*)'}, ...

```

```

28         'Choose a frame to input','MultiSelect','on');
29     end
30     filename=fullfile(pathname,filename);
31
32     end

```

3.5 importother.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% IMPORTOTHER.M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % This function takes in a cell array of the "other" DAQ files to process
6  % and reads out the absorption cell (photodiode) and TTL signals.
7  %
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10
11 function [abscell,ttlsg] = importother(filename)
12 if nargin == 0
13     filename = '';
14 end
15
16 if isempty(filename)
17     %Prompt the user for the file
18     [filename,pathname]=uigetfile({'*.*','Input Other File (*.*)'},'Choose other
        frame to input','MultiSelect','on');
19     filename=fullfile(pathname,filename);
20 end
21 if iscell(filename)
22     %For a list of files
23     infilename=infilename;
24 else
25     infilename=cellstr(filename);
26 end
27
28 % Copy over the other data to local directory, add header
29 for fnum=1:numel(infilename)
30
31     longfname=infilename{fnum};

```

```

32     [pathstr ,name,ext]=fileparts (longfname);
33     shortfname=sprintf( '%s%s ',name,ext );
34
35     fprintf( 'Converting  ' '%s' '...\n',shortfname)
36     fid=fopen(longfname , 'rt+');
37
38     processedfname = sprintf( 'processed_%s ',shortfname);
39     procfilenames{fnum} = processedfname;
40     fidout = fopen(processedfname , 'w');
41
42     header = 'voltage\tttl\tnoise\n';
43     fprintf(fidout ,header);
44
45     while true
46         thisline = fgets(fid);
47         if ~ischar(thisline); break; end
48         fwrite (fidout ,thisline);
49     end
50
51     fclose(fid);
52     fclose(fidout);
53 end
54
55 % Read in the processed data
56 for fnum=1:numel(procfilenames)
57     fprintf( 'Processing  ' '%s' '...\n',procfilenames{fnum})
58     d{fnum} = struct2cell(tdfread(procfilenames{fnum}));
59 end
60 fprintf( 'Concatenating data...\n')
61 for fnum=1:numel(procfilenames)
62     for iter=1:2 % This reads in only cols 1 and 2 (3 is noise)
63         if fnum==1
64             data{1,iter} = d{1,fnum}{iter ,1};
65         else
66             data{1,iter} = [data{1,iter} ; d{1,fnum}{iter ,1}];
67         end
68     end
69 end
70 other = cell2mat(data);
71 absceil = other(:,1);

```

```

72  ttlsig = other(:,2);
73  end

```

3.6 importdata.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% IMPORTDATA.M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % This function takes in a cell array of the data files to process and the
6  % pixels to cut off data (if chosen to do so)
7  %
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 function data = importdata(datafiles ,pixmin ,pixmax)
11 for fnum=1: numel( datafiles )
12     fprintf( 'Processing  '%s'...\n', datafiles{fnum})
13     d{fnum} = struct2cell( tdfread( datafiles{fnum}));
14 end
15 fprintf( 'Concatenating data...\n')
16 for fnum=1: numel( datafiles )
17     for pix=pixmin: pixmax
18         if fnum==1
19             data{1,pix} = d{1,fnum}{pix,1};
20         else
21             data{1,pix} = [data{1,pix} ; d{1,fnum}{pix,1}];
22         end
23     end
24 end
25 data = cell2mat(data);
26 end

```

3.7 findUncertainty.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % Find the standard deviation so that uncertainty can be reported
6  % Subtract mean from data set

```

```

7 % Square the resulting variation , sum the squares , divide sum by number of
8 % data points , take square root -> gives standard deviation
9 %
10 % This method should be applied to final velocity data and absorption data
11 % reported , we can report values at each pixel for
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13
14 close all
15
16 [row, col] = size(dataarr{1}.sweepVels);
17 % VELOCITY
18 for i = 1:numel(dataarr)
19     unc.stdDevVel{i} = zeros(1,col);
20     for j = 1:col
21         unc.stdDevVel{i}(j) = sqrt( ...
22             sum((dataarr{i}.sweepVels(:,j) - ...
23                 mean(dataarr{i}.sweepVels(:,j))).^2) / ...
24             numel(dataarr{i}.sweepVels(:,j) - ...
25                 mean(dataarr{i}.sweepVels(:,j))));
26     end
27 end
28
29 % ABSORBANCE
30 for i = 1:numel(dataarr)
31     unc.stdDevAbs{i} = zeros(1,col);
32     for j = 1:col
33         unc.stdDevAbs{i}(j) = sqrt( ...
34             sum((dataarr{i}.sweepAbsorb(:,j) - ...
35                 mean(dataarr{i}.sweepAbsorb(:,j))).^2) / ...
36             numel(dataarr{i}.sweepAbsorb(:,j) - ...
37                 mean(dataarr{i}.sweepAbsorb(:,j))));
38     end
39 end
40
41 figure
42 plot(unc.stdDevVel{1}, 'linewidth', 1.5)
43 axis([-inf inf 0 1])
44 hold on
45 plot(unc.stdDevVel{2}, 'linewidth', 1.5)
46 plot(unc.stdDevVel{3}, 'linewidth', 1.5)

```

```

47 plot(unc.stdDevVel{4}, 'linewidth', 1.5)
48 legend('Set 1', 'Set 2', 'Set 3', 'Set 4')
49 hold off
50
51 figure
52 plot(unc.stdDevAbs{1}, 'linewidth', 1.5)
53 axis([-inf inf 0 0.1])
54 hold on
55 plot(unc.stdDevAbs{2}, 'linewidth', 1.5)
56 plot(unc.stdDevAbs{3}, 'linewidth', 1.5)
57 plot(unc.stdDevAbs{4}, 'linewidth', 1.5)
58 legend('Set 1', 'Set 2', 'Set 3', 'Set 4')
59
60
61 %% PROPER PLOTTING – VELOCITY
62 close all
63
64 figure
65 set(gcf, ...
66     'PaperPositionMode', 'auto', ...
67     'Position', [50 50 1200 1000])
68
69 subplot(2,2,1)
70 %plot(unc.stdDevVel{1}, 'black', 'linewidth', 1.5)
71 scatter(1:2048, unc.stdDevVel{1}, 5, 'black', 'filled')
72 axis([-inf inf 0 1])
73 set(gca, ...
74     'FontUnits', 'points', ...
75     'FontWeight', 'normal', ...
76     'FontSize', 14, ...
77     'FontName', 'Times')
78 ylabel('Standard Deviation( $\sigma$ )', ...
79     'FontUnits', 'points', ...
80     'interpreter', 'latex', ...
81     'FontSize', 16, ...
82     'FontName', 'Times')
83 ylpos = get(get(gca, 'ylabel'), 'position');
84 ylpos(1) = ylpos(1) * 1.05;
85 set(get(gca, 'ylabel'), 'position', ylpos)
86 xlabel('Pixel', ...

```

```

87     'FontUnits', 'points', ...
88     'interpreter', 'latex', ...
89     'FontSize', 16, ...
90     'FontName', 'Times')
91 xlp = get(get(gca, 'xlabel'), 'position');
92 xlp(2) = xlp(2)*1.2;
93 set(get(gca, 'xlabel'), 'position', xlp)
94 title('Set 1 (Top of Plume)', ...
95     'FontUnits', 'points', ...
96     'interpreter', 'latex', ...
97     'FontSize', 16, ...
98     'FontName', 'Times')
99
100 subplot(2,2,2)
101 % plot(unc.stdDevVel{2}, 'black', 'linewidth', 1.5)
102 scatter(1:2048, unc.stdDevVel{2}, 5, 'black', 'filled')
103
104 axis([-inf inf 0 1])
105 set(gca, ...
106     'FontUnits', 'points', ...
107     'FontWeight', 'normal', ...
108     'FontSize', 14, ...
109     'FontName', 'Times')
110 ylabel('Standard Deviation( $\sigma$ )', ...
111     'FontUnits', 'points', ...
112     'interpreter', 'latex', ...
113     'FontSize', 16, ...
114     'FontName', 'Times')
115 ylp = get(get(gca, 'ylabel'), 'position');
116 ylp(1) = ylp(1)*1.05;
117 set(get(gca, 'ylabel'), 'position', ylp)
118 xlabel('Pixel', ...
119     'FontUnits', 'points', ...
120     'interpreter', 'latex', ...
121     'FontSize', 16, ...
122     'FontName', 'Times')
123 xlp = get(get(gca, 'xlabel'), 'position');
124 xlp(2) = xlp(2)*1.2;
125 set(get(gca, 'xlabel'), 'position', xlp)
126 title('Set 2 (40mm Down)', ...

```

```

127     'FontUnits', 'points', ...
128     'interpreter', 'latex', ...
129     'FontSize', 16, ...
130     'FontName', 'Times')
131
132 subplot(2,2,3)
133 % plot(unc.stdDevVel{3}, 'black', 'linewidth', 1.5)
134 scatter(1:2048, unc.stdDevVel{3}, 5, 'black', 'filled')
135 axis([-inf inf 0 1])
136 set(gca, ...
137     'FontUnits', 'points', ...
138     'FontWeight', 'normal', ...
139     'FontSize', 14, ...
140     'FontName', 'Times')
141 ylabel('Standard Deviation( $\sigma$ )', ...
142     'FontUnits', 'points', ...
143     'interpreter', 'latex', ...
144     'FontSize', 16, ...
145     'FontName', 'Times')
146 ylpos = get(get(gca, 'ylabel'), 'position');
147 ylpos(1) = ylpos(1)*1.05;
148 set(get(gca, 'ylabel'), 'position', ylpos)
149 xlabel('Pixel', ...
150     'FontUnits', 'points', ...
151     'interpreter', 'latex', ...
152     'FontSize', 16, ...
153     'FontName', 'Times')
154 xlpos = get(get(gca, 'xlabel'), 'position');
155 xlpos(2) = xlpos(2)*1.2;
156 set(get(gca, 'xlabel'), 'position', xlpos)
157 title('Set 3 (80mm Down)', ...
158     'FontUnits', 'points', ...
159     'interpreter', 'latex', ...
160     'FontSize', 16, ...
161     'FontName', 'Times')
162
163 subplot(2,2,4)
164 % plot(unc.stdDevVel{4}, 'black', 'linewidth', 1.5)
165 scatter(1:2048, unc.stdDevVel{4}, 5, 'black', 'filled')
166 axis([-inf inf 0 1])

```

```

167 set(gca,...
168     'FontUnits','points',...
169     'FontWeight','normal',...
170     'FontSize',14,...
171     'FontName','Times')
172 ylabel('Standard Deviation( $\sigma$ )',...
173     'FontUnits','points',...
174     'interpreter','latex',...
175     'FontSize',16,...
176     'FontName','Times')
177 ypos = get(get(gca,'ylabel'),'position');
178 ypos(1) = ypos(1)*1.05;
179 set(get(gca,'ylabel'),'position',ypos)
180 xlabel('Pixel',...
181     'FontUnits','points',...
182     'interpreter','latex',...
183     'FontSize',16,...
184     'FontName','Times')
185 xpos = get(get(gca,'xlabel'),'position');
186 xpos(2) = xpos(2)*1.2;
187 set(get(gca,'xlabel'),'position',xpos)
188 title('Set 4 (120mm Down)',...
189     'FontUnits','points',...
190     'interpreter','latex',...
191     'FontSize',16,...
192     'FontName','Times')
193
194 figure_fname = sprintf('./matlab_figures/std-dev-vel-by-pixel');
195 print(figure_fname,'-dpng','-r300')
196
197 % PROPER PLOTTING- ABSORBANCE
198
199 figure
200 set(gcf,...
201     'PaperPositionMode','auto',...
202     'Position',[50 50 1200 1000])
203
204 subplot(2,2,1)
205 % plot(unc.stdDevAbs{1},'black','linewidth',1.5)
206 scatter(1:2048,unc.stdDevAbs{1},5,'black','filled')

```

```

207 axis([-inf inf 0 0.1])
208 set(gca,...
209     'FontUnits','points',...
210     'FontWeight','normal',...
211     'FontSize',14,...
212     'FontName','Times')
213 ylabel('Standard Deviation( $\sigma$ )',...
214     'FontUnits','points',...
215     'interpreter','latex',...
216     'FontSize',16,...
217     'FontName','Times')
218 ylpos = get(get(gca,'ylabel'),'position');
219 ylpos(1) = ylpos(1)*1.05;
220 set(get(gca,'ylabel'),'position',ylpos)
221 xlabel('Pixel',...
222     'FontUnits','points',...
223     'interpreter','latex',...
224     'FontSize',16,...
225     'FontName','Times')
226 xlpos = get(get(gca,'xlabel'),'position');
227 xlpos(2) = xlpos(2)*1.2;
228 set(get(gca,'xlabel'),'position',xlpos)
229 title('Set 1 (Top of Plume)',...
230     'FontUnits','points',...
231     'interpreter','latex',...
232     'FontSize',16,...
233     'FontName','Times')
234
235 subplot(2,2,2)
236 % plot(unc.stdDevAbs{2},'black','linewidth',1.5)
237 scatter(1:2048,unc.stdDevAbs{2},5,'black','filled')
238 axis([-inf inf 0 0.1])
239 set(gca,...
240     'FontUnits','points',...
241     'FontWeight','normal',...
242     'FontSize',14,...
243     'FontName','Times')
244 ylabel('Standard Deviation( $\sigma$ )',...
245     'FontUnits','points',...
246     'interpreter','latex',...

```

```

247     'FontSize', 16, ...
248     'FontName', 'Times')
249 ylp = get(get(gca, 'ylabel'), 'position');
250 ylp(1) = ylp(1)*1.05;
251 set(get(gca, 'ylabel'), 'position', ylp)
252 xlabel('Pixel', ...
253     'FontUnits', 'points', ...
254     'interpreter', 'latex', ...
255     'FontSize', 16, ...
256     'FontName', 'Times')
257 xlp = get(get(gca, 'xlabel'), 'position');
258 xlp(2) = xlp(2)*1.2;
259 set(get(gca, 'xlabel'), 'position', xlp)
260 title('Set 2 (40mm Down)', ...
261     'FontUnits', 'points', ...
262     'interpreter', 'latex', ...
263     'FontSize', 16, ...
264     'FontName', 'Times')
265
266 subplot(2,2,3)
267 % plot(unc.stdDevAbs{3}, 'black', 'linewidth', 1.5)
268 scatter(1:2048, unc.stdDevAbs{3}, 5, 'black', 'filled')
269 axis([-inf inf 0 0.1])
270 set(gca, ...
271     'FontUnits', 'points', ...
272     'FontWeight', 'normal', ...
273     'FontSize', 14, ...
274     'FontName', 'Times')
275 ylabel('Standard Deviation( $\sigma$ )', ...
276     'FontUnits', 'points', ...
277     'interpreter', 'latex', ...
278     'FontSize', 16, ...
279     'FontName', 'Times')
280 ylp = get(get(gca, 'ylabel'), 'position');
281 ylp(1) = ylp(1)*1.05;
282 set(get(gca, 'ylabel'), 'position', ylp)
283 xlabel('Pixel', ...
284     'FontUnits', 'points', ...
285     'interpreter', 'latex', ...
286     'FontSize', 16, ...

```

```

287     'FontName', 'Times')
288 xlp_pos = get(get(gca, 'xlabel'), 'position');
289 xlp_pos(2) = xlp_pos(2)*1.2;
290 set(get(gca, 'xlabel'), 'position', xlp_pos)
291 title('Set 3 (80mm Down)', ...
292     'FontUnits', 'points', ...
293     'interpreter', 'latex', ...
294     'FontSize', 16, ...
295     'FontName', 'Times')
296
297 subplot(2,2,4)
298 plot(unc.stdDevAbs{4}, 'black', 'linewidth', 1.5)
299 scatter(1:2048, unc.stdDevAbs{4}, 5, 'black', 'filled')
300 axis([-inf inf 0 0.1])
301 set(gca, ...
302     'FontUnits', 'points', ...
303     'FontWeight', 'normal', ...
304     'FontSize', 14, ...
305     'FontName', 'Times')
306 ylabel('Standard Deviation( $\sigma$ )', ...
307     'FontUnits', 'points', ...
308     'interpreter', 'latex', ...
309     'FontSize', 16, ...
310     'FontName', 'Times')
311 ylp_pos = get(get(gca, 'ylabel'), 'position');
312 ylp_pos(1) = ylp_pos(1)*1.05;
313 set(get(gca, 'ylabel'), 'position', ylp_pos)
314 xlabel('Pixel', ...
315     'FontUnits', 'points', ...
316     'interpreter', 'latex', ...
317     'FontSize', 16, ...
318     'FontName', 'Times')
319 xlp_pos = get(get(gca, 'xlabel'), 'position');
320 xlp_pos(2) = xlp_pos(2)*1.2;
321 set(get(gca, 'xlabel'), 'position', xlp_pos)
322 title('Set 4 (120mm Down)', ...
323     'FontUnits', 'points', ...
324     'interpreter', 'latex', ...
325     'FontSize', 16, ...
326     'FontName', 'Times')

```

```

327
328 figure_fname = sprintf('./matlab_figures/std-dev-abs-by-pixel');
329 print (figure_fname, '-dpng', '-r300')

```

3.8 findPossibleVelsFromExB.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % Finds possible values for exit vel based on Bui ExB calcs
6  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7
8  close all;
9
10 exb.voltages = [18 21 24];
11 exb.b = 0.7; %Tesla
12 %exb.d = 0.003; %m
13 exb.d = 0.015; %m
14
15 exb.vel = exb.voltages./(exb.d.*exb.b)

```

3.9 findAxialVelocityForOffAngle.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % This finds the axial velocity of a given particle as the code would
6  % determine, if the particle is off-center travelling at the maximum
7  % velocity
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 % positive angles left, toward mirrors/cameras
11 % negative angles right, toward laser collimator (source)
12
13 xx.vel = 10; %km/s
14 xx.angle = 10; %deg
15
16 xx.laser = 17; %deg

```

```

17
18 xx.sideangle = 180 - 107 - xx.angle;
19 xx.laseraxisvel = xx.vel*cosd(xx.sideangle);
20 xx.axialvel = xx.laseraxisvel/cosd(73)

```

3.10 getAveragedAbsorbanceByVelocityBin.m

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3 % Last Updated: 1 Feb 2019
4 %
5 % This code is intended to, at certain pixels, average the absorbance
6 % across all sweeps, by velocity bin. The result is a vector of average
7 % absorbances at each bin for the entire 0.4 seconds of data.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 AA.pixels = [350 550 1400 1500 1600 1700];
11 AA.dtRange = 26:61;
12
13 for i = 1:numel(dataarr)
14     AA.avgBinnedAbs{i} = zeros(numel(AA.dtRange),numel(AA.pixels));
15     for j = 1:numel(AA.pixels) %loop thru pix of interest
16         AA.pix = AA.pixels(j);
17         for k = 1:numel(AA.dtRange) %at each vel bin
18             AA.avgAbs = 0;
19             AA.idx = AA.dtRange(k);
20             for w = 1:numel(dataarr{i}.sweepAbsV5) %loop thru sweeps
21                 AA.avgAbs = AA.avgAbs + ...
22                     dataarr{i}.sweepAbsV5{w}(AA.idx,AA.pix);
23             end
24             AA.avgAbs = AA.avgAbs/numel(dataarr{i}.sweepAbsV5);
25             AA.avgBinnedAbs{i}(k,j) = AA.avgAbs;
26         end
27     end
28 end

```

3.11 getAveragedNumberDensities.m

```

1 close all;
2

```

```

3 % This will go through and average the relative number densities for a
4 % sweep (relative num density represented by absorption)
5 % 0 to 20 km/s corresponds to indices 26:61
6
7 %test = dot(dataarr{1}.sweepAbsV5{4}(26:61,1400),velldist(26:61))/numel(26:61) % gives
    average vel in km/s
8 % Want to do this to get averaged vels, also average the absorption
9 % so we get the average vel and average absorption
10 %test2 = mean(dataarr{1}.sweepAbsV5{4}(26:61,1400))
11
12 [row, col] = size(dataarr{1}.sweepAbsV5{1}); %assumes these variables exist
13 for scannum = 1:numel(dataarr)
14     fprintf('Processing scan %s...\n',num2str(scannum))
15     dataarr{scannum}.sweepVels = zeros(numel(dataarr{scannum}.sweepAbsV5),col);
16     dataarr{scannum}.sweepAbsorb = zeros(numel(dataarr{scannum}.sweepAbsV5),col);
17     dataarr{scannum}.overallResults = zeros(2,col); % 1 - avg vel, 2 - avg absorbance
18     for i = 1:numel(dataarr{scannum}.sweepAbsV5) % loop thru sweeps
19         for j = 1:col % loop thru pixels
20             %
21                 dataarr{scannum}.sweepVels(i,j) = ...
22                     dot(dataarr{scannum}.sweepAbsV5{i}(26:61,j), ...
23                         velldist(26:61))/numel(26:61);
24             dataarr{scannum}.sweepVels(i,j) = ...
25                 dot(dataarr{scannum}.sweepAbsV5{i}(26:61,j), ...
26                     velldist(26:61))/ ...
27                     sum(dataarr{scannum}.sweepAbsV5{i}(26:61,j));
28             dataarr{scannum}.sweepAbsorb(i,j) = ...
29                 mean(dataarr{scannum}.sweepAbsV5{i}(26:61,j));
30         end
31     end
32     fprintf('Completed processing individual sweeps...\n')
33     for j = 1:col % loop thru pixels again
34         %
35             dataarr{scannum}.overallResults(1,j) = ...
36                 dot(dataarr{scannum}.sweepVels(:,j), ...
37                     dataarr{scannum}.sweepAbsorb(:,j))/ ...
38                     numel(dataarr{scannum}.sweepAbsV5);
39             dataarr{scannum}.overallResults(1,j) = ...
40                 dot(dataarr{scannum}.sweepVels(:,j), ...
41                     dataarr{scannum}.sweepAbsorb(:,j))/ ...
42                     sum(dataarr{scannum}.sweepAbsorb(:,j));
43             dataarr{scannum}.overallResults(2,j) = ...

```

```

42         mean(dataarr{scannum}.sweepAbsorb(:,j));
43     end
44     fprintf('Completed averaging sweeps...\n')
45 end
46 fprintf('Processing complete.\n')

```

3.12 pixel2mm.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PIXEL2MM.M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % This script reads in an image file (from the cameras) and finds the
6  % pixel-to-mm and pixel-to-inch mappings.
7  %
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9  close all
10
11 pix2mm.image = imread('./pix2mm/pix2mm-1.png');
12 pix2mm.image2 = imread('./pix2mm/pix2mm-2_0.5mm.png');
13 pix2mm.image3 = imread('./pix2mm/pix2mm-3_1.5mm.png');
14 pix2mm.image4 = imread('./pix2mm/pix2mm-4_2.5mm.png');
15
16
17 plot(mean(pix2mm.image))
18 hold on
19 plot(mean(pix2mm.image2))
20 plot(mean(pix2mm.image3))
21 plot(mean(pix2mm.image4))
22
23 pix2mm.numpix = 1324;
24 pix2mm.inches = 1.5;
25 pix2mm.in2mm = 25.4;
26 pix2mm.mm = pix2mm.inches*pix2mm.in2mm;
27
28 pix2mm.pix2in = pix2mm.numpix/pix2mm.inches; % number of pixels per inch
29 pix2mm.pix2mm = pix2mm.numpix/pix2mm.mm; % number of pixels per mm
30
31 hold off
32 figure

```

```

33 plot(mean(avgRef))
34
35 % These chosen from observing the plot of average intensity across avgRef
36 pix2mm.maxpixel = 1729;
37 pix2mm.minpixel = 337;
38
39 % maximum visible is then
40 % counting both edge pixels
41 pix2mm.visiblePixels = pix2mm.maxpixel - pix2mm.minpixel + 1;
42 pix2mm.visibleInches = pix2mm.visiblePixels/pix2mm.pix2in;
43 pix2mm.visiblemm = pix2mm.visiblePixels/pix2mm.pix2mm;

```

3.13 getAveragedDAQFromSweep1.m

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3 % Last Updated: 1 Feb 2019
4 %
5 % Assumes that you've loaded data
6 %
7 % This grabs the averaged DAQ data for comparison to the reference, to show
8 % that the noise from the hollow cathode lamp overwhelms the laser
9 % intensity for the data we collected
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11
12 dataarr{1}.avgDAQ = zeros(size(dataarr{1}.statref{1}));
13 for i = 1:numel(dataarr{1}.statref)
14     dataarr{1}.avgDAQ = dataarr{1}.avgDAQ + dataarr{1}.statref{i} + ...
15         avgLampRef;
16 end
17 dataarr{1}.avgDAQ = dataarr{1}.avgDAQ./numel(dataarr{1}.statref);

```

3.14 miscCalculations.m

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3 % Last Updated: 1 Feb 2019
4 %
5 % General calculations, unrelated to the other data processing. This will
6 % figure out the maximum theoretical velocity and the total sample time.

```

```

7  %
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10 clc; clear; close all;
11
12 % find max velocity we'll be able to see
13 Vd = 300; % discharge voltage
14 n = 1; % number ionized
15 e = 1.60217662E-19; % coulombs
16 Mion = 131.29; %amu
17 amu2kg = 1.66054E-27; %kg
18
19 maxvel = sqrt(2*n*e*Vd/(Mion*amu2kg));
20
21 fprintf('The maximum velocity is %.2f m/s\n', maxvel)
22
23 % how many seconds of data did we capture
24 cameraFPS = 80000; %80,000 Hz
25 samples = 32000;
26 totalTime = samples/cameraFPS;
27
28 fprintf('The total time sampled was %.4f seconds\n',totalTime)

```

3.15 plotAveragedAbsorbanceByVelocityBin.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % Plots the averaged absorbances by velocity bin
6  %
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8
9  close all
10 AA.legend = [];
11
12 figure
13
14 set(gcf,...
15     'PaperPositionMode','auto',...

```

```

16     'Position',[50 50 1200 1000])
17
18 subplot(2,2,1)
19 hold on
20 err = zeros(1,36);
21 for i = 1:numel(AA.pixels)
22     err(:) = unc.stdDevAbs{1}(AA.pixels(i));
23     errorbar(veldist(26:61),AA.avgBinnedAbs{1}(:,i), ...
24         err,'o','MarkerSize',5)
25     AA.legend{i} = sprintf('Pixel %s', ...
26         num2str(AA.pixels(i)));
27     axis([-inf inf 0 1])
28 end
29 hold off
30 set(gca,...
31     'FontUnits','points',...
32     'FontWeight','normal',...
33     'FontSize',14,...
34     'FontName','Times')
35 ylabel('Absorbance',...
36     'FontUnits','points',...
37     'interpreter','latex',...
38     'FontSize',16,...
39     'FontName','Times')
40 ylpos = get(get(gca,'ylabel'),'position');
41 ylpos(1) = ylpos(1)*1.05;
42 set(get(gca,'ylabel'),'position',ylpos)
43 xlabel('$km/s$',...
44     'FontUnits','points',...
45     'interpreter','latex',...
46     'FontSize',16,...
47     'FontName','Times')
48 xlpos = get(get(gca,'xlabel'),'position');
49 xlpos(2) = xlpos(2)*1.2;
50 set(get(gca,'xlabel'),'position',xlpos)
51 title('Set 1 (Top of Plume)',...
52     'FontUnits','points',...
53     'interpreter','latex',...
54     'FontSize',16,...
55     'FontName','Times')

```

```

56 legend(AA.legend,...
57     'FontUnits','points',...
58     'FontWeight','normal',...
59     'FontSize',13,...
60     'FontName','Times',...
61     'Location','NorthWest')
62
63 subplot(2,2,2)
64 hold on
65 for i = 1:numel(AA.pixels)
66     err(:) = unc.stdDevAbs{2}(AA.pixels(i));
67     errorbar(veldist(26:61),AA.avgBinnedAbs{2}(:,i), ...
68         err,'o','MarkerSize',5)
69     AA.legend{i} = sprintf('Pixel %s', ...
70         num2str(AA.pixels(i)));
71     axis([-inf inf 0 1])
72 end
73 hold off
74 set(gca,...
75     'FontUnits','points',...
76     'FontWeight','normal',...
77     'FontSize',14,...
78     'FontName','Times')
79 ylabel('Absorbance',...
80     'FontUnits','points',...
81     'interpreter','latex',...
82     'FontSize',16,...
83     'FontName','Times')
84 ylpos = get(get(gca,'ylabel'),'position');
85 ylpos(1) = ylpos(1)*1.05;
86 set(get(gca,'ylabel'),'position',ylpos)
87 xlabel('$km/s$',...
88     'FontUnits','points',...
89     'interpreter','latex',...
90     'FontSize',16,...
91     'FontName','Times')
92 xlpos = get(get(gca,'xlabel'),'position');
93 xlpos(2) = xlpos(2)*1.2;
94 set(get(gca,'xlabel'),'position',xlpos)
95 title('Set 2 (40mm Down)',...

```

```

96     'FontUnits','points',...
97     'interpreter','latex',...
98     'FontSize',16,...
99     'FontName','Times')
100 legend(AA.legend,...
101     'FontUnits','points',...
102     'FontWeight','normal',...
103     'FontSize',13,...
104     'FontName','Times',...
105     'Location','NorthWest')
106
107 subplot(2,2,3)
108 hold on
109 for i = 1:numel(AA.pixels)
110     err(:) = unc.stdDevAbs{3}(AA.pixels(i));
111     errorbar(veldist(26:61),AA.avgBinnedAbs{3}(:,i), ...
112         err, 'o', 'MarkerSize',5)
113     AA.legend{i} = sprintf('Pixel %s', ...
114         num2str(AA.pixels(i)));
115     axis([-inf inf 0 1])
116 end
117 hold off
118 set(gca,...
119     'FontUnits','points',...
120     'FontWeight','normal',...
121     'FontSize',14,...
122     'FontName','Times')
123 ylabel('Absorbance',...
124     'FontUnits','points',...
125     'interpreter','latex',...
126     'FontSize',16,...
127     'FontName','Times')
128 ylpos = get(get(gca,'ylabel'),'position');
129 ylpos(1) = ylpos(1)*1.05;
130 set(get(gca,'ylabel'),'position',ylpos)
131 xlabel('$\text{km/s}^2$',...
132     'FontUnits','points',...
133     'interpreter','latex',...
134     'FontSize',16,...
135     'FontName','Times')

```

```

136 xlp = get(get(gca,'xlabel'),'position');
137 xlp(2) = xlp(2)*1.2;
138 set(get(gca,'xlabel'),'position',xlp)
139 title('Set 3 (80mm Down)',...
140       'FontUnits','points',...
141       'interpreter','latex',...
142       'FontSize',16,...
143       'FontName','Times')
144 legend(AA.legend,...
145        'FontUnits','points',...
146        'FontWeight','normal',...
147        'FontSize',13,...
148        'FontName','Times',...
149        'Location','SouthWest')
150
151 subplot(2,2,4)
152 hold on
153 for i = 1:numel(AA.pixels)
154     err(:) = unc.stdDevAbs{4}(AA.pixels(i));
155     errorbar(veldist(26:61),AA.avgBinnedAbs{4}(:,i), ...
156             err,'o','MarkerSize',5)
157     AA.legend{i} = sprintf('Pixel %s', ...
158                             num2str(AA.pixels(i)));
159     axis([-inf inf 0 1])
160 end
161 hold off
162 set(gca,...
163      'FontUnits','points',...
164      'FontWeight','normal',...
165      'FontSize',14,...
166      'FontName','Times')
167 ylabel('Absorbance',...
168        'FontUnits','points',...
169        'interpreter','latex',...
170        'FontSize',16,...
171        'FontName','Times')
172 ylp = get(get(gca,'ylabel'),'position');
173 ylp(1) = ylp(1)*1.05;
174 set(get(gca,'ylabel'),'position',ylp)
175 xlabel('$km/s$',...

```

```

176     'FontUnits','points',...
177     'interpreter','latex',...
178     'FontSize',16,...
179     'FontName','Times')
180 xlpas = get(get(gca,'xlabel'),'position');
181 xlpas(2) = xlpas(2)*1.2;
182 set(get(gca,'xlabel'),'position',xlpas)
183 title('Set 4 (120mm Down)',...
184     'FontUnits','points',...
185     'interpreter','latex',...
186     'FontSize',16,...
187     'FontName','Times')
188 legend(AA.legend,...
189     'FontUnits','points',...
190     'FontWeight','normal',...
191     'FontSize',13,...
192     'FontName','Times',...
193     'Location','SouthWest')
194
195 figure_fname = sprintf("./matlab_figures/vel-dist-with-error-specific-pix");
196 print (figure_fname, '-dpng', '-r300')
197
198 figure
199 plot (veldist(26:61),AA.avgBinnedAbs{4}(:,5),'linewidth',1.5)
200 xlabel('km/s')
201 ylabel('absorbance')
202 axis([-inf inf 0.649 0.6535])

```

3.16 plotThingsFiltered.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Author: 2d Lt Avery Leonard, Air Force Institute of Technology
3  % Last Updated: 1 Feb 2019
4  %
5  % Plots pretty much everything. Run after the data is loaded into the
6  % workspace, either after running analyzedata.m or loading from a .MAT
7  % file.
8  %
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10

```

```

11
12 close all;
13
14 pixels_to_plot = [700 800 900 1000 1100 1200 1300 1400 1500 1600];
15
16 filterversion = "normal";
17 for i = 1:numel(dataarr)
18     for j = 1:numel(pixels_to_plot)
19         figure
20         subplot(3,3,1)
21         plot(veldist, dataarr{i}.sweepAbsV5{1}(:, pixels_to_plot(j)), 'linewidth', 2)
22         title(sprintf('Filtered absorbance at point %s, sweep 1, set %s', num2str(
                pixels_to_plot(j)), num2str(i)))
23         axis([-inf inf 0 1])
24         xlabel('km/s')
25         ylabel('1 - I/I0')
26         subplot(3,3,2)
27         plot(veldist, dataarr{i}.sweepAbsV5{2}(:, pixels_to_plot(j)), 'linewidth', 2)
28         title(sprintf('Filtered absorbance at point %s, sweep 2, set %s', num2str(
                pixels_to_plot(j)), num2str(i)))
29         axis([-inf inf 0 1])
30         ylabel('1 - I/I0')
31         xlabel('km/s')
32         subplot(3,3,3)
33         plot(veldist, dataarr{i}.sweepAbsV5{3}(:, pixels_to_plot(j)), 'linewidth', 2)
34         title(sprintf('Filtered absorbance at point %s, sweep 3, set %s', num2str(
                pixels_to_plot(j)), num2str(i)))
35         axis([-inf inf 0 1])
36         ylabel('1 - I/I0')
37         xlabel('km/s')
38         subplot(3,3,4)
39         plot(veldist, dataarr{i}.sweepAbsV5{4}(:, pixels_to_plot(j)), 'linewidth', 2)
40         title(sprintf('Filtered absorbance at point %s, sweep 4, set %s', num2str(
                pixels_to_plot(j)), num2str(i)))
41         axis([-inf inf 0 1])
42         ylabel('1 - I/I0')
43         xlabel('km/s')
44         subplot(3,3,5)
45         plot(veldist, dataarr{i}.sweepAbsV5{5}(:, pixels_to_plot(j)), 'linewidth', 2)
46         title(sprintf('Filtered absorbance at point %s, sweep 5, set %s', num2str(

```

```

        pixels_to_plot(j)),num2str(i)))
47     axis([-inf inf 0 1])
48     ylabel('1 - I/I0')
49     xlabel('km/s')
50     subplot(3,3,6)
51     plot(veldist,dataarr{i}.sweepAbsV5{6}(:,pixels_to_plot(j)),'linewidth',2)
52     title(sprintf('Filtered absorbance at point %s, sweep 6, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
53     axis([-inf inf 0 1])
54     ylabel('1 - I/I0')
55     xlabel('km/s')
56     subplot(3,3,7)
57     plot(veldist,dataarr{i}.sweepAbsV5{7}(:,pixels_to_plot(j)),'linewidth',2)
58     title(sprintf('Filtered absorbance at point %s, sweep 7, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
59     axis([-inf inf 0 1])
60     ylabel('1 - I/I0')
61     xlabel('km/s')
62     subplot(3,3,8)
63     plot(veldist,dataarr{i}.sweepAbsV5{8}(:,pixels_to_plot(j)),'linewidth',2)
64     title(sprintf('Filtered absorbance at point %s, sweep 8, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
65     axis([-inf inf 0 1])
66     ylabel('1 - I/I0')
67     xlabel('km/s')
68     subplot(3,3,9)
69     plot(veldist,dataarr{i}.sweepAbsV5{9}(:,pixels_to_plot(j)),'linewidth',2)
70     title(sprintf('Filtered absorbance at point %s, sweep 9, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
71     axis([-inf inf 0 1])
72     ylabel('1 - I/I0')
73     xlabel('km/s')
74
75     figure_fname = sprintf("./matlab_figures/absorbance_sweeps_1-9_pixel%s_set%s_
        %s_filter",num2str(pixels_to_plot(j)),num2str(i),filterversion);
76     set(gcf,'PaperPositionMode','auto')
77     set(gcf,'Position',[10 50 2000 1000])
78     print(figure_fname,'-dpng','-r300')
79     end
80 end

```

```

81
82 filterversion = "zero-phase";
83 for i = 1:numel(dataarr)
84     for j = 1:numel(pixels_to_plot)
85         figure
86         subplot(3,3,1)
87         plot(veldist , dataarr{i}.sweepAbsV6{1}(:, pixels_to_plot(j)), 'linewidth', 2)
88         title(sprintf('Filtered absorbance at point %s, sweep 1, set %s', num2str(
            pixels_to_plot(j)), num2str(i)))
89         axis([-inf inf 0 1])
90         xlabel('km/s')
91         ylabel('1 - I/I0')
92         subplot(3,3,2)
93         plot(veldist , dataarr{i}.sweepAbsV6{2}(:, pixels_to_plot(j)), 'linewidth', 2)
94         title(sprintf('Filtered absorbance at point %s, sweep 2, set %s', num2str(
            pixels_to_plot(j)), num2str(i)))
95         axis([-inf inf 0 1])
96         ylabel('1 - I/I0')
97         xlabel('km/s')
98         subplot(3,3,3)
99         plot(veldist , dataarr{i}.sweepAbsV6{3}(:, pixels_to_plot(j)), 'linewidth', 2)
100        title(sprintf('Filtered absorbance at point %s, sweep 3, set %s', num2str(
            pixels_to_plot(j)), num2str(i)))
101        axis([-inf inf 0 1])
102        ylabel('1 - I/I0')
103        xlabel('km/s')
104        subplot(3,3,4)
105        plot(veldist , dataarr{i}.sweepAbsV6{4}(:, pixels_to_plot(j)), 'linewidth', 2)
106        title(sprintf('Filtered absorbance at point %s, sweep 4, set %s', num2str(
            pixels_to_plot(j)), num2str(i)))
107        axis([-inf inf 0 1])
108        ylabel('1 - I/I0')
109        xlabel('km/s')
110        subplot(3,3,5)
111        plot(veldist , dataarr{i}.sweepAbsV6{5}(:, pixels_to_plot(j)), 'linewidth', 2)
112        title(sprintf('Filtered absorbance at point %s, sweep 5, set %s', num2str(
            pixels_to_plot(j)), num2str(i)))
113        axis([-inf inf 0 1])
114        ylabel('1 - I/I0')
115        xlabel('km/s')

```

```

116     subplot(3,3,6)
117     plot(veldist,dataarr{i}.sweepAbsV6{6}(:,pixels_to_plot(j)),'linewidth',2)
118     title(sprintf('Filtered absorbance at point %s, sweep 6, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
119     axis([-inf inf 0 1])
120     ylabel('1 - I/I0')
121     xlabel('km/s')
122     subplot(3,3,7)
123     plot(veldist,dataarr{i}.sweepAbsV6{7}(:,pixels_to_plot(j)),'linewidth',2)
124     title(sprintf('Filtered absorbance at point %s, sweep 7, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
125     axis([-inf inf 0 1])
126     ylabel('1 - I/I0')
127     xlabel('km/s')
128     subplot(3,3,8)
129     plot(veldist,dataarr{i}.sweepAbsV6{8}(:,pixels_to_plot(j)),'linewidth',2)
130     title(sprintf('Filtered absorbance at point %s, sweep 8, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
131     axis([-inf inf 0 1])
132     ylabel('1 - I/I0')
133     xlabel('km/s')
134     subplot(3,3,9)
135     plot(veldist,dataarr{i}.sweepAbsV6{9}(:,pixels_to_plot(j)),'linewidth',2)
136     title(sprintf('Filtered absorbance at point %s, sweep 9, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
137     axis([-inf inf 0 1])
138     ylabel('1 - I/I0')
139     xlabel('km/s')
140
141     figure_fname = sprintf("./matlab_figures/absorbance_sweeps_1-9_pixel%s_set%s_
        %s_filter",num2str(pixels_to_plot(j)),num2str(i),filterversion);
142     set(gcf,'PaperPositionMode','auto')
143     set(gcf,'Position',[10 50 2000 1000])
144     print(figure_fname,'-dpng','-r300')
145     end
146 end
147
148 %% Plots to show the difference between regular filtering and zero-phase filtering
149 close all;
150 figure

```

```

151 plot(veldist, dataarr{2}.sweepAbsV5{1}(:,1500), 'linewidth', 2)
152 hold on
153 plot(veldist, dataarr{2}.sweepAbsV6{1}(:,1500), 'linewidth', 2)
154 axis([-inf inf 0 1])
155 % ylabel('1 - I/I0')
156 % xlabel('km/s')
157 % legend('Normal Filtering', 'Zero-Phase Filtering')
158 % set(gcf, 'PaperPositionMode', 'auto')
159 %     'Position', [25 25 750 450], ...
160 % set(gcf, 'Position', [10 50 800 500])
161 set(gcf, ...
162     'PaperPositionMode', 'auto', ...
163     'Position', [500 500 800 500])
164 set(gca, ...
165     'Units', 'normalized', ...
166     'Position', [.12 .18 .8 .73], ...
167     'FontUnits', 'points', ...
168     'FontWeight', 'normal', ...
169     'FontSize', 14, ...
170     'FontName', 'Times')
171 ylabel({'$1 - \frac{I}{I_0}$'}, ...
172     'FontUnits', 'points', ...
173     'interpreter', 'latex', ...
174     'FontSize', 20, ...
175     'FontName', 'Times')
176 ylpos = get(get(gca, 'ylabel'), 'position');
177 ylpos(1) = ylpos(1)*1.05;
178 set(get(gca, 'ylabel'), 'position', ylpos)
179 xlabel('$\frac{km}{s}$', ...
180     'FontUnits', 'points', ...
181     'interpreter', 'latex', ...
182     'FontSize', 20, ...
183     'FontName', 'Times')
184 xlpos = get(get(gca, 'xlabel'), 'position');
185 xlpos(2) = xlpos(2)*1.2;
186 set(get(gca, 'xlabel'), 'position', xlpos)
187 legend({'Normal Filtering', 'Zero-Phase Filtering'}, ...
188     'FontUnits', 'points', ...
189     'FontWeight', 'normal', ...
190     'FontSize', 14, ...

```

```

191     'FontName', 'Times', ...
192     'Location', 'NorthEast')
193 figure_fname = './matlab_figures/regular-and-zero-phase-filter-comparison';
194 print(figure_fname, '-dpng', '-r300')
195
196
197
198 %% More plotting, but only for 0-20 km/s
199 % Do this because we only care about that range
200 % The above show it well enough
201 close all;
202
203 filterversion = "normal";
204 for i = 1:numel(dataarr)
205     for j = 1:numel(pixels_to_plot)
206         figure
207         subplot(3,3,1)
208         plot(veldist(26:61), dataarr{i}.sweepAbsV5{1}(26:61, pixels_to_plot(j)), '
                linewidth', 2)
209         title(sprintf('Filtered absorbance at point %s, sweep 1, set %s', num2str(
                pixels_to_plot(j)), num2str(i)))
210         axis([-inf inf 0 1])
211         xlabel('km/s')
212         ylabel('1 - I/I0')
213         subplot(3,3,2)
214         plot(veldist(26:61), dataarr{i}.sweepAbsV5{2}(26:61, pixels_to_plot(j)), '
                linewidth', 2)
215         title(sprintf('Filtered absorbance at point %s, sweep 2, set %s', num2str(
                pixels_to_plot(j)), num2str(i)))
216         axis([-inf inf 0 1])
217         ylabel('1 - I/I0')
218         xlabel('km/s')
219         subplot(3,3,3)
220         plot(veldist(26:61), dataarr{i}.sweepAbsV5{3}(26:61, pixels_to_plot(j)), '
                linewidth', 2)
221         title(sprintf('Filtered absorbance at point %s, sweep 3, set %s', num2str(
                pixels_to_plot(j)), num2str(i)))
222         axis([-inf inf 0 1])
223         ylabel('1 - I/I0')
224         xlabel('km/s')

```

```

225     subplot(3,3,4)
226     plot(veldist(26:61),dataarr{i}.sweepAbsV5{4}(26:61,pixels_to_plot(j)), '
        linewidth',2)
227     title(sprintf('Filtered absorbance at point %s, sweep 4, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
228     axis([-inf inf 0 1])
229     ylabel('1 - I/I0')
230     xlabel('km/s')
231     subplot(3,3,5)
232     plot(veldist(26:61),dataarr{i}.sweepAbsV5{5}(26:61,pixels_to_plot(j)), '
        linewidth',2)
233     title(sprintf('Filtered absorbance at point %s, sweep 5, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
234     axis([-inf inf 0 1])
235     ylabel('1 - I/I0')
236     xlabel('km/s')
237     subplot(3,3,6)
238     plot(veldist(26:61),dataarr{i}.sweepAbsV5{6}(26:61,pixels_to_plot(j)), '
        linewidth',2)
239     title(sprintf('Filtered absorbance at point %s, sweep 6, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
240     axis([-inf inf 0 1])
241     ylabel('1 - I/I0')
242     xlabel('km/s')
243     subplot(3,3,7)
244     plot(veldist(26:61),dataarr{i}.sweepAbsV5{7}(26:61,pixels_to_plot(j)), '
        linewidth',2)
245     title(sprintf('Filtered absorbance at point %s, sweep 7, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
246     axis([-inf inf 0 1])
247     ylabel('1 - I/I0')
248     xlabel('km/s')
249     subplot(3,3,8)
250     plot(veldist(26:61),dataarr{i}.sweepAbsV5{8}(26:61,pixels_to_plot(j)), '
        linewidth',2)
251     title(sprintf('Filtered absorbance at point %s, sweep 8, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
252     axis([-inf inf 0 1])
253     ylabel('1 - I/I0')
254     xlabel('km/s')

```

```

255     subplot(3,3,9)
256     plot(veldist(26:61),dataarr{i}.sweepAbsV5{9}(26:61,pixels_to_plot(j)), '
        linewidth',2)
257     title(sprintf('Filtered absorbance at point %s, sweep 9, set %s',num2str(
        pixels_to_plot(j)),num2str(i)))
258     axis([-inf inf 0 1])
259     ylabel('1 - I/I0')
260     xlabel('km/s')
261
262     figure_fname = sprintf("./matlab-figures/cropped_absorbance_sweeps_1-9_pixel%
        s_set%s_%s_filter",num2str(pixels_to_plot(j)),num2str(i),filterversion);
263     set(gcf,'PaperPositionMode','auto')
264     set(gcf,'Position',[10 50 2000 1000])
265     print(figure_fname, '-dpng', '-r300')
266 end
267 end
268
269 %% Plotting for the averaged number densities and velocities
270 close all;
271
272 figure
273 set(gcf,...
274     'PaperPositionMode','auto',...
275     'Position',[50 50 1200 1000])
276
277 subplot(2,2,1)
278 %surf(1:2048,1:199,dataarr{1}.sweepVels,'edgecolor','none')
279 mesh(dataarr{1}.sweepVels)
280 axis([0 2048 0 200 7 13])
281 colormap jet
282 caxis([9 11.5])
283 view(-16,66)
284 set(gca,...
285     'FontUnits','points',...
286     'FontWeight','normal',...
287     'FontSize',14,...
288     'FontName','Times')
289 ylabel('Sweep',...
290     'FontUnits','points',...
291     'interpreter','latex',...

```

```

292     'FontSize',16,...
293     'FontName','Times')
294 % ypos = get(get(gca,'ylabel'),'position');
295 % ypos(1) = ypos(1)*1.05;
296 % set(get(gca,'ylabel'),'position',ypos)
297 xlabel('Pixel',...
298     'FontUnits','points',...
299     'interpreter','latex',...
300     'FontSize',16,...
301     'FontName','Times')
302 % xpos = get(get(gca,'xlabel'),'position');
303 % xpos(2) = xpos(2)*1.2;
304 % set(get(gca,'xlabel'),'position',xpos)
305 ylabel('Sweep Velocity',...
306     'FontUnits','points',...
307     'interpreter','latex',...
308     'FontSize',16,...
309     'FontName','Times')
310 zpos = get(get(gca,'zlabel'),'position');
311 zpos(1) = zpos(1) - 30;
312 zpos(3) = zpos(3) + 1;
313 set(get(gca,'zlabel'),'position',zpos)
314 title('Set 1 (Top of Plume)',...
315     'FontUnits','points',...
316     'interpreter','latex',...
317     'FontSize',16,...
318     'FontName','Times')
319
320 subplot(2,2,2)
321 mesh(dataarr{2}.sweepVels)
322 axis([0 2048 0 200 7 13])
323 colormap jet
324 caxis([9 11.5])
325 view(-16,66)
326 set(gca,...
327     'FontUnits','points',...
328     'FontWeight','normal',...
329     'FontSize',14,...
330     'FontName','Times')
331 ylabel('Sweep',...

```

```

332     'FontUnits','points',...
333     'interpreter','latex',...
334     'FontSize',16,...
335     'FontName','Times')
336 % ylpos = get(get(gca,'ylabel'),'position');
337 % ylpos(1) = ylpos(1)*1.05;
338 % set(get(gca,'ylabel'),'position',ylpos)
339 xlabel('Pixel',...
340     'FontUnits','points',...
341     'interpreter','latex',...
342     'FontSize',16,...
343     'FontName','Times')
344 % xlpas = get(get(gca,'xlabel'),'position');
345 % xlpas(2) = xlpas(2)*1.2;
346 % set(get(gca,'xlabel'),'position',xlpas)
347 ylabel('Sweep Velocity',...
348     'FontUnits','points',...
349     'interpreter','latex',...
350     'FontSize',16,...
351     'FontName','Times')
352 zlpas = get(get(gca,'zlabel'),'position');
353 zlpas(1) = zlpas(1) - 30;
354 zlpas(3) = zlpas(3) + 1;
355 set(get(gca,'zlabel'),'position',zlpas)
356 title('Set 2 (40mm Down)',...
357     'FontUnits','points',...
358     'interpreter','latex',...
359     'FontSize',16,...
360     'FontName','Times')
361
362 subplot(2,2,3)
363 mesh(dataarr{3}.sweepVels)
364 axis([0 2048 0 200 7 13])
365 colormap jet
366 caxis([9 11.5])
367 view(-16,66)
368 set(gca,...
369     'FontUnits','points',...
370     'FontWeight','normal',...
371     'FontSize',14,...

```

```

372     'FontName', 'Times')
373 ylabel('Sweep', ...
374     'FontUnits', 'points', ...
375     'interpreter', 'latex', ...
376     'FontSize', 16, ...
377     'FontName', 'Times')
378 % ylpos = get(get(gca, 'ylabel'), 'position');
379 % ylpos(1) = ylpos(1)*1.05;
380 % set(get(gca, 'ylabel'), 'position', ylpos)
381 xlabel('Pixel', ...
382     'FontUnits', 'points', ...
383     'interpreter', 'latex', ...
384     'FontSize', 16, ...
385     'FontName', 'Times')
386 zlpos = get(get(gca, 'zlabel'), 'position');
387 zlpos(1) = zlpos(1) - 30;
388 zlpos(3) = zlpos(3) + 1;
389 set(get(gca, 'zlabel'), 'position', zlpos)
390 % xlpos = get(get(gca, 'xlabel'), 'position');
391 % xlpos(2) = xlpos(2)*1.2;
392 % set(get(gca, 'xlabel'), 'position', xlpos)
393 zlabel('Sweep Velocity', ...
394     'FontUnits', 'points', ...
395     'interpreter', 'latex', ...
396     'FontSize', 16, ...
397     'FontName', 'Times')
398 title('Set 3 (80mm Down)', ...
399     'FontUnits', 'points', ...
400     'interpreter', 'latex', ...
401     'FontSize', 16, ...
402     'FontName', 'Times')
403
404 subplot(2,2,4)
405 mesh(dataarr{4}.sweepVels)
406 axis([0 2048 0 200 7 13])
407 colormap jet
408 caxis([9 11.5])
409 view(-16,66)
410 set(gca, ...
411     'FontUnits', 'points', ...

```

```

412     'FontWeight', 'normal', ...
413     'FontSize', 14, ...
414     'FontName', 'Times')
415 ylabel('Sweep', ...
416     'FontUnits', 'points', ...
417     'interpreter', 'latex', ...
418     'FontSize', 16, ...
419     'FontName', 'Times')
420 % ylpas = get(get(gca, 'ylabel'), 'position');
421 % ylpas(1) = ylpas(1)*1.05;
422 % set(get(gca, 'ylabel'), 'position', ylpas)
423 xlabel('Pixel', ...
424     'FontUnits', 'points', ...
425     'interpreter', 'latex', ...
426     'FontSize', 16, ...
427     'FontName', 'Times')
428 % xlpas = get(get(gca, 'xlabel'), 'position');
429 % xlpas(2) = xlpas(2)*1.2;
430 % set(get(gca, 'xlabel'), 'position', xlpas)
431 zlabel('Sweep Velocity', ...
432     'FontUnits', 'points', ...
433     'interpreter', 'latex', ...
434     'FontSize', 16, ...
435     'FontName', 'Times')
436 zlpas = get(get(gca, 'zlabel'), 'position');
437 zlpas(1) = zlpas(1) - 30;
438 zlpas(3) = zlpas(3) + 1;
439 set(get(gca, 'zlabel'), 'position', zlpas)
440 title('Set 4 (120mm Down)', ...
441     'FontUnits', 'points', ...
442     'interpreter', 'latex', ...
443     'FontSize', 16, ...
444     'FontName', 'Times')
445
446
447 figure_fname = sprintf("./matlab-figures/mesh-avg-velocity-by-sweep");
448 print(figure_fname, '-dpng', '-r300')
449
450 %
451 % Now for the densities/absorbance

```

```

452 %
453
454 figure
455 set(gcf,...
456     'PaperPositionMode','auto',...
457     'Position',[50 50 1200 1000])
458
459 subplot(2,2,1)
460 mesh(dataarr{1}.sweepAbsorb)
461 colormap jet
462 set(gca,...
463     'FontUnits','points',...
464     'FontWeight','normal',...
465     'FontSize',14,...
466     'FontName','Times')
467 ylabel('Sweep',...
468     'FontUnits','points',...
469     'interpreter','latex',...
470     'FontSize',16,...
471     'FontName','Times')
472 % ylpos = get(get(gca,'ylabel'),'position');
473 % ylpos(1) = ylpos(1)*1.05;
474 % set(get(gca,'ylabel'),'position',ylpos)
475 xlabel('Pixel',...
476     'FontUnits','points',...
477     'interpreter','latex',...
478     'FontSize',16,...
479     'FontName','Times')
480 % xlpso = get(get(gca,'xlabel'),'position');
481 % xlpso(2) = xlpso(2)*1.2;
482 % set(get(gca,'xlabel'),'position',xlpso)
483 zlabel('Absorbance',...
484     'FontUnits','points',...
485     'interpreter','latex',...
486     'FontSize',16,...
487     'FontName','Times')
488 title('Set 1 (Top of Plume)',...
489     'FontUnits','points',...
490     'interpreter','latex',...
491     'FontSize',16,...

```

```

492     'FontName', 'Times')
493
494 subplot(2,2,2)
495 mesh(dataarr{2}.sweepAbsorb)
496 colormap jet
497 set(gca,...
498     'FontUnits','points',...
499     'FontWeight','normal',...
500     'FontSize',14,...
501     'FontName','Times')
502 ylabel('Sweep',...
503     'FontUnits','points',...
504     'interpreter','latex',...
505     'FontSize',16,...
506     'FontName','Times')
507 % ypos = get(get(gca,'ylabel'),'position');
508 % ypos(1) = ypos(1)*1.05;
509 % set(get(gca,'ylabel'),'position',ypos)
510 xlabel('Pixel',...
511     'FontUnits','points',...
512     'interpreter','latex',...
513     'FontSize',16,...
514     'FontName','Times')
515 % xpos = get(get(gca,'xlabel'),'position');
516 % xpos(2) = xpos(2)*1.2;
517 % set(get(gca,'xlabel'),'position',xpos)
518 zlabel('Absorbance',...
519     'FontUnits','points',...
520     'interpreter','latex',...
521     'FontSize',16,...
522     'FontName','Times')
523 title('Set 2 (40mm Down)',...
524     'FontUnits','points',...
525     'interpreter','latex',...
526     'FontSize',16,...
527     'FontName','Times')
528
529 subplot(2,2,3)
530 mesh(dataarr{3}.sweepAbsorb)
531 colormap jet

```

```

532 set(gca,...
533     'FontUnits','points',...
534     'FontWeight','normal',...
535     'FontSize',14,...
536     'FontName','Times')
537 ylabel('Sweep',...
538     'FontUnits','points',...
539     'interpreter','latex',...
540     'FontSize',16,...
541     'FontName','Times')
542 % ylpos = get(get(gca,'ylabel'),'position');
543 % ylpos(1) = ylpos(1)*1.05;
544 % set(get(gca,'ylabel'),'position',ylpos)
545 xlabel('Pixel',...
546     'FontUnits','points',...
547     'interpreter','latex',...
548     'FontSize',16,...
549     'FontName','Times')
550 % xlpso = get(get(gca,'xlabel'),'position');
551 % xlpso(2) = xlpso(2)*1.2;
552 % set(get(gca,'xlabel'),'position',xlpso)
553 zlabel('Absorbance',...
554     'FontUnits','points',...
555     'interpreter','latex',...
556     'FontSize',16,...
557     'FontName','Times')
558 title('Set 3 (80mm Down)',...
559     'FontUnits','points',...
560     'interpreter','latex',...
561     'FontSize',16,...
562     'FontName','Times')
563
564 subplot(2,2,4)
565 mesh(dataarr{4}.sweepAbsorb)
566 colormap jet
567 set(gca,...
568     'FontUnits','points',...
569     'FontWeight','normal',...
570     'FontSize',14,...
571     'FontName','Times')

```

```

572 ylabel('Sweep',...
573         'FontUnits','points',...
574         'interpreter','latex',...
575         'FontSize',16,...
576         'FontName','Times')
577 % ypos = get(get(gca,'ylabel'),'position');
578 % ypos(1) = ypos(1)*1.05;
579 % set(get(gca,'ylabel'),'position',ypos)
580 xlabel('Pixel',...
581         'FontUnits','points',...
582         'interpreter','latex',...
583         'FontSize',16,...
584         'FontName','Times')
585 % xpos = get(get(gca,'xlabel'),'position');
586 % xpos(2) = xpos(2)*1.2;
587 % set(get(gca,'xlabel'),'position',xpos)
588 zlabel('Absorbance',...
589         'FontUnits','points',...
590         'interpreter','latex',...
591         'FontSize',16,...
592         'FontName','Times')
593 title('Set 4 (120mm Down)',...
594        'FontUnits','points',...
595        'interpreter','latex',...
596        'FontSize',16,...
597        'FontName','Times')
598
599
600 figure_fname = sprintf("./matlab_figures/mesh-avg-absorbance-by-sweep");
601 print (figure_fname, '-dpng', '-r300')
602
603
604 %
605 % Now the avg velocity, absorbance
606 %
607
608 figure
609 set(gcf,...
610     'PaperPositionMode','auto',...
611     'Position',[50 50 1200 1000])

```

```

612
613 subplot(2,2,1)
614 yyaxis left
615 plot(dataarr{1}.overallResults(1,:), 'linewidth',1.5)
616 set(gca,...
617     'FontUnits','points',...
618     'FontWeight','normal',...
619     'FontSize',14,...
620     'FontName','Times',...
621     'YColor','black')
622 ylabel('Velocity ($km/s$)',...
623     'FontUnits','points',...
624     'interpreter','latex',...
625     'FontSize',16,...
626     'FontName','Times')
627 % ylpos = get(get(gca,'ylabel'),'position');
628 % ylpos(1) = ylpos(1)*1.05;
629 % set(get(gca,'ylabel'),'position',ylpos)
630 axis([-inf inf -1 15])
631
632 hold on
633 yyaxis right
634 plot(dataarr{1}.overallResults(2,:), 'linewidth',1.5)
635 legend('Average velocity','Average absorbance')
636 axis([-inf inf -1 1])
637 set(gca,...
638     'FontUnits','points',...
639     'FontWeight','normal',...
640     'FontSize',14,...
641     'FontName','Times',...
642     'YColor','black')
643 ylabel('Absorbance',...
644     'FontUnits','points',...
645     'interpreter','latex',...
646     'FontSize',16,...
647     'FontName','Times')
648 % ylpos = get(get(gca,'ylabel'),'position');
649 % ylpos(1) = ylpos(1)*1.05;
650 % set(get(gca,'ylabel'),'position',ylpos)
651 xlabel('Pixel',...

```

```

652     'FontUnits','points',...
653     'interpreter','latex',...
654     'FontSize',16,...
655     'FontName','Times')
656 xlpow = get(get(gca,'xlabel'),'position');
657 xlpow(2) = xlpow(2)*1.2;
658 set(get(gca,'xlabel'),'position',xlpow)
659 title('Set 1 (Top of Plume)',...
660     'FontUnits','points',...
661     'interpreter','latex',...
662     'FontSize',16,...
663     'FontName','Times')
664
665 subplot(2,2,2)
666 yyaxis left
667 plot(dataarr{2}.overallResults(1,:), 'linewidth',1.5)
668 set(gca,...
669     'FontUnits','points',...
670     'FontWeight','normal',...
671     'FontSize',14,...
672     'FontName','Times',...
673     'YColor','black')
674 ylabel('Velocity ($km/s$)',...
675     'FontUnits','points',...
676     'interpreter','latex',...
677     'FontSize',16,...
678     'FontName','Times')
679 % ylpow = get(get(gca,'ylabel'),'position');
680 % ylpow(1) = ylpow(1)*1.05;
681 % set(get(gca,'ylabel'),'position',ylpow)
682 axis([-inf inf -1 15])
683 hold on
684
685 yyaxis right
686 plot(dataarr{2}.overallResults(2,:), 'linewidth',1.5)
687 legend('Average velocity','Average absorbance')
688 axis([-inf inf -1 1])
689 set(gca,...
690     'FontUnits','points',...
691     'FontWeight','normal',...

```

```

692     'FontSize',14,...
693     'FontName','Times',...
694     'YColor','black')
695 ylabel('Absorbance',...
696     'FontUnits','points',...
697     'interpreter','latex',...
698     'FontSize',16,...
699     'FontName','Times')
700 % ylpas = get(get(gca,'ylabel'),'position');
701 % ylpas(1) = ylpas(1)*1.05;
702 % set(get(gca,'ylabel'),'position',ylpas)
703 xlabel('Pixel',...
704     'FontUnits','points',...
705     'interpreter','latex',...
706     'FontSize',16,...
707     'FontName','Times')
708 xlpas = get(get(gca,'xlabel'),'position');
709 xlpas(2) = xlpas(2)*1.2;
710 set(get(gca,'xlabel'),'position',xlpas)
711 title('Set 2 (40mm Down)',...
712     'FontUnits','points',...
713     'interpreter','latex',...
714     'FontSize',16,...
715     'FontName','Times')
716
717 subplot(2,2,3)
718 yyaxis left
719 plot(dataarr{3}.overallResults(1,:), 'linewidth',1.5)
720 set(gca,...
721     'FontUnits','points',...
722     'FontWeight','normal',...
723     'FontSize',14,...
724     'FontName','Times',...
725     'YColor','black')
726 ylabel('Velocity ($km/s$)',...
727     'FontUnits','points',...
728     'interpreter','latex',...
729     'FontSize',16,...
730     'FontName','Times')
731 % ylpas = get(get(gca,'ylabel'),'position');

```

```

732 % ylpos(1) = ylpos(1)*1.05;
733 % set(get(gca,'ylabel'),'position',ylpos)
734 axis([-inf inf -1 15])
735 hold on
736
737 yyaxis right
738 plot(dataarr{3}.overallResults(2,:), 'linewidth',1.5)
739 legend('Average velocity','Average absorbance')
740 axis([-inf inf -1 1])
741 set(gca,...
742     'FontUnits','points',...
743     'FontWeight','normal',...
744     'FontSize',14,...
745     'FontName','Times',...
746     'YColor','black')
747 ylabel('Absorbance',...
748     'FontUnits','points',...
749     'interpreter','latex',...
750     'FontSize',16,...
751     'FontName','Times')
752 % ylpos = get(get(gca,'ylabel'),'position');
753 % ylpos(1) = ylpos(1)*1.05;
754 % set(get(gca,'ylabel'),'position',ylpos)
755 xlabel('Pixel',...
756     'FontUnits','points',...
757     'interpreter','latex',...
758     'FontSize',16,...
759     'FontName','Times')
760 xlpas = get(get(gca,'xlabel'),'position');
761 xlpas(2) = xlpas(2)*1.2;
762 set(get(gca,'xlabel'),'position',xlpas)
763 title('Set 3 (80mm Down)',...
764     'FontUnits','points',...
765     'interpreter','latex',...
766     'FontSize',16,...
767     'FontName','Times')
768
769 subplot(2,2,4)
770 yyaxis left
771 plot(dataarr{4}.overallResults(1,:), 'linewidth',1.5)

```

```

772 set(gca,...
773     'FontUnits','points',...
774     'FontWeight','normal',...
775     'FontSize',14,...
776     'FontName','Times',...
777     'YColor','black')
778 ylabel('Velocity ($km/s$)',...
779     'FontUnits','points',...
780     'interpreter','latex',...
781     'FontSize',16,...
782     'FontName','Times')
783 % ylpos = get(get(gca,'ylabel'),'position');
784 % ylpos(1) = ylpos(1)*1.05;
785 % set(get(gca,'ylabel'),'position',ylpos)
786 axis([-inf inf -1 15])
787 hold on
788 yyaxis right
789 plot(dataarr{4}.overallResults(2,:), 'linewidth',1.5)
790 legend('Average velocity','Average absorbance')
791 axis([-inf inf -1 1])
792 set(gca,...
793     'FontUnits','points',...
794     'FontWeight','normal',...
795     'FontSize',14,...
796     'FontName','Times',...
797     'YColor','black')
798 ylabel('Absorbance',...
799     'FontUnits','points',...
800     'interpreter','latex',...
801     'FontSize',16,...
802     'FontName','Times')
803 % ylpos = get(get(gca,'ylabel'),'position');
804 % ylpos(1) = ylpos(1)*1.05;
805 % set(get(gca,'ylabel'),'position',ylpos)
806 xlabel('Pixel',...
807     'FontUnits','points',...
808     'interpreter','latex',...
809     'FontSize',16,...
810     'FontName','Times')
811 xlpos = get(get(gca,'xlabel'),'position');

```

```

812 xlpow(2) = xlpow(2)*1.2;
813 set(get(gca,'xlabel'),'position',xlpow)
814 title('Set 4 (120mm Down)',...
815       'FontUnits','points',...
816       'interpreter','latex',...
817       'FontSize',16,...
818       'FontName','Times')
819
820 figure_fname = sprintf("./matlab_figures/avg-vel-absorbance-by-pixel");
821 print(figure_fname, '-dpng', '-r300')
822
823 % Now get all absorbances on one plot and all vels on another
824 % Velocities
825
826 figure
827
828 plot(dataarr{1}.overallResults(1,:), 'linewidth',1.5)
829 hold on
830 plot(dataarr{2}.overallResults(1,:), 'linewidth',1.5)
831 plot(dataarr{3}.overallResults(1,:), 'linewidth',1.5)
832 plot(dataarr{4}.overallResults(1,:), 'linewidth',1.5)
833
834 axis([-inf inf 0 15])
835 % ylabel('1 - I/I0')
836 % xlabel('km/s')
837 % legend('Normal Filtering','Zero-Phase Filtering')
838 % set(gcf,'PaperPositionMode','auto')
839 %      'Position',[25 25 750 450],...
840 % set(gcf,'Position',[10 50 800 500])
841 set(gcf,...
842      'PaperPositionMode','auto',...
843      'Position',[500 500 800 500])
844 set(gca,...
845      'Units','normalized',...
846      'Position',[.12 .18 .8 .73],...
847      'FontUnits','points',...
848      'FontWeight','normal',...
849      'FontSize',14,...
850      'FontName','Times')
851 ylabel({'$km/s$'},...

```

```

852     'FontUnits','points',...
853     'interpreter','latex',...
854     'FontSize',20,...
855     'FontName','Times')
856 ylpos = get(get(gca,'ylabel'),'position');
857 ylpos(1) = ylpos(1)*1.25;
858 set(get(gca,'ylabel'),'position',ylpos)
859 xlabel('Pixel',...
860     'FontUnits','points',...
861     'interpreter','latex',...
862     'FontSize',20,...
863     'FontName','Times')
864 xlpos = get(get(gca,'xlabel'),'position');
865 xlpos(2) = xlpos(2)*1.2;
866 set(get(gca,'xlabel'),'position',xlpos)
867 legend({'Set 1 (Top of Plume)',...
868     'Set 2 (40mm Down)',...
869     'Set 3 (80mm Down)',...
870     'Set 4 (120mm Down)'},...
871     'FontUnits','points',...
872     'FontWeight','normal',...
873     'FontSize',14,...
874     'FontName','Times',...
875     'Location','SouthWest')
876
877 figure_fname = sprintf("./matlab_figures/avg-velocity-by-pixel");
878 print(figure_fname, '-dpng', '-r300')
879
880 % Absorbances
881
882 figure
883 plot(dataarr{1}.overallResults(2,:), 'linewidth',1.5)
884 hold on
885 plot(dataarr{2}.overallResults(2,:), 'linewidth',1.5)
886 plot(dataarr{3}.overallResults(2,:), 'linewidth',1.5)
887 plot(dataarr{4}.overallResults(2,:), 'linewidth',1.5)
888
889 axis([-inf inf -1 1])
890 % ylabel('1 - I/I0')
891 % xlabel('km/s')

```

```

892 % legend('Normal Filtering','Zero-Phase Filtering')
893 % set(gcf,'PaperPositionMode','auto')
894 %      'Position',[25 25 750 450],...
895 % set(gcf,'Position',[10 50 800 500])
896 set(gcf,...
897     'PaperPositionMode','auto',...
898     'Position',[500 500 800 500])
899 set(gca,...
900     'Units','normalized',...
901     'Position',[.12 .18 .8 .73],...
902     'FontUnits','points',...
903     'FontWeight','normal',...
904     'FontSize',14,...
905     'FontName','Times')
906 ylabel({'Absorbance'},...
907     'FontUnits','points',...
908     'interpreter','latex',...
909     'FontSize',20,...
910     'FontName','Times')
911 ylpos = get(get(gca,'ylabel'),'position');
912 ylpos(1) = ylpos(1)*1.05;
913 set(get(gca,'ylabel'),'position',ylpos)
914 xlabel('Pixel',...
915     'FontUnits','points',...
916     'interpreter','latex',...
917     'FontSize',20,...
918     'FontName','Times')
919 xlpos = get(get(gca,'xlabel'),'position');
920 xlpos(2) = xlpos(2)*1.05;
921 set(get(gca,'xlabel'),'position',xlpos)
922 legend({'Set 1 (Top of Plume)',...
923     'Set 2 (40mm Down)',...
924     'Set 3 (80mm Down)',...
925     'Set 4 (120mm Down)'} ,...
926     'FontUnits','points',...
927     'FontWeight','normal',...
928     'FontSize',14,...
929     'FontName','Times',...
930     'Location','SouthWest')
931

```

```

932 figure_fname = sprintf("./matlab-figures/avg-absorbance-by-pixel");
933 print (figure_fname, '-dpng', '-r300')
934
935
936 %% Plots showing averaged camera ref (mesh) and avg photodiode ref
937 close all;
938
939 % Camera ref
940 mesh(avgRef)
941 colormap jet
942 axis tight
943 view(-12,21)
944
945 set(gcf,...
946     'PaperPositionMode','auto',...
947     'Position',[400 400 700 600])
948
949 set(gca,...
950     'FontUnits','points',...
951     'FontWeight','normal',...
952     'FontSize',16,...
953     'FontName','Times')
954 ylabel('Sample',...
955     'FontUnits','points',...
956     'interpreter','latex',...
957     'FontSize',20,...
958     'FontName','Times')
959 % ylpas = get(get(gca,'ylabel'),'position');
960 % ylpas(1) = ylpas(1)*1.05;
961 % set(get(gca,'ylabel'),'position',ylpas)
962 set(get(gca,'ylabel'),'position',[-192.25,95,-5])
963 xlabel('Pixel',...
964     'FontUnits','points',...
965     'interpreter','latex',...
966     'FontSize',20,...
967     'FontName','Times')
968 % xlpas = get(get(gca,'xlabel'),'position');
969 % xlpas(2) = xlpas(2)*1.2;
970 % set(get(gca,'xlabel'),'position',xlpas)
971 zlabel('Intensity (8-bit)',...

```

```

972     'FontUnits','points',...
973     'interpreter','latex',...
974     'FontSize',20,...
975     'FontName','Times')
976
977 figure_fname = sprintf("./matlab-figures/avg-camera-ref");
978 print (figure_fname, '-dpng', '-r300')
979
980
981 % DAQ ref
982 figure
983 plot (avgLampRef, 'k', 'linewidth',1.5)
984
985 %axis tight
986 % ylabel('I - I/I0')
987 % xlabel('km/s')
988 % legend('Normal Filtering','Zero-Phase Filtering')
989 % set(gcf,'PaperPositionMode','auto')
990 %     'Position',[25 25 750 450],...
991 % set(gcf,'Position',[10 50 800 500])
992 set(gcf,...
993     'PaperPositionMode','auto',...
994     'Position',[400 400 700 600])
995 set(gca,...
996     'Units','normalized',...
997     'Position',[.15 .12 .8 .82],...
998     'FontUnits','points',...
999     'FontWeight','normal',...
1000    'FontSize',16,...
1001    'FontName','Times')
1002 ylabel({'Intensity (Volts)'} ,...
1003     'FontUnits','points',...
1004     'interpreter','latex',...
1005     'FontSize',20,...
1006     'FontName','Times')
1007 % ypos = get(get(gca,'ylabel'),'position');
1008 % ypos(1) = ypos(1)*1.05;
1009 % set(get(gca,'ylabel'),'position',ypos)
1010 xlabel('Sample',...
1011     'FontUnits','points',...

```

```

1012     'interpreter','latex',...
1013     'FontSize',20,...
1014     'FontName','Times')
1015 % xlpow = get(get(gca,'xlabel'),'position');
1016 % xlpow(2) = xlpow(2)*1.05;
1017 % set(get(gca,'xlabel'),'position',xlpow)
1018
1019 figure_fname = sprintf("./matlab_figures/avg-daq-ref");
1020 print(figure_fname, '-dpng', '-r300')
1021
1022 % Actual DAQ data (showing how the hollow cathode lamp seems to have
1023 % overpowered the laser)
1024
1025 figure
1026 plot(dataarr{1}.avgDAQ, 'k', 'linewidth',1.5)
1027
1028 %axis tight
1029 % ylabel('I - I/I0')
1030 % xlabel('km/s')
1031 % legend('Normal Filtering','Zero-Phase Filtering')
1032 % set(gcf,'PaperPositionMode','auto')
1033 %     'Position',[25 25 750 450],...
1034 % set(gcf,'Position',[10 50 800 500])
1035 set(gcf,...
1036     'PaperPositionMode','auto',...
1037     'Position',[400 400 700 600])
1038 set(gca,...
1039     'Units','normalized',...
1040     'Position',[.15 .12 .8 .82],...
1041     'FontUnits','points',...
1042     'FontWeight','normal',...
1043     'FontSize',16,...
1044     'FontName','Times')
1045 ylabel({'Intensity (Volts)'} ,...
1046     'FontUnits','points',...
1047     'interpreter','latex',...
1048     'FontSize',20,...
1049     'FontName','Times')
1050 % ylpow = get(get(gca,'ylabel'),'position');
1051 % ylpow(1) = ylpow(1)*1.05;

```

```

1052 % set(get(gca,'ylabel'),'position',ylpos)
1053 xlabel('Sample',...
1054         'FontUnits','points',...
1055         'interpreter','latex',...
1056         'FontSize',20,...
1057         'FontName','Times')
1058 % xlpow = get(get(gca,'xlabel'),'position');
1059 % xlpow(2) = xlpow(2)*1.05;
1060 % set(get(gca,'xlabel'),'position',xlpow)
1061
1062 figure_fname = sprintf("./matlab_figures/avg-daq-actual-set1");
1063 print(figure_fname,'-dpng','-r300')

```

Bibliography

1. Bell, W. E., "United States Patent 3,893,771," 1975.
2. Matsui, M., Auweter-Kurtz, M., Herdrich, G., and Komurasaki, K., "Enthalpy Measurement in Inductively Heated Plasma Generator Flow by Laser Absorption Spectroscopy," *AIAA Journal*, Vol. 43, No. 9, 2005, pp. 2060–2064.
3. Yokota, S., Matsui, M., Sako, D., Yamamoto, N., Koizumi, H., and Komurasaki, K., "Diagnostics of Hall Thruster Plume by Laser Absorption Spectroscopy," *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Sacramento, California, July 9-12, 2006*, AIAA-2006-5028.
4. Kinefuchi, K., Funaki, I., and Toki, K., "Laser absorption velocimetry of plasma flow in two-dimensional magnetoplasmadynamic arcjet," *Journal of Propulsion and Power*, Vol. 22, No. 5, 2006, pp. 1085–1090.
5. Matsui, M., Yokota, S., Sako, D., Komurasaki, K., and Y, A., "Number Density Distributions of Xenon Atom in Hall Thruster Plumes," *43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cincinnati, OH, July 8-11, 2007*, AIAA-2007-5306.
6. Lee, D., *Velocity Plume Profiles for Hall Thrusters Using Laser Diagnostic*, Master's thesis, Air Force Institute of Technology, 2010.
7. Lary, E. C., Meyerland, R. G., and Salz, F., "Ion Acceleration in a Gyro-Dominated Neutral Plasma," *Bulletin of the American Physical Society*, 1962, p. 441.
8. Seikel, G. R. and Reshotko, E., "Hall Current Ion Accelerator," *Bulletin of the American Physical Society*, 1962, p. 414.
9. Janes, G. S. and Lowder, R. S., "Anomalous electron diffusion and ion acceleration in a low-density plasma," *Physics of Fluids*, Vol. 9, No. 6, 1966, pp. 1115–1123.
10. Choueiri, E. Y., "Plasma oscillations in Hall thrusters," *Physics of Plasmas*, Vol. 8, No. 4, 2001, pp. 1411–1426.
11. Beal, B., "The Air Force Research Laboratory's In-Space Propulsion Program," Tech. rep., Air Force Research Laboratory, 2015.
12. Sutton, G. P. and Biblarz, O., *Rocket Propulsion Elements*, John Wiley & Sons, Hoboken, 9th ed., 2017.
13. Taylor, B. N. and Thompson, A., "The International System of Units," Tech. rep., National Institute of Standards and Technology, Gaithersburg, 2008.

14. Goebel, D. M. and Katz, I., "Fundamentals of Electric Propulsion: Ion and Hall Thrusters," Tech. rep., Jet Propulsion Laboratory, 2008.
15. Chabay, R. and Sherwood, B., *Matter & Interactions II: Electric and Magnetic Interactions*, John Wiley & Sons, 3rd ed., 2011.
16. Hargus, W. A., *Investigation of the Plasma Acceleration Mechanism Within a Coaxial Hall Thruster*, Doctorate, Stanford University, 2001.
17. Yoshikawa, S. and Rose, D. J., "Anomalous Diffusion of a Plasma across a Magnetic Field," *The Physics of Fluids*, Vol. 5, No. 3, 1962, pp. 334–340.
18. Tilinin, G. N., "High-frequency plasma waves in a Hall accelerator with an extended acceleration zone," *Soviet Physics: Technical Physics*, Vol. 22, No. 8, 1977, pp. 974–978.
19. Esipchuk, Y. B., Morozov, A. I., Tilinin, G. N., and Trofimov, A. V., "Plasma oscillations in closed-drift accelerators with an extended acceleration zone," *Soviet Physics: Technical Physics*, Vol. 18, No. 7, 1974, pp. 928–932.
20. Hofer, R. R., Johnson, L. K., Goebel, D. M., and Fitzgerald, D. J., "Effects of an Internally-Mounted Cathode on Hall Thruster Plume Properties," *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, AIAA-2006-4482.
21. Chesta, E., Lam, C. M., Meezan, N. B., Schmidt, D. P., and Cappelli, M. A., "A Characterization of Plasma Fluctuations within a Hall Discharge," *IEEE Transactions on Plasma Science*, Vol. 29, No. 4, 2001, pp. 582–591.
22. Haken, H., *Laser Theory*, Springer-Verlag, 1984.
23. Demtröder, W., *Laser Spectroscopy: Basic Concepts and Instrumentation*, Springer-Verlag, 3rd ed., 2003.
24. Werle, P., Mücke, R., and Slemr, F., "The limits of signal averaging in atmospheric trace-gas monitoring by tunable diode-laser absorption spectroscopy (TD-LAS)," *Applied Physics B Photophysics and Laser Chemistry*, Vol. 57, No. 2, 1993, pp. 131–139.
25. Swinehart, D. F., "The Beer-Lambert Law," *Journal of Chemical Education*, Vol. 39, No. 7, 1962, pp. 333.
26. Cedolin, R. J., Hargus, W. A., Storm, P. V., Hanson, R. K., and Cappelli, M. A., "Laser-induced fluorescence study of a xenon Hall thruster," *Applied Physics B: Lasers and Optics*, Vol. 65, 1997, pp. 459–469.
27. Hargus, W. A. and Cappelli, M. A., "Laser-induced fluorescence measurements of velocity within a Hall discharge," *Applied Physics B: Lasers and Optics*, Vol. 72, 2001, pp. 961–969.

28. Hargus, W. and Charles, C., "Near Plume laser Induced Fluorescence Velocity Measurements of a 600 W Hall Thruster," *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Hartford, CT, July 21-23, 2008*, AIAA-2008-5004.
29. Hiller, B., McDaniel, J. C., Rea Jr., E. C., and Hanson, R. K., "Laser-Induced Fluorescence Technique for Velocity Field Measurements in Subsonic Flows." *Optics Letters*, Vol. 8, No. 9, 1983, pp. 126-127.
30. Hiller, B. and Hanson, R. K., "Two-frequency laser-induced fluorescence technique for rapid velocity-field measurements in gas flows," *Optics Letters*, Vol. 10, No. 5, 1985, pp. 206-208.
31. Hiller, B. and Hanson, R. K., "Simultaneous planar measurements of velocity and pressure fields in gas flows using laser-induced fluorescence." *Applied optics*, Vol. 27, No. 1, 1988, pp. 33-48.
32. Bui, D. M., *Plume Characterization of Busek 600W Hall Thruster*, Master's thesis, Air Force Institute of Technology, 2012.
33. Kurt J Lesker Company, "300 Series Convection Vacuum Gauge Module User Manual."
34. Extorr Inc., "Instruction Manual: Extorr XT Series RGA," 2009.
35. Busek Co. Inc., "BHT-600 Busek Hall Effect Thruster Operating and Maintenance Manual," 2010.
36. Busek Co. Inc., "Busek Hollow Cathodes," Tech. rep., 2013.
37. Basler AG, "Basler raL2048-80km Camera Specification," Tech. rep.
38. Saloman, E. B., "Energy levels and observed spectral lines of Xenon, XeI through Xe LIV," *Journal of Physical and Chemical Reference Data*, Vol. 33, No. 3, 2004, pp. 765-921.
39. Sansonetti, J. E. and Martin, W. C., "Handbook of Basic Atomic Spectroscopic Data," *Journal of Physical and Chemical Reference Data*, Vol. 34, No. 4, 2005, pp. 1559-2259.

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From — To)		
07-03-2019		Master's Thesis		Sept 2017 — Mar 2019		
4. TITLE AND SUBTITLE SHEET VELOCITY MEASUREMENTS USING LASER ABSORPTION SPECTROSCOPY IN A XENON HALL EFFECT THRUSTER PLUME				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Avery W. Leonard, 2d Lt, USAF				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENY-MS-19-M-227		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research Attn: Dr. Mitat Birkan 875 N. Randolph Street, Suite 325 Arlington, VA 22203 mitat.birkan@us.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.						
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.						
14. ABSTRACT A new laser absorption spectroscopy (LAS) velocimetry system, designed to obtain 2D planar velocity fields for ionized Xenon in the plume of a Hall effect thruster by probing the transition of Xe II at 834.72 nm, was implemented at the Air Force Institute of Technology (AFIT) Space Propulsion Analysis and System Simulator (SPASS) Lab vacuum chamber. A single horizontal laser sheet was used to probe singly-ionized Xenon in the plume of a Busek BHT-600 Hall thruster and obtain a histogram of axial velocity, to validate the system. Similar velocities to those obtained by an earlier intrusive characterization of the thruster were observed.						
15. SUBJECT TERMS Electric Propulsion, Hall thruster, Electrostatic, LAS, Laser Spectroscopy						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Carl Hartsfield, AFIT/ENY	
U	U	U	U	193	19b. TELEPHONE NUMBER (include area code) (937) 255-3636, x4667; carl.hartsfield@afit.edu	