EMBRY-RIDDLE
Aeronautical University™
SCHOLARLY COMMONS

8-2018

A Study into Data Analysis of Varying Types of Langmuir Probes

William Merritt

A STUDY INTO DATA ANALYSIS OF VARYING TYPES OF LANGMUIR PROBES

BY

WILLIAM MERRITT

A Thesis

Submitted to the Department of Physical Sciences

In partial fulfillment of the requirements

for the degree of

Master of Science in Engineering Physics

08/2018

Embry-Riddle Aeronautical University

Daytona Beach, Florida

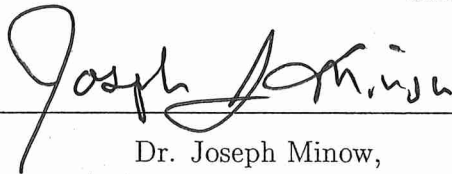# A STUDY INTO DATA ANALYSIS OF VARYING TYPES OF LANGMUIR PROBES

by

William Merritt

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Aroh Barjatya, Associate Professor of Engineering Physics, Daytona Beach Campus, and Thesis Committee Members Dr. Joseph Minow, NASA Technical Fellow for Space Environments, and Dr. Bereket Berhane, Associate Professor, Daytona Beach Campus, and has been approved by the Thesis Committee. It was submitted to the Department of Physical Sciences in partial fulfillment of the requirements of the degree of
Master of Science in Engineering Physics

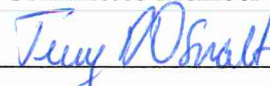THESIS COMMITTEE:

Dr. Aroh Barjatya,
Committee Chair

Dr. Joseph Minow,
Committee Member

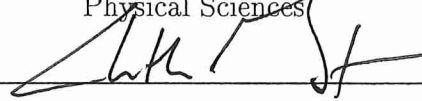Dr. Bereket Berhane,
Committee Member

Dr. Matthew Zettergren,
Graduate Program Chair,
Engineering Physics

Dr. Terry Oswalt,
Department Chair,
Physical Sciences

Dr. Karen Gaines,
Dean, College of Arts and Sciences

Dr. Christopher Grant,
Vice Provost of Academic Support

iii

Abstract

Langmuir probes are ubiquitously used for in-situ measurements of plasma parameters. These probes have been placed on many different platforms, including experimental sounding rockets for measurements in mesosphere-lower-thermosphere, and also onboard satellites to obtain data sets over an extended period of time in the ionosphere. To accommodate such different situations, many different variations of the Langmuir probe design have been made. This thesis covers two such implementations, as well as the data analysis and issues that can arise with such instruments. The first of these implementations is a set of sweeping Langmuir probes on the Floating Potential Measurement Unit (FPMU) that is deployed on-board the International Space Station. We compare the output of NASA's current data processing algorithm for FPMU to that of our own algorithm. This work shows how instruments degrade overtime, and how data analysis can partially work around such degradation. Our analysis also demonstrated how various environmental effects need to be accounted for to get an accurate measurement during data analysis of Langmuir probes. The second implementation considered in this thesis is a new multi-needle Langmuir probe (mNLP) design as recently flown aboard some German sounding rockets. Our work confirms that mNLP instrument shows great promise, but also cautions in its data processing algorithms which can easily lead to 50% errors unless appropriately dealt with. We then present a new way to analyze mNLP data that can bring the measurement error to within 10%.

Acknowledgments

I would like to thank Dr. Aroh Barjatya for being my adviser throughout my research. I also thank my family who have been with me through this journey. I would also like to thank my friends who have offered encouragement during this time as well. I would also like to thank my God, without whom none of this would be possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the simplest sense Langmuir probes are electrometers. A voltage bias is applied to an electrode immersed in plasma and the resulting collection current is measured. The collection current is dependent on many different variables, including plasma density, plasma temperature, and probe geometry. The resulting current-voltage (IV) curve can thus be analyzed to determine physical parameters such as electron and ion density and electron temperature, as well as spacecraft floating potential [Mott-Smith and Langmuir, 1926]. A typical IV curve observed by a Langmuir probe in a plasma with a Maxwellian velocity distribution is shown in Figure 1.1. Note that the ion current has been exaggerated by an order of magnitude to ease the viewing of the plot. Further note that the current "from" the probe to the plasma (i.e. electron collection current) is considered positive. When the primary collection current is electron, that region is called the electron saturation current. Likewise, when the primary collection current is ion, that region is called the ion saturation current. The middle region, where the electrons are repelled due to negative bias, and yet the electron collection current is larger than the ion collection current is called the electron retardation region. The floating potential is the voltage at which the net current from electrons and ions is equal, and the plasma potential is the voltage when no fields exist between the electrode and the plasma and all thermal ions and electrons are collected. The current collection equations are stated in the next sub-section followed by a brief introduction to various implementations of Langmuir probes [Barjatya, 2007].
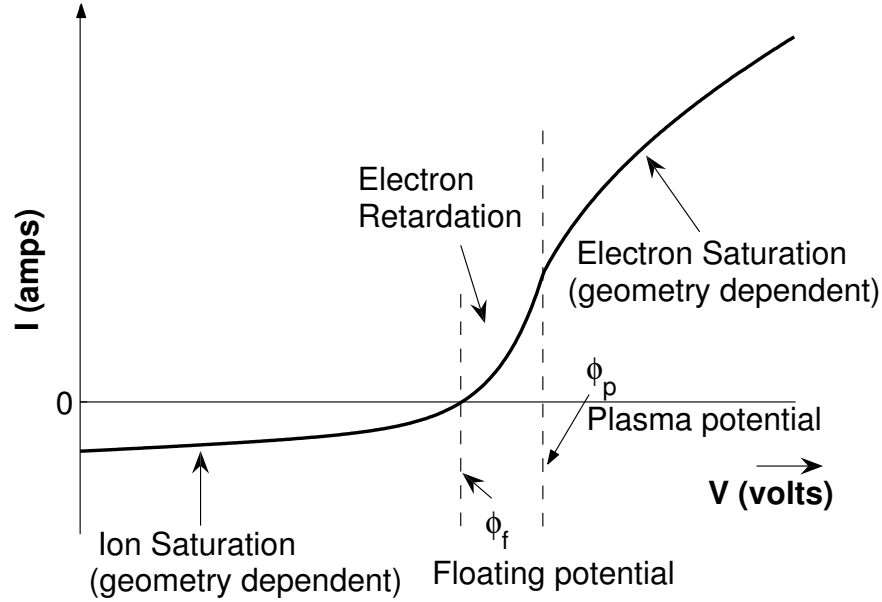
Figure 1.1: A typical I-V curve

1.1   Background Langmuir Probe Theory

When a surface is immersed in plasma, the surface collects currents related to the thermal velocity of a particle j given by equation 1.1.

$$v_{th_j} = \sqrt{\frac{k_B T_j}{m_j}}$$

(1.1)

where k_B is Boltzmann constant, m_j is the particle mass, and T_j is the particle temperature. Thermal velocity of ions is significantly less than that of electrons, since even the smallest ions (atomic hydrogen) are around two thousand times more massive than an electron. The direction of thermal motion is random. The thermal velocity equations govern how quickly the particles are moving around in the plasma, and thus are important in determining how much current is gathered by the probe. Given a certain temperature and density of the plasma, an equation for random thermal current can be derived. Equation 1.2 describes the current collected when

there is no net bias on the probe surface, i.e. at plasma potential.

$$I_{th_j} = n_j q_j A \sqrt{\frac{k_B T_j}{2\pi m_j}} \tag{1.2}$$

where n_j is the species (j) density, q_j is the species charge, and A is the probe current collection area. When a voltage bias is applied to a surface in the plasma, particles of the opposite charge of the sign of the potential will be attracted to the surface, while the particles of the same charge sign as the voltage bias will be repulsed. The amount of particles per unit time, and thus the collection current, is highly dependent on the bias and geometry of the probe and a modified set of current collection equations are derived for a plasma with maxwellian distribution of particles. The current expressions have two possibilities: space charge limited and orbital motion limited [Chen, 1965]. The space charge limited current collection expressions apply when the probe dimensions are much larger than debye length in the surrounding plasma. This situation is rarely encountered in space plasmas and is typically seen in high plasma density chambers. The Orbital Motion Limited (OML) theory applies when the probe dimensions are smaller than few times debye length in the surrounding plasma. Equation 1.3 describes the OML current expressions that govern how current of different species (i.e. electrons and ions) behave in an ideal case in their respective saturation regions.

$$I_j = I_{th_j} \left(1 + \frac{q_j(\phi - \phi_p)}{k_B T_j}\right)^\beta \tag{1.3}$$

where ϕ is applied probe potential relative to spacecraft chassis and ϕ_p is the plasma potential, and the value of β varies according to probe geometry:

$$\beta = 1 \quad \text{(spherical)} \tag{1.4}$$

$$\beta = \frac{1}{2} \quad \text{(cylindrical)} \tag{1.5}$$

$$\beta = 0 \quad \text{(planar)} \tag{1.6}$$
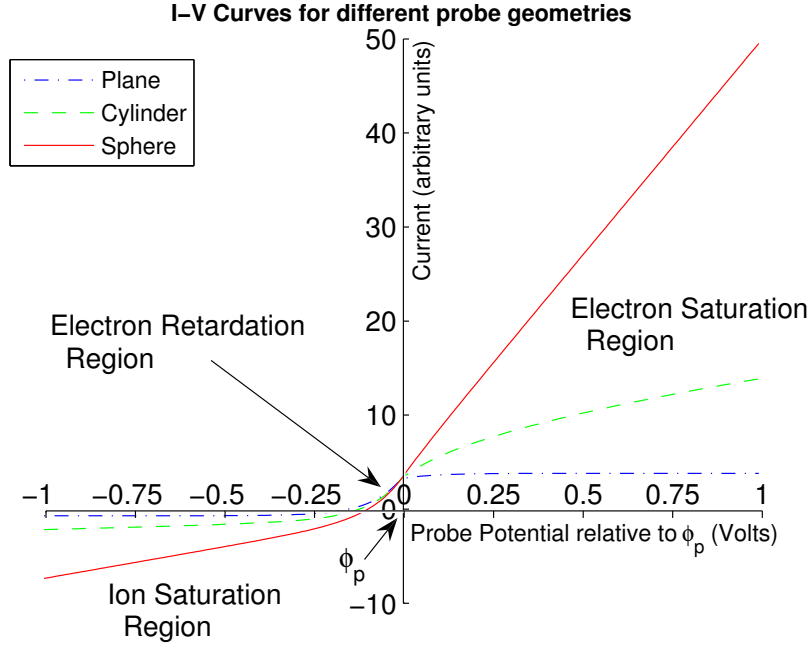
Figure 1.2: Typical I-V curves for different probe geometries

The above presented equations however are only for saturation regions. Since these currents are inversely proportional to the square root of the particle's mass, and ion mass is much larger than electron mass, the electron saturation region has a higher magnitude than the ion saturation region. The IV curves of different probe geometries is shown in figure 1.2. Note the unique feature of a planar probe: irrespective of the applied potential in the saturation region the current maintains a maximum value.

Besides the two saturation regions, the third region in any IV curve is the electron retardation region, in which although the surface is negatively biased, the current collection is still dominated by electrons. This region exists because of the much higher thermal velocity of electrons (as mentioned above), and thus needs a highly negative bias to force them away. Since this term is directly related to temperature working against bias, its shape is mainly determined by the temperature of the plasma, and is exponential in geometry.

$$I_e = I_{th_e} \exp\left(\frac{\phi - \phi_p}{k_B T_e}\right) \tag{1.7}$$

The theory however is seldom simple to apply in non-ideal cases. Many of these are due to physical constraints such as probe size and uniform surface work function, while others might be more based upon the environment the probe is in such as mesothermal plasma, existing magnetic field effects, secondary electron emission current from energetic particles, and photoelectron emission current in the presence of sunlight. Each of these situations are dealt on a case by case basis [Brace, 1998] as there is no unified set of equations that predict Langmuir probe behavior in all conditions.

1.2   Langmuir Probe Implementations

A swept bias Langmuir probe is the traditional implementation of the technique and is very commonly used in space measurements. There are several limitations and challenges to this implementation though. For instance, the entire IV curve results only in one measurement of plasma density and temperature. This limits the instrument to a fairly slow measurement cadence, and consequently a low spatial resolution when the instrument is deployed on a moving platform such as a satellite or rocket. Additionally, all probes have surface contamination and surface work function non-uniformities that manifests itself as a RC filter on the current collection [Piel et al., 2001]. This results in hysteresis in the IV curves between upsweep and downsweep. An example of this is shown in Figure 1.3. When hysteresis is present only electron temperature derived from the upsweep is reliable, as on downsweeps the hysteresis distorts the structure of the electron retardation region [Hirt et al., 2001] . There are ways to circumvent contamination effects such as heating the probe surface which boils of the contamination [Amatucci et al., 1993], or sweeping the probe fast enough (> 25 Hz) that the RC filter is shorted [Oyama et al., 2012].

Another most important side effect of swept bias Langmuir probes is the impact it has on spacecraft floating potential. Note that the electrical ground of probe electronics is the spacecraft chassis. The probe and the spacecraft both immersed in plasma environment form a closed loop circuit. So when a probe is biased deep in electron saturation region and is collecting significant electron collection current,
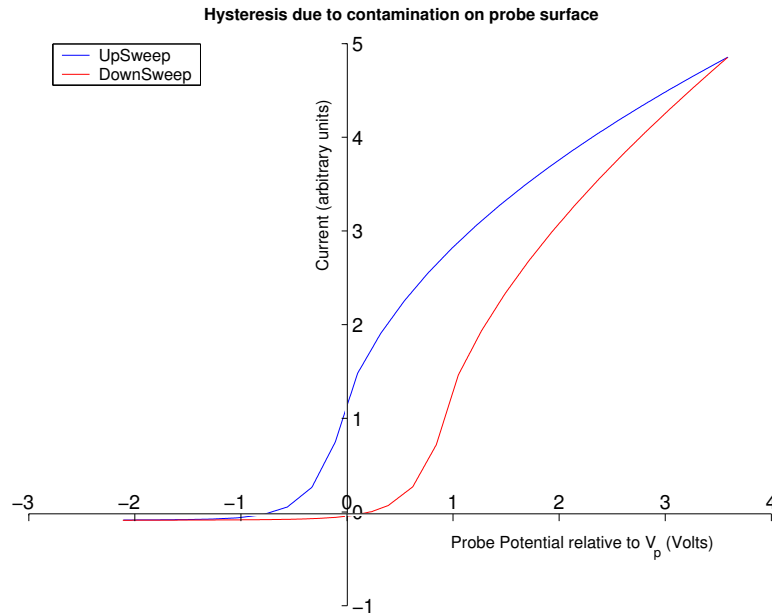
Figure 1.3: An upsweep and downsweep showing hysteresis in consecutive I-V curves

the spacecraft surface has to balance it in two different ways. Either the spacecraft surface has to collect equivalent ion current, which is hard to do as ions have much slower thermal velocity, or the spacecraft surface has to charge negative to reduce the electron current collection as well as marginally increase ion collection current. Thus, if the spacecraft-surface to probe-surface area ratio is less than 10,000 then any sweeping probe operating in electron saturation region will negatively charge the spacecraft [Szuszczewicz, 1972, Barjatya et al., 2013]. This periodic variation of spacecraft floating potential, called as spacecraft charging, can adversely effect other electronic instruments onboard the spacecraft. As a result of this sweeping bias Langmuir probes are seldom used on small spacecrafts such as sounding rockets or small satellites and CubeSats.

A second common implementation of Langmuir probes that avoids varying the spacecraft potential is a fixed-bias probe. Under this implementation the probe surface is biased at a fixed potential deep into the saturation regions. As the saturation current is directly proportional to density, rapid measurement of collection current

at a fixed bias results in a high temporal and spatial resolution relative density measurement. It is critical to note that one single fixed bias probe can only give relative plasma density measurement, not absolute. Probes biased in ion saturation region give relative ion density and probes biased in electron saturation region give relative electron density. There are two unique implementations of fixed bias probes. A Planar Ion Probe is a flat plate probe that is biased in ion saturation region. As noted in figure 1.2 the ion current collection for flat probe geometry does not change with with increasing voltage. At spacecraft orbital velocities the ion ram current (i.e. the current collected simply by ions ramming into the flat probe sensor) is an order of magnitude larger than ion collection current given by equation 1.3. Thus, if one has precise knowledge of spacecraft velocity, the probe ram cross section area, and assume singly charged ions then we can derive absolute ion density from Planar Ion Probe. The second unique implementation of fixed bias probes is a multi-Needle Langmuir Probe (mNLP). Under this technique multiple needles (very small cylindrical probes) are biased in the electron saturation region [Jacobsen et al., 2010]. Using these three points in the electron saturation region and assuming some electron temperature one can least squares fit equation 1.3 to ϕ_p, n_e, and β. As the bias on needles is not being swept, there is no dynamic spacecraft charging. This also allows for significantly faster sampling of absolute electron density.

The next chapter presents data analysis from Floating Potential Measurement Unit, which is a suite of instruments aboard the International Space Station and includes a spherical and cylindrical Langmuir probe. We will then present error analysis of the mNLP technique which has recently been gaining a lot of interest in the scientific community. Our work on mNLP error analysis has already been published in a peer reviewed journal as Barjatya and Merritt [2018]. We then summarize and conclude the thesis.

Chapter 2

Floating Potential Measurement Unit

2.1 FPMU Overview

The Floating Potential Measurement Unit (FPMU) is an instrument that has been on-board the International Space Station (ISS) since 2006 [Barjatya et al., 2009, Wright et al., 2008]. It was designed to study the charging of the ISS relative to the the space environment. While monitoring the charging of the station was the main purpose of the instrument, its instrument suite allows for an in depth study of the space environment including measuring ion and electron densities, and electron temperatures in the F-region of the ionosphere. FPMU consists of four different instruments, the narrow Langmuir probe (NLP), the wide Langmuir probe (WLP), a plasma impedance probe (PIP), and a floating potential probe (FPP). Each of these instruments were designed to measure or monitor a specific plasma parameter, as detailed next.

The FPP is the simplest of the four probes in the suite. It's a gold-plated sphere of radius of 5.08 cm, which is isolated from the ISS chassis ground by $> 10^{11}$ Ohms. FPP is used to measure the floating potential of the ISS relative to the chassis ground at a rate of 128 Hz and in a range between -180V to 180V from the Station's chassis potential. The ISS chassis floating potential was an unknown value when the FPMU was being designed. Thus the FPP was given a wide measuring range to make sure that the probe will sense the ISS charging potential even as it approaches a value
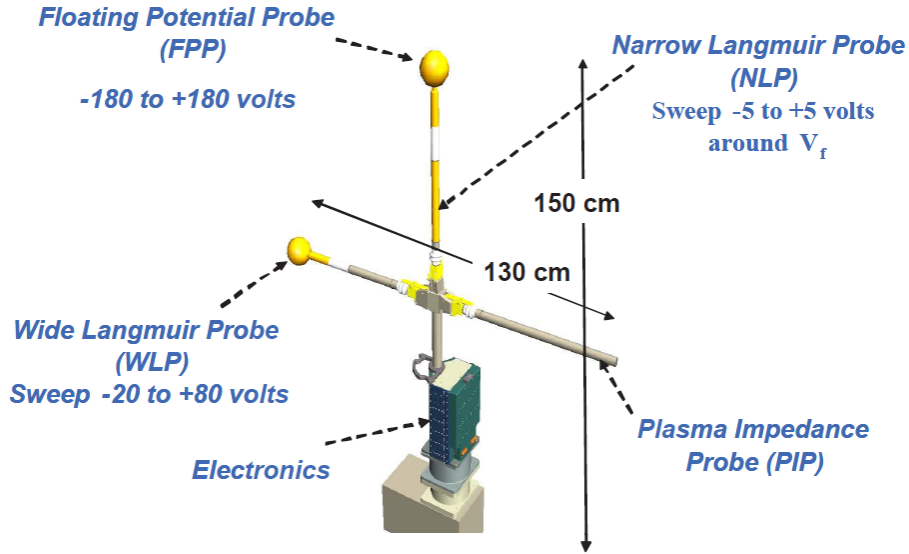
Figure 2.1: FPMU instrument showing the placement of WLP, NLP, PIP and FPP.

high as -140 V, which was a concern for arcing [Hastings et al., 1992].

The WLP is a gold-plated sphere of radius 5.08 cm, that sweeps from -20V to 80V relative to the ISS chassis ground in 2048 steps over 1 second. The WLP was also designed with an internal halogen lamp to heat up the probe surface to boil off contaminants on the surface and thereby reducing hysteresis in the up and down sweeps. This probe's current measurement has two different channels. The first channel is a low-gain channel for currents measuring from -1.40×10^{-4}A to 2.7265×10^{-3}A with an ADC resolution of 7.00×10^{-7}A/count. The second channel is a high-gain channel that is used to measure currents from $-7.00 \times 10^{-6}A$ to $7.3325 \times 10^{-6}A$ with a step size of 3.50×10^{-9}A/count. This sweeping Langmuir probe implementation was called 'wide' owing to a wider range of swept potentials as compared to the 'narrow' Langmuir probe.

The NLP is a gold-plated cylindrical probe with radius of 1.43 cm and length of 5.08 cm. The NLP sweeps in 19mV increments from -4.9V to 4.9V centered around the ISS floating potential as obtained from the FPP in 512 steps over 1 second. This

allows for concentrated sweeping around the electron retardation region with higher resolution voltage steps than the WLP. This would ideally allow better precision in finding the electron temperature. To allow for a wide range of currents while still maintaining fair resolution at small currents, this instrument also has two different channels, each with it's own range and resolution. The first channel is a low-gain channel for currents measuring from -1.75×10^{-5}A to 3.406×10^{-4}A with an ADC count(effective resolution) of 1.75×10^{-7}A/count. The second channel is a high-gain channel used to measure currents from -8.75×10^{-7}A to 9.16×10^{-7}A measuring in steps as small as 8.75×10^{-10}A/count.

The PIP in the suite was added to test an instrument design and allow for an additional absolute electron density measurement to be compare with the Langmuir probes derived densities. This particular instrument is not a Langmuir probe but rather an antenna that sweeps from 100kHz to 20Mhz, measuring impedance in the plasma. Knowing these impedances one can then find the upper hybrid frequency that when combined with a known value of the magnetic field can be used to derive absolute plasma density. Analyzing the data from this experimental instrument was outside the scope of this work and has not been presented here.

Barjatya et al. [2009] have analyzed and presented the data from FPMU in 2006. The same physical instrument has been operating aboard the ISS for the past decade and as part of a new project we were again tasked to determine if the instrument is functioning properly and the data can still be reduced similar to back in 2006. We present our analysis in the next several sub-sections.

2.2   FPMU Data

To test our algorithms, and develop new ones if need be, we were given one day of FPMU data from 2015. The dataset contained raw data from all four FPMU instruments, as well as reduced plasma parameters (density and temperature) from WLP and NLP, derived from algorithms developed and presented in Wright et al. [2008]. Figure 2.2 shows the NASA provided reduced ion density Ni from WLP and NLP for one orbit, superimposed with electron density N_e from the International
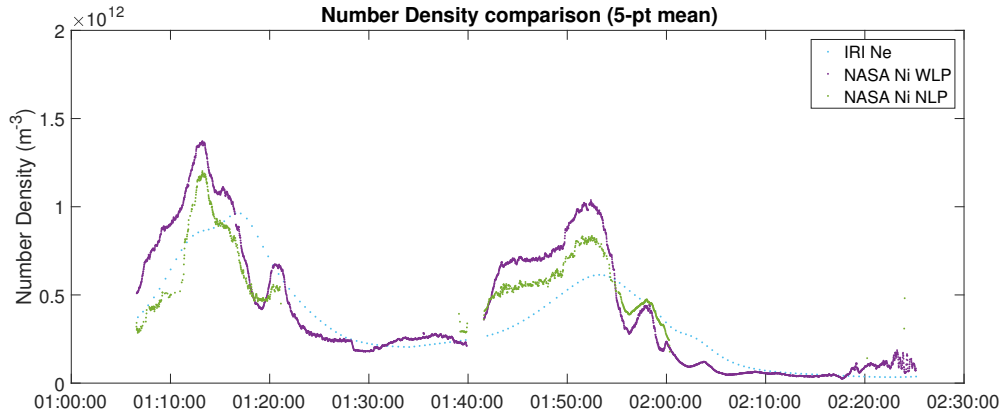
Figure 2.2: NASA derived ion densities compared with IRI model output electron densities

Reference Ionosphere (IRI) model. Ideally, we would like to compare reduced Ne with IRI Ne, but NASA currently only derives Ni as it is easier and faster to do so than Ne (Dr. Kenneth Wright, Personal Communication with Dr. Barjatya). As ionospheric plasma is expected to be quasi-neutral, Ni is expected and assumed to be equal to Ne.

IRI is an empirical model, and as such it is to be expected that the in-situ density does not match the model output in magnitude, and only the general overall curve agrees. But the disagreement between the NLP and WLP derived Ni in the high density regions is unexpected. Furthermore, note that the NLP Ni is missing for parts of the curve, such as between 1:20 and 1:40 hrs, or after 2:00 hrs. This was not due to a data drop-out but rather the inability of the existing NASA NLP Ni algorithm to settle on a derived value from raw NLP data. Also important to note is that the WLP derived Te shows spikes in temperature that are sometimes a factor of four or higher than the IRI predicted temperatures.

Figure 2.3 shows two orbits of data from 2006 as analyzed by Barjatya et al. [2009]. Although this particular figure does not show IRI density data, the excellent match between NLP and WLP derived Ne and Ni is evident. Thus, the current FPMU data processing algorithms being used by NASA can certainly be improved. We next analyze the same orbit using the Barjatya et al. [2009] algorithm, hereafter called as
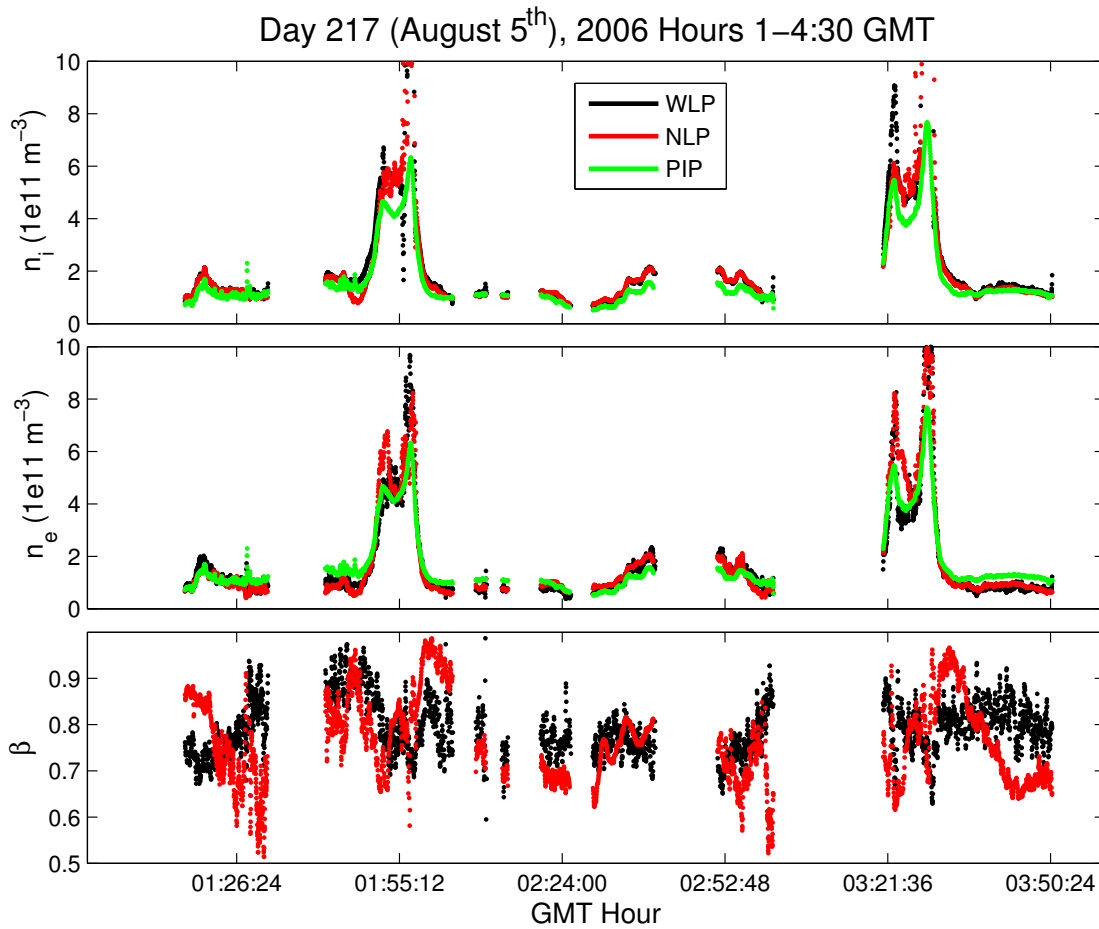
Figure 2.3: Two orbits of data from 2006 being applied the USU algorithm. Taken from Barjatya [2007]

the USU algorithm. We then address each and every unique features of the derived density discrepancies and develop a new algorithm.

2.3 FPMU Data Analysis

Figure 2.4 shows the results of applying the USU algorithm to the new 2015 data. A few points are evident. One, the USU algorithm's derivation of WLP Ni fairly matches the NASA derived WLP Ni. That said, the USU algorithm's derivation of WLP Ne not only has a lot of spread but it also seems to deviate in the low density
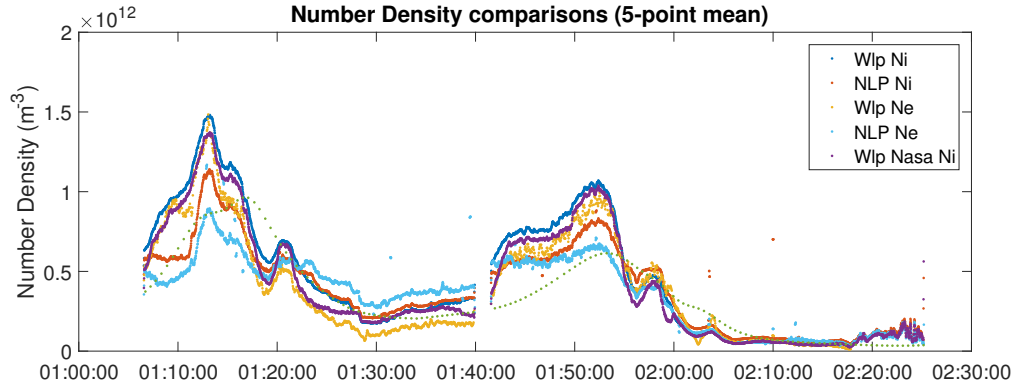
Figure 2.4: The USU algorithm applied to one orbit of data from 2015 and compared with NASA provided values as well as IRI Ne.

regions. Second, the USU algorithm's NLP Ni also matches the NASA provided NLP Ni, albeit the USU algorithm produces NLP Ni even in regions that current NASA algorithm does not. That said, the discrepancy between NLP Ni and WLP Ni remains even with USU algorithm.

Before we start creating a new data analysis algorithm, we first look at some individual WLP curves. One such WLP sweep is shown in figure 2.5. Two interesting features are shown. One, there is an unexplained but pronounced 'kink' in the electron saturation region that was either non-existent or barely present in 2006 IV curves (see Barjatya [2007] for examples). And second, the WLP measurement is saturated in the electron saturation region, which was again not seen in 2006 IV curves and thus the USU algorithm never took that into account. This is likely because the data from 2006 is near solar minimum at the end of Solar Cycle 23 when Ne is low at ISS altitudes. The new data from 2015 is from a period just after the peak of solar activity in Solar Cycle 24 when Ne values are high resulting in saturation of the WLP I-V sweeps.

The writing of new algorithm begins by first taking into account the above features in the electron saturation region. We will still continue to use the OML equations as presented in Chapter 1 and as used in the USU algorithm. First, we fit within the ion saturation region. Due to the large mass of the ions compared to the electrons, there is very little curvature in this region, so we cannot fit this region to the OML
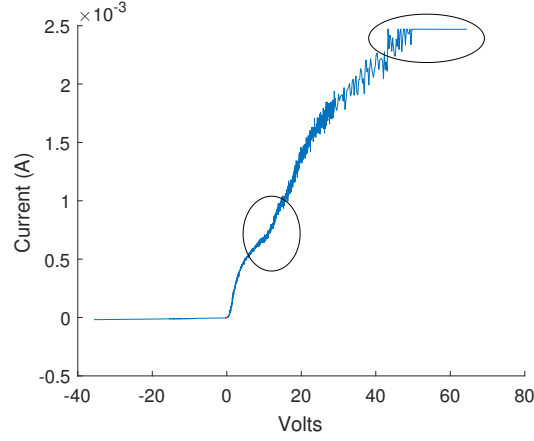
Figure 2.5: An example of full WLP IV curve indicating two important points: first is an unexplained 'kink' in the electron saturation region and the second is the saturation of the instrument

equations. Instead we do a linear fit in this region and then project it to an estimated plasma potential, where the current value derived from the linear fit is assumed to be the ion ram collection current. This is because at the plasma potential there is no net electrical pull on the ions toward or away from the probe, thus the only ion collection currents are the thermal and ram currents. The value of ion thermal current is negligible when compared to the ion ram current, which is given by equation 2.1.

$$I_{ram,i} = n_i q_i A_{ram} V \tag{2.1}$$

where n_i is ion density, q_i is ion charge, A is the ram cross section of the probe, and V is the ram velocity. Given a measured ion ram current, and with assumption of singly charged ions and knowledge of the ISS velocity, one can then determine the value of ion density. Using this value of Ni we fit the electron retardation region, 15 points before the floating potential and 7 points after the floating potential. These points are fit to a sum of 1.7 and 2.1. A least-squares fit to this region helps us derive electron temperature and plasma potential while assuming the electron density to be the same as the ion density derived in the step before. With the fitted value of electron temperature and plasma potential we do another least-squares fit in the
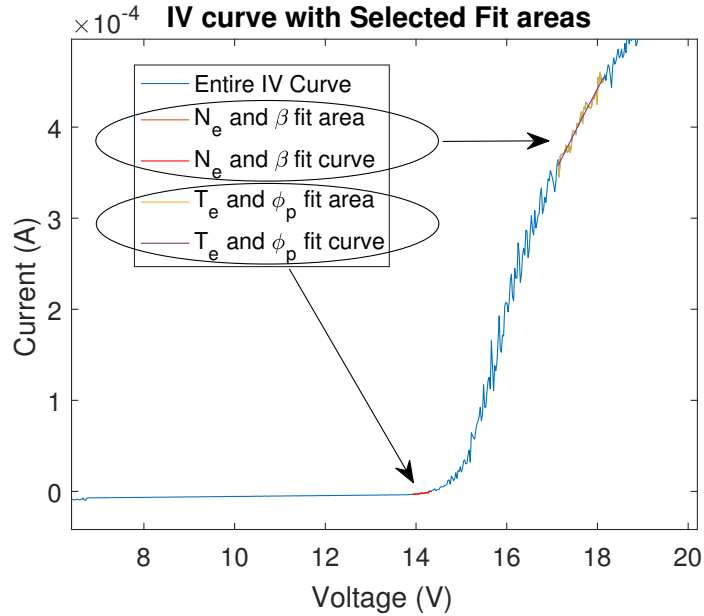
Figure 2.6: An example of WLP IV curve indicating limited parts of the IV curve in the retardation and saturation regions that are used to do the least squares fitting to OML equations to derive the various plasma parameters.

electron saturation region to the equation 1.3. We only fit a few volts above the floating potential however, due to the two features as noted in figure 2.5. The regions of IV curve that are fit for Te and ϕ_p as well as Ne and β are shown in figure 2.6.

We will next look into different issues that appear when individual sweeps are inspected. Each of these issues account for the various discrepancies shown in figure 2.4.

2.3.1   Channel switch between low-gain and high-gain

First we look into the separation between WLP Ni and NLP Ni in the high density regions, as shown between 1:05 and 1:15 hrs and 1:40 and 1:55 hrs in figure 2.4. Investigating sweep by sweep through these regions and looking into the ion saturation regions of both WLP and NLP data sets, we see a sudden step in ion saturation region in WLP IV curves during certain regions of the orbit. This is shown in figure 2.7. Upon close inspection we note that this step occurs near a fixed current value,
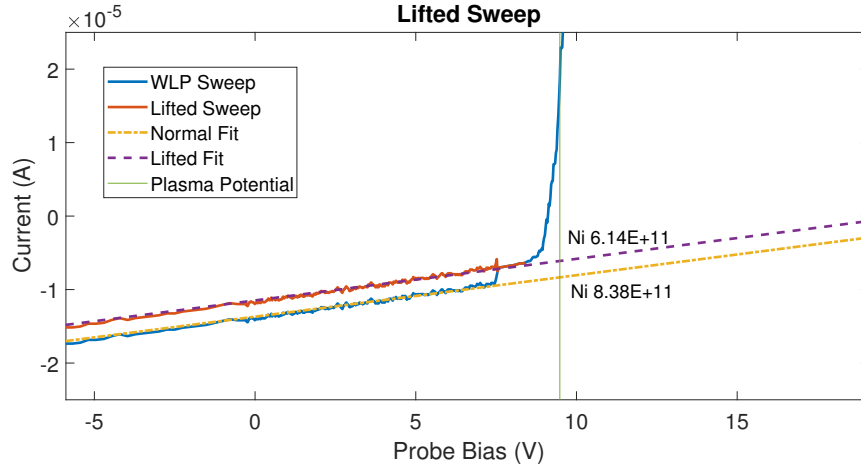
Figure 2.7: WLP IV Curve showing an unexpected step in the ion saturation region (at 8V in this particular instance). Also shown are the derived Ni values when fit to the ion saturation current as-is and after it is elevated such as to remove the step.

which corresponds to the lower bound of the high-gain channel of the instrument i.e. -8.75×10^{-7}A. It is clear that the step is a result of switching from high-gain channel to the low-gain channel. The only conclusion we can draw from this is that the calibration coefficients that convert Analog to Digital Converter (ADC) counts into amperes have changed over the past decade. This is highly likely as the instrument was only designed for an operational life of 3 years [Swenson et al., 2005] instead of the 9 years of life that this data was obtained from.

There is no way to ascertain which of the two calibrations (high-gain vs low-gain) is to be trusted when converting from ADC counts to current in amperes. But one has to assume that one or the other is measuring the true collection current. As the high-gain is the most sensitive channel and also covering the zero-crossing of the current, we assume that calibration coefficients for that channel are true, and we determine the step size that would be needed to correct the error in low-gain channel so the discontinuity between low-gain and high-gain is gone. Ideally, this correction (aka normalization of low-gain current to high-gain current) would be done on complete IV curves seen by both channels, prior to their mixing. But as we have been given a single IV curve that has pre-mixed low-gain and high-gain measurement, we use an algorithm to find this step size through 10000 sweeps. The results are shown in figure
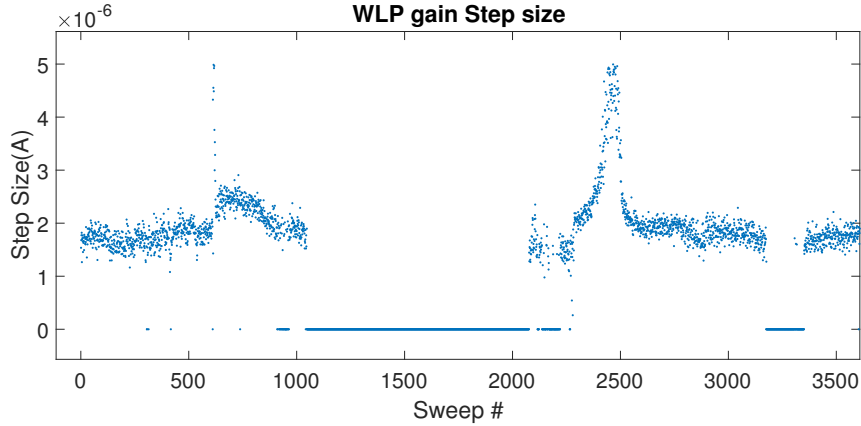
Figure 2.8: Channel step size over 10000 sweeps

2.8. We can see, that within a margin, the step size is in fact consistent, albeit with some spread in the points. The spread is just an artifact of the algorithm not being able to catch the step size precisely in every sweep. Note that the dropouts in the plot data is simply in the low density regions where the low-gain channel is not needed, and as the entire sweep is in the high-gain channel there is no step discontinuity. From these plots we can see that the step is about $2\mu A$ in magnitude. Using the lifted value of low-gain channel, we see that the derived WLP Ni is now a lower value, as shown in 2.7. It is important to note that while this $2\mu A$ channel step is applicable throughout the day's worth of data that we were provided, it is unclear if this will be consistent throughout the decade worth of data. In other words, this channel-switch step might be changing in time and it is highly recommended that any operational code for FPMU data analysis normalizes the low-gain IV curve to high-gain IV curve before mixing them, or alternately check for this step size on pre-mixed IV curves on a monthly or quarterly basis and account for them.

We now correct this step for every low-gain channel switch throughout the orbit and re-derive the WLP Ni and WLP Ne. This is shown in figure 2.9. As compared to figure 2.4 the newly derived WLP Ni and WLP Ne agree much better with the NLP Ni. This is especially true between 1:40 to 1:55 hrs. There remains some 'flat-looking' mis-match between WLP Ni and NLP Ni between 1:05 and 1:15 hrs, although its lesser than it was in figure 2.4. We will investigate and address this next.
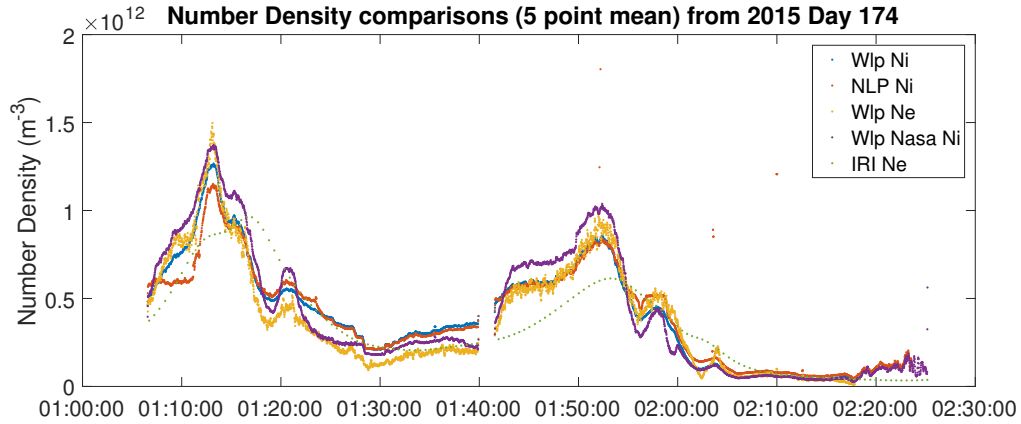
Figure 2.9: The derived densities after accounting for the step which was a result of channel switch from high-gain to low-gain.

2.3.2 NLP ion current distortion

As we continue to look through the derived densities in figure 2.4, we note that when entering into the equatorial anomaly portion of the orbit (between 1:05 and 1:15 hrs), there is a flat distortion in the NLP data. Upon investigation of the whole day's worth of data, this same discrepancy is seen several times. In fact, it was even present in 2006, see figure 2.3 between 1:45 and 1:50 GMT hrs, but never addressed or accounted for. To investigate this further we look at the IV curves in this region. While the WLP IV curves show no major discrepancies, the NLP IV curves shows an odd dip in the ion saturation region, as well as noticeable amount of quantization noise. This is shown in figure 2.10.

Just like the unexplained kink in the electron saturation region in WLP (figure 2.5), as well as the 'negative characteristic' seen in electron saturation region in NLP [Barjatya et al., 2009], this dip in the NLP ion saturation region is also unexplained. It is most likely some unknown instrument physics that only shows up when the angle of the NLP to the ambient magnetic field is within certain bounds that are only observed when the ISS is in the vicinity of the equator. As we don't know why this feature exists in the IV curve, we find a way to fit around it to minimize this error. Fitting for Ni to the right of this dip would be influenced too much by the electron retardation region current. Thus, we fit to a much smaller ion saturation region to
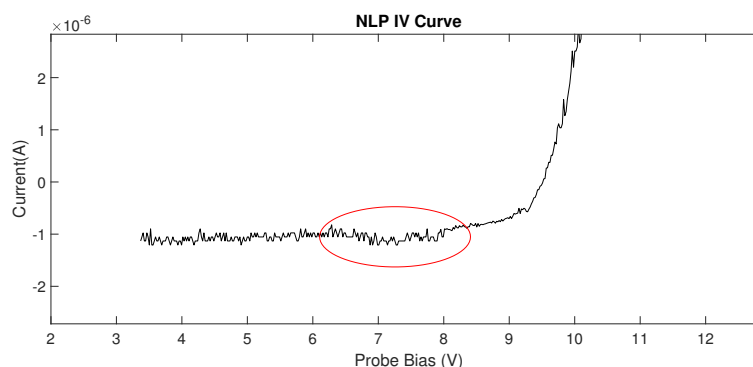
Figure 2.10: An unexplained dip in the NLP ion saturation current region.

the left (more negatively biased) portion of the IV curve. The only issue with doing this however, is that it limits the amount of data points to fit to, and as can be seen in the figure, the data region is highly discretized due to the lack of resolution in the high-gain channel, and thus slightly decreases the quality of the fit. The results of this circumvention are shown in figure 2.11. As is evident, applying this different fit region pulls up the NLP Ni into agreement with the WLP Ni and the flat-feature in the NLP Ni between 1:05 and 1:15 hrs is now gone.

2.3.3   Hysteresis

Next we investigate presence of hysteresis in NLP and WLP IV curves. Figure 2.12 plots six consecutive sweeps simultaneously from both instruments. While NLP showed hysteresis in 2006 also, it seems to have become far worse in 2015. Accumulation of new contamination while in orbit for 10 years is rather unlikely as sputtering from ramming ions is expected to clean the surface. The only conclusion we can come to is that there have been some deformities on NLP surface that have led to a non-uniform work function resulting in increased hysteresis. The FPMU team at NASA MSFC has discussed this issue of NLP hysteresis after seeing the results of our work. An FPMU image survey is being considered to see if there is any mechanical damage on the NLP due to orbital debris impact, or non-uniform erosion of the gold plating, or some other mechanism [Dr. Joseph Minow, Personal Communication]. NLP IV curves will hence not be used for deriving Ne and Te. That said, derivation of Ni
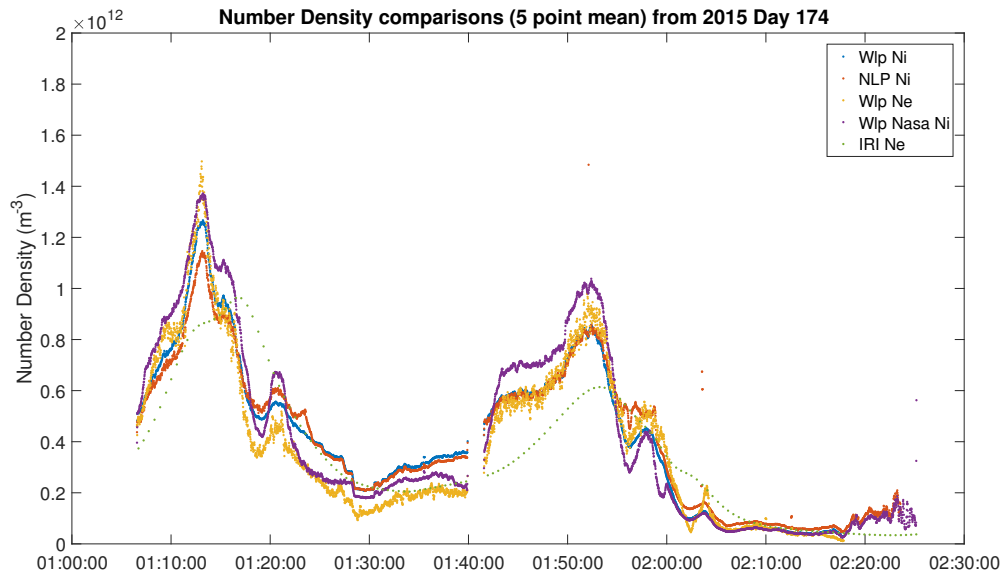
Figure 2.11: The derived densities after circumventing the dip in the ion saturation region in NLP IV curves.
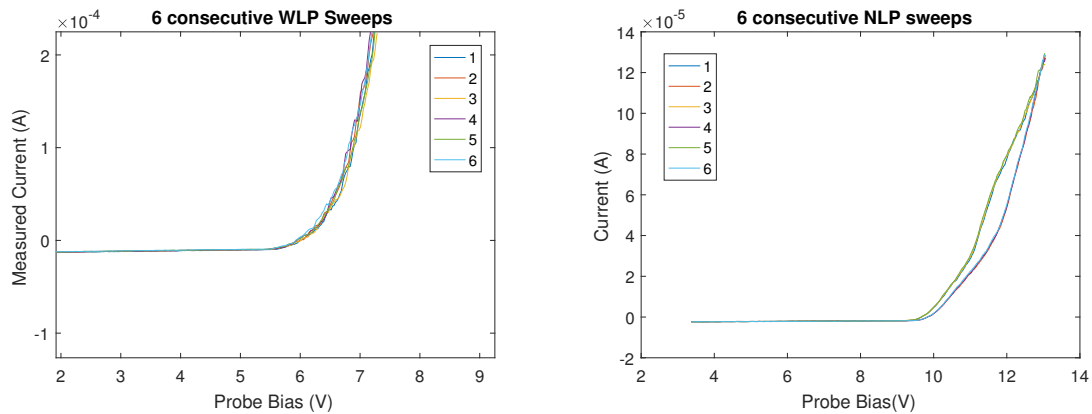
Figure 2.12: Plots of six consecutive IV curves of each instrument. Although there is little apparent hysteresis in WLP, the NLP shows significant hysteresis.

should still be unaffected.

Also worth noticing is that WLP does not seem to show any significant hysteresis as further shown in two representative consecutive IV curves in figure 2.15. The derived Ne and Ni from 50 consecutive WLP sweeps is shown figure 2.14. The derived
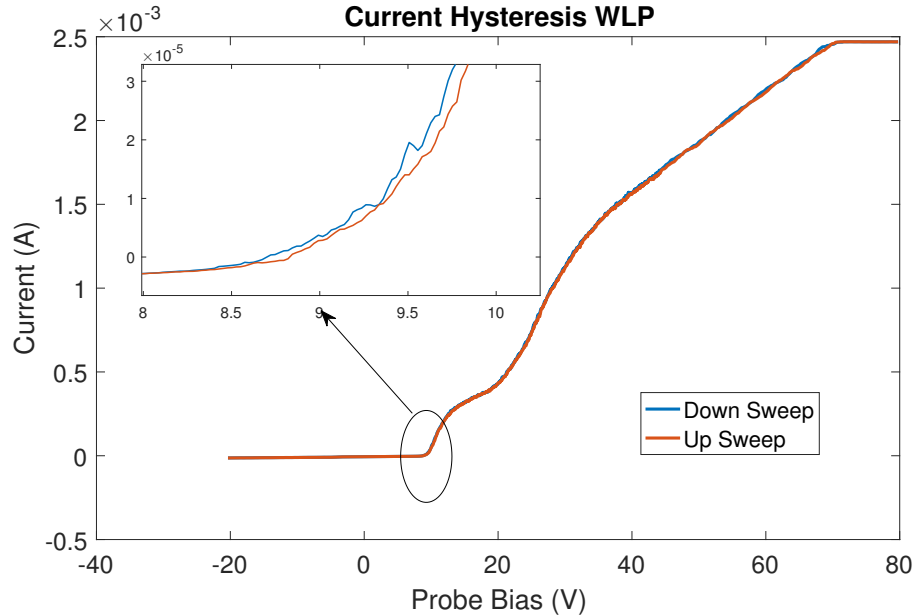
Figure 2.13: Two complete consecutive WLP IV curves

WLP Ne values from the electron saturation region in these 50 sweeps show an alternating pattern, which can be misconstrued as a noisy instrument. When separated based on upsweep and downsweep, it is clearly evident that the downsweep derived Ne is being affected by the very slight difference in IV curves. Henceforth we only rely on WLP upsweeps for Ne derivation.

2.3.4 Photoelectric effect

We next look into the separation between Ni and Ne in the lower density portion of the orbit. Specifically, between 1:20 and 1:40 hrs in figure 2.11. Note that while WLP Ni and NLP Ni agree with each other, they do not agree with WLP Ne. As a sweep by sweep inspection of WLP and NLP IV curves does not come up with any explanation, we must look into other effects that could cause this disagreement between Ni and Ne values. We notice that while this disagreement between Ni and Ne exists in the low density region from 1:20 to 1:40 hrs, its does not exist in the even lower density region from 2:05 to 2:25 hrs. As this issue appears in only half of the 90 minute orbit, a possible logical explanation would be the photoelectric current
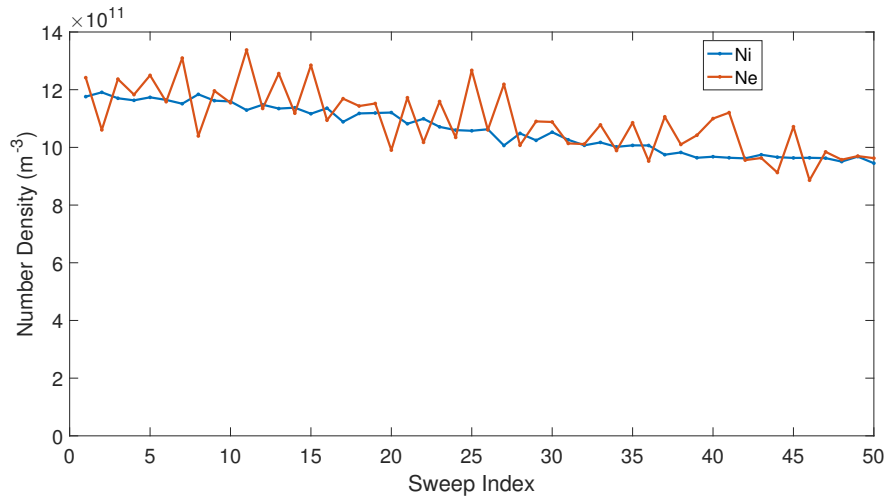
Figure 2.14: 50 consecutive derived Ne and Ni in a high density region. The two consecutive IV curves shown in 2.15 are amongst the 50 sweeps chosen here.
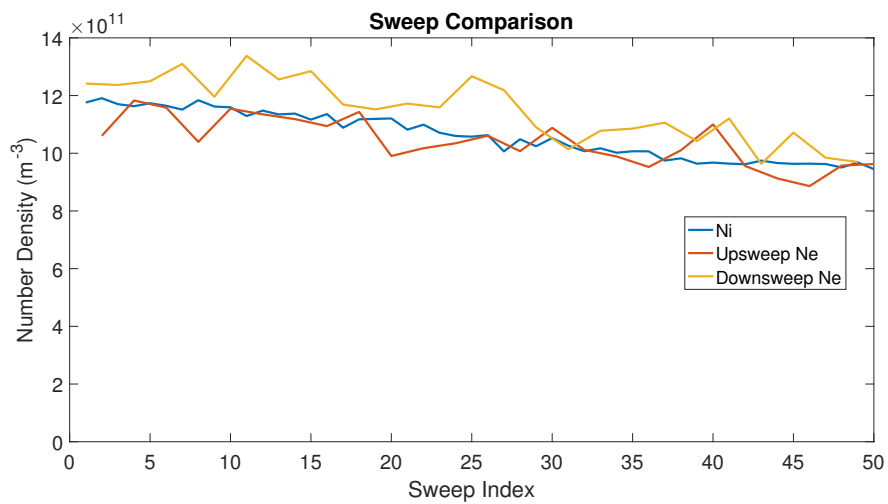
Figure 2.15: The same derieved Ne and Ni as in Fig:2.14, but isolated as upsweeps and downsweeps
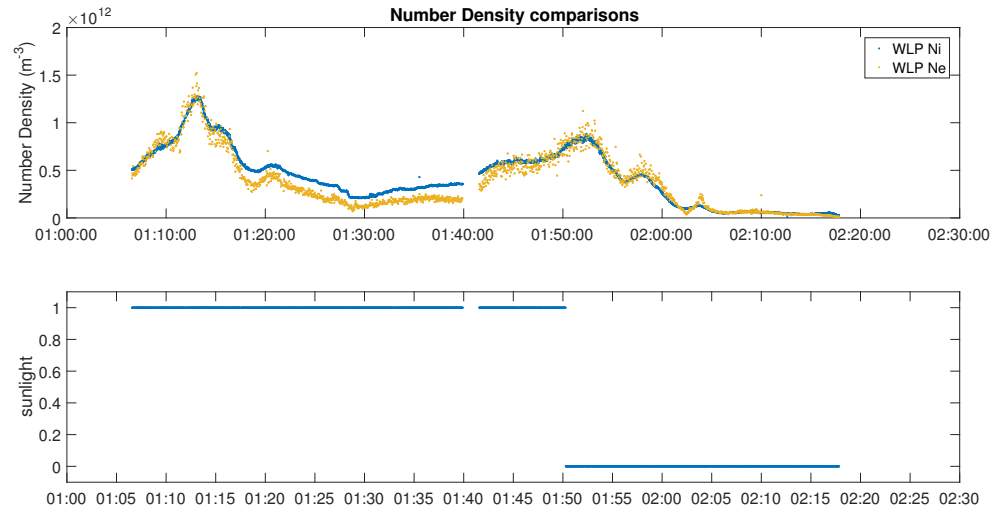
Figure 2.16: Only Ne and Ni from WLP are shown. Note that the disagreement between Ne and Ni is only in the lower density region when the ISS is in sunlight, as indicated by the sunlight flag being equal to 1.

emission from the probe surface as the ISS moves from eclipse to daytime. A simple confirmation of this is to plot our density curves and compare it to the sunlight data, as is shown in figure 2.16.

As we can see, this separation between Ni and Ne values occurs in sunlit portions of the orbit in the lower density regions. This separation is caused by photo-electrons being kicked off from the probe surface by incoming solar radiation. Emission of photoelectron current especially affects the ion saturation region where the probe is biased negative. This emission current thus gets added to the incoming ion current. Photoelectron emission current does not effect the electron saturation region because the probe is biased positive and prohibits emission of photoelectrons.

The magnitude of the photoelectric current is dependent upon several factors: the work function of the metal surface, the spectrum of the incoming light, and the cross section area exposed to solar radiation. The first two of these are constant and known, as the work function of gold (the probe's surface) is a known constant, as well as the spectrum of sunlight which is the only light source providing high enough energy photons to free electrons from the probes surface. This leaves the cross section of the

instrument exposed to sunlight as the unknown variable. While the obvious solution would be the cross section of the probes (circular for WLP and rectangular for NLP), this isn't always true. Once the ISS enters sunlight, as well as when the ISS moves directly between Earth and Sun, sunlight will be reflected by other ISS surfaces as well as albedo from Earth's surface, back onto the probe causing more surface area than just the cross section to be emitting photoelectrons. As photoelectron current is an addition on top of the ion collection current (note the opposite polarity of the charge species), this causes an artificial inflation in derived ion densities if not accounted for properly.

To compensate for this effect, we must include photoelectric emission current from the probes. Gold's photoelectric emission current density is $29\mu A/m^2$ [Hastings and Garrett, 1996]. Our approximation for the amount of photoelectric current emitted assumes that at the dawn/dusk regions of the orbit the area illuminated (and thus emitting photo electrons) is equal to the projected area of the probe (i.e. πr^2 for sphere), whereas when the ISS is directly in between the Earth and the Sun the area emitting the photoelectric current is the entire surface area of the probe attributable to reflections from ISS surface and/or Earth's albedo. Accounting for the photoelectron emission current in such a way gives us results as shown in figure 2.18. As expected, this compensation causes the derived ion density to drop down to be closer to the derived electron density values. It is important to note that while the agreement is better between Ne and Ni, it is not perfect. One possible reason for this could be that we have used linear interpolation between sunlight illuminated area being πr^2 at eclipse exit and being $4\pi r^2$ when directly between Earth and Sun. The real transition between these two bounds might be some non-linear function. That said, it is better to account for photoelectron current emission even with a linear interpolation approximation than to not account for it at all.

2.4   Summary

We were provided one day's worth of FPMU flight data from 2015 and tasked to provide a third party verification to the NASA derived plasma parameters. We found
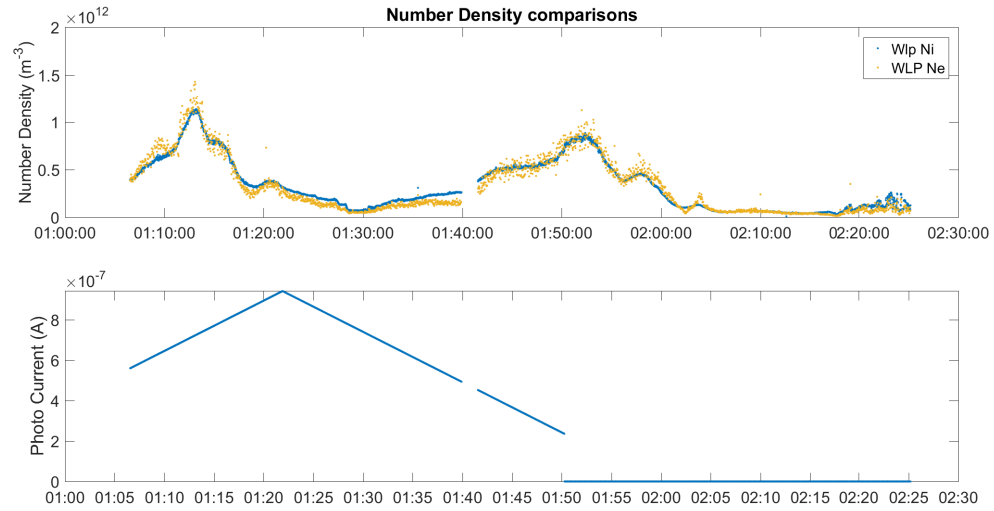
Figure 2.17: The WLP Ni and Ne derived densities after accounting for photoelectric current emission

several discrepancies in the derived plasma parameters leading us to believe that instrument performance has changed, arguably degraded, over the duration of more than a decade that the instrument has been operational on ISS. We have found that there is more hysteresis in the NLP IV sweeps which brings into question NLP derived Ne and Te values. We also found an unexplained dip in the NLP ion saturation region that, for now, we have accounted for. Although this circumvention works for the day's worth of data provided, it may or may not work long term if this feature is evolving. It has also become evident that some sort of normalization needs to be done in the WLP high-gain channel and low-gain channel IV curves. This should preferably be done prior to mixing the two channels but can also be done after mixing them by accounting for the step seen when switching from high-gain to low-gain channel. This may need to be checked on a monthly basis. Lastly, we found that photoelectron emission current is not negligible if the ISS is flying through low density region (i.e. outside of equatorial ionization anomaly) during daytime. Although photoelectron emission current for gold is well known, the exact probe surface area illuminated with solar radiation is unknown. For now we are approximating it as a linear interpolation between ram cross section area at eclipse exit and total surface area when ISS is
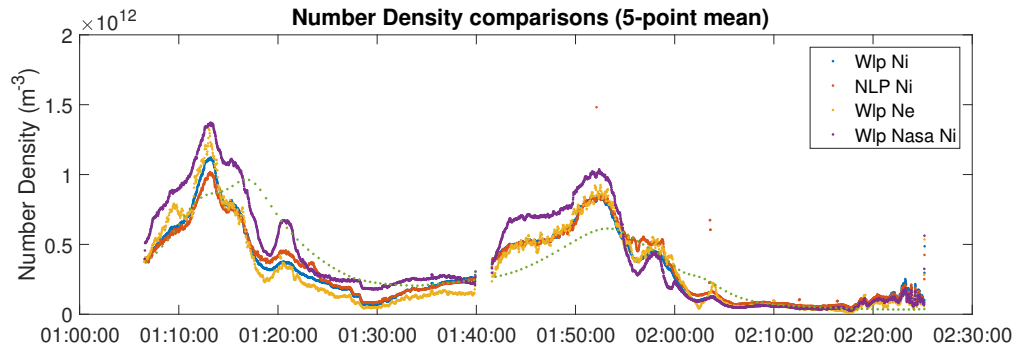
Figure 2.18: The final derived densities after accounting for all the corrections as mentioned above.

directly between Earth and Sun. Overall, we have now provided a new algorithm that should work better than current NASA algorithm, as is evident by the fact that WLP Ni, WLP Ne, and NLP Ni are in fairly good agreement. That said, there is still a lot of work that remains to be done to better ascertain electron temperature. The matlab algorithm is listed in Appendix A. The entire day's worth of analyzed FPMU data using the new algorithm is shown in Appendix B.

Chapter 3

Error analysis of multi-Needle Langmuir Probes

Note: This chapter has been published as Barjatya and Merritt, "Error analysis of multi-Needle Langmuir Probe measurement technique", Review of Scientific Instruments, 89, 043507 (2018); doi: 10.1063/1.5022820, and is being produced here verbatim.

3.1 Introduction

Langmuir probes are the most commonly used instruments for plasma density diagnostics on sounding rockets and satellites. The technique is simple: a metallic sensor immersed in plasma is applied a voltage V and the collected current I is measured. The resulting I-V curve is then analyzed to determine various plasma parameters such as electron and ion density, electron temperature, and spacecraft floating potential [Barjatya et al., 2009]. The instrument can be implemented in primarily two ways. First, and most commonly, as a fixed-bias probe wherein the voltage is kept constant relative to the spacecraft chassis ground. As the collected current is directly proportional to density, this implementation results in high cadence relative density measurement as long as there are no significant spacecraft charging events and the plasma temperature remains in a fairly narrow range (usually within few hundred

27

Kelvin). The second way is a sweeping Langmuir probe where the voltage is swept from some negative bias to a positive bias, thereby recording the entire I-V curve. As one sweep can only give one measurement of each plasma parameter, the sweeping potential implementation of the Langmuir probe has lower cadence measurement of plasma parameters. Both implementations are susceptible to surface contamination [Barjatya et al., 2013, Steigies and Barjatya, 2012], although only the sweeping probe adversely affects other electric probes on the spacecraft by swinging the spacecraft floating potential, especially when the spacecraft-to-probe surface area ratio is smaller than few thousand times [Barjatya, 2007]. Thus, in order to avoid affecting the payload floating potential, fixed bias probes are largely favored to measure relative plasma density.

Multi-Needle Langmuir Probe (mNLP) is a relatively new technique that uses multiple fixed bias Langmuir probes to derive absolute plasma density that is independent of spacecraft charging [Jacobsen et al., 2010]. This instrument technique has been used on several sounding rockets [Bekkeng et al., 2013] [Fisher et al., 2016] and is also being implemented for CubeSats and small satellites. This paper first presents a brief overview of the technique and then elucidates how the current data processing of the mNLP can lead to significant errors. We then propose an alternate method of data analysis that is expected to work better.

3.2 Multi-Needle Langmuir Probe Technique

The electron saturation current collected by a Langmuir probe operating in an Orbital Motion Limited (OML) regime is given by equation 3.1,

$$I_e = n_e e A \sqrt{\frac{k_B T_e}{2\pi m_e}} \left(1 + \frac{e(\phi - \phi_p)}{k_B T_e}\right)^{\beta} \tag{3.1}$$

where e, n_e, T_e, and m_e are the charge, density, temperature and mass of electrons, k_B is the Boltzmann constant, A is the surface area of the probe, ϕ is the applied potential relative to ϕ_p plasma potential, and the variable β is set to 0, 0.5 or 1 based

on the probe geometry of flat plate, cylinder or sphere, respectively.

Operating in OML regime requires the probe diameter to be much smaller than the debye sheath. The multi-Needle Langmuir probe accomplishes that by using less than 1 mm diameter needles as fixed bias Langmuir probes. The mNLP technique relies on the fact that for cylindrical Langmuir probes, the square of the saturation current has a linear relationship with the applied relative potential. One can then derive absolute electron density using only the measurements at discrete points in electron saturation region. The equations governing the process are shown below

$$I_e^2 = \frac{(n_e e A)^2}{2\pi m_e} \left(k_B T_e + e(\phi - \phi_p) \right) \tag{3.2}$$

$$\frac{dI^2}{d\phi} = M = \frac{n_e^2 e^3 A^2}{2\pi m_e} \tag{3.3}$$

$$n_e = \sqrt{\frac{2\pi m_e M}{e^3 A^2}} \tag{3.4}$$

This is the method used by Jacobsen et al. [2010] in a paper covering data analysis of the mNLP instrument aboard the ICI-2 sounding rocket mission. Typically anywhere from 3 to 8 needles are used to create a line fit between square of the measured current and the relative potential difference between the applied potential. The unique benefit of the mNLP technique is that only the potential difference between the applied potentials to the needles is relevant, making this technique relatively immune to spacecraft charging as long as sufficient number of needles (more than 3) are operating in electron saturation region. As rockets and satellites typically charge -1V to -2V in nighttime ionospheric conditions, needles biased higher than 3.5V should not be affected by spacecraft charging.

3.3 Data analysis discussion

Several papers [Barjatya et al., 2009, 2013, Hirt et al., 2001]have shown that value of β in equation 3.1 rarely follows OML theory values. It is important to note that

the papers referenced here had probe size larger than expected Debye length so the departure of β from OML theory predicted values was to be expected. The entire premise of the mNLP technique is that the very thin 'needle' probes are much smaller than the Debye length and consequently behave in the OML regime with the collected current following the $\beta = 0.5$ curve in the saturation region. One way to show that the probe measurements conform to OML expressions is by showing the linearity of the I^2 measurements w.r.t applied voltage. Jacobsen et al. [2010] have shown 6 such instances throughout an ionospheric rocket flight. They have shown the correlation coefficients of a linear fit of I^2 measurements to V vary between 0.997 to 0.9993. Similarly, Friedrich et al. [2013] have noted that the ECOMA 7, 8 and 9 flights had the I^2 vs V linear correlation coefficients between 0.97 and 0.99, but do not mention how that translates into error bars on the density calculation. This paper investigates the magnitude of error in derived absolute plasma density even when the I^2 vs V linear fit correlation coefficients are as good as seen on ICI2 and ECOMA 7,8 and 9 flights.

Using equation 3.1, we simulated electron saturation currents at four different voltages similar to ICI2: 2.5, 4, 5.5 and 10V, using three combinations of electron temperature and density that are representative of various regions and conditions within the ionosphere: 800K and $1 \times 10^9 m^{-3}$, 1200K and $1 \times 10^{11} m^{-3}$, 2000K and $1 \times 10^{12} m^{-3}$. The simulated current values at these four potentials were generated with β value varying between 0.45 to 0.85. We then calculated the linearity of the I^2 measurement vs potential difference between the points. This is shown in figure 3.1. For these three combinations of density and temperature, the linearity fit of these four points is largely the same across the different β values and only varies slightly for higher β values. It is crucial to note that for $\beta = 0.6$, the coefficient of correlation is 0.9998 or better in all three cases. This is better than the best correlation case shown in the Jacobsen et al. [2010] paper, which was 0.9993. Thus, it is reasonable to assume that the β value observed in-situ during ICI2 rocket flight was unlikely to be 0.5. For these combinations of temperature and density, the Debye length is expected to vary from 3 mm to 60 mm, which is an order of magnitude or larger than ICI2 mNLP needle radius of 0.25 mm.
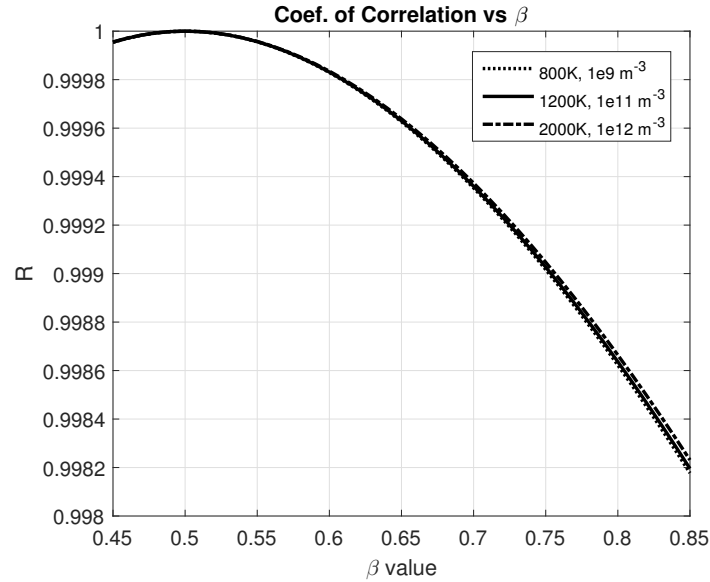
Figure 3.1: Pearson Coefficient of Correlation for a range of β values for three combinations of density and temperature. Note the negligible variation between three combinations of density and temperature, which only marginally changes at higher β values.

After this data was simulated, we used equation 3.4 to derive the electron density, i.e. the densities were derived assuming $\beta = 0.5$. The resulting densities were then compared with the simulation input densities for error. This comparison is shown in figure 3.2. As expected, for the currents simulated with β value fairly close to 0.5, the use of equation 3.4 results in very little error. But if the simulated β value deviated even 10% (to 0.55) then the error in calculated density using mNLP technique can easily approach 30% or more. With a $\beta = 0.6$, the error in derived density can be as large as 70%, even though the four I^2 points show excellent linearity, as was indicated by figure 3.1.
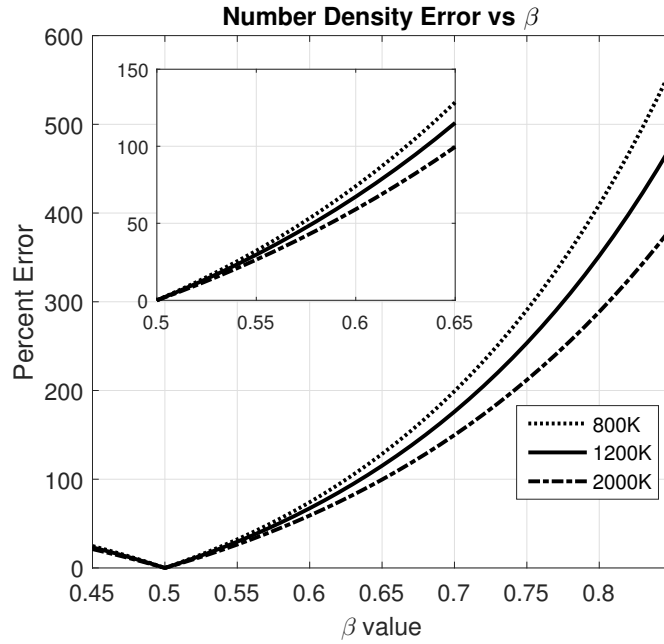
Figure 3.2: Number density error for varying values of β. The inset is a zoomed section from $\beta = 0.5$ to $\beta = 0.65$.

Note that we have not simulated any spacecraft charging in these plots. A worst case spacecraft charging of -2.5V, such as seen by the Bekkeng et al. [2013], will have adversely affected the 2.5V biased needle measurement and further worsened the linear fit. In fact, Bekkeng et al. [2013] not only ignored the 2.5V needle data point, but also the 4V needle point as that data was corrupted. They derived electron density using the mNLP technique (i.e. equation 3.4) with only two needles. In the Earth's mesosphere, the densities are lower and hence the Debye length is much larger. Thus, one would expect the mNLP instrument to behave in the OML regime and the observed β value to be closer to 0.5. Despite a large Debye length, the mNLP derived density was a factor of 2 (i.e. 100%) different when compared with Faraday rotation derived absolute density [Friedrich et al., 2013]. However, once normalized to the Faraday rotation density numbers at 97 km, the mNLP derived densities were within 15-20% of the Faraday rotation derived density profile. This normalization defeats the purpose of using mNLP instrument as an absolute density measurement and requires another instrument to be present onboard the rocket/satellite to provide

the absolute density measurement to which mNLP data could be normalized to.
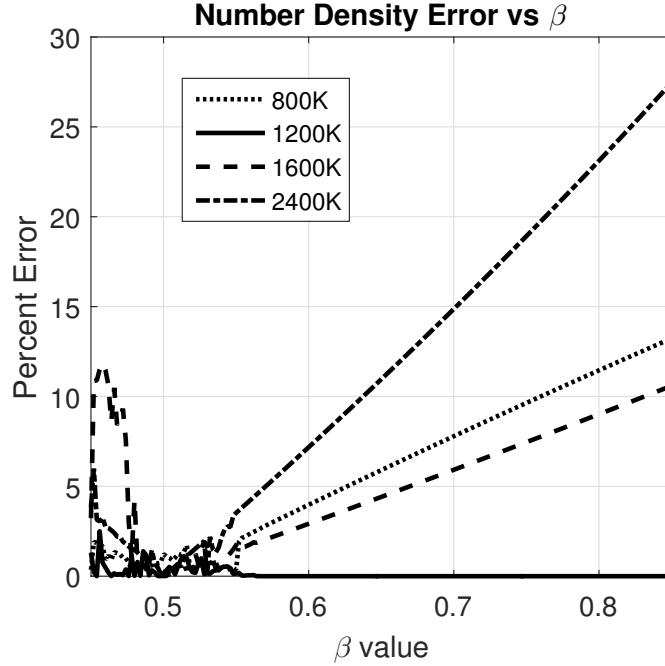
Figure 3.3: Number density error when fitting for β over three points: 4V, 5.5V and 10V. Note that the error is nearly zero when the assumed temperature is exactly the same for simulated current values i.e. 1200K. A 100% error in assumed T_e (i.e. 2400K) only results in 7.5% error in derived n_e at $\beta = 0.6$.

In light of the above, we instead propose using a β fitting technique similar to Barjatya et al. [2013] and Barjatya et al. [2009]. We have four unknowns: β, n_e, T_e, and ϕ_p (i.e. spacecraft charging). We propose that the four measurement points (or more) from a mNLP-type instrument be used to fit for these four unknowns in a least squares sense to the OML current collection equation 3.1. Although four points are sufficient for fitting for four unknown parameters, but assuming a worst case scenario where the lowest biased 2.5V needle is corrupted by spacecraft charging and only three points/needles are available, we fit for β, n_e, and ϕ_p over measurements at 4, 5.5 and 10V. We do the fits 'assuming' various temperatures that deviate from the simulated temperatures by +/-50% and +100%. And finally, also note that we generated the simulated currents using a spacecraft charging value of -2.5V. The resulting error between derived densities and input densities after fitting for β, n_e, and ϕ_p are shown

in figure 3.3. Note that the error drops down significantly as compared to doing an analysis assuming that $\beta = 0.5$ (see figure 3.2). This is even true when the assumed temperatures are significantly off from the temperatures used to simulate the currents. This is to be expected because saturation current regime is fairly independent of electron temperature. Note that at $\beta = 0.5$ the T_e term cancels out, thus the error is less dependent on assumed T_e value when closer to $\beta = 0.5$, and worsens with temperature as the observed β value increases.

We next simulated currents on four voltages: 3.3V, 4V, 5.5V, and 10V and fit for β, n_e, and ϕ_p. This is shown in figure 4. As there are more points then there are unknowns, the fits are much cleaner. So we recommend that any future implementations of mNLP type probes use at least four points that are not corrupted by spacecraft charging. The more the better, albeit that comes at a cost of increased data to downlink.
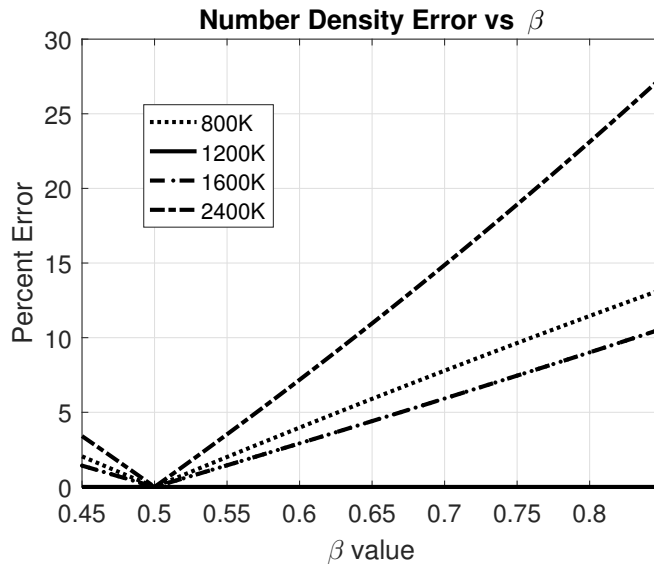
Figure 3.4: Number density error when fitting for β over four points: 3.3V, 4V, 5.5V and 10V. The fits are a lot cleaner for lower β values and the error in calculated density continues to be much lower than when assuming $\beta = 0.5$.

3.4 Summary

We have shown here that the existing analysis method for mNLP probes, which assumes the square of the measured needle currents has a linear relationship to applied potential, can result in significant errors in calculated absolute electron density. This error is a result of the assumption that the electron saturation current varies with $\beta = 0.5$. Our work has shown that even a 10% error in β observed by the needles can result in 30% or more error in calculated density. In a real scenario, the needle current measurements at discrete points in the electron saturation region will be corrupted by inherent electronic noise as well as any wake effects, thereby increasing the resulting error percentage. Additionally, if the needles are spatially separated then any local density variations have the potential to vary the β value seen by individual needles, thereby further increasing the error in calculated density. Nevertheless, the error that one gets by least squares fitting for β, n_e, and ϕ_p, and hence deriving absolute electron density will be far less than assuming the β to be 0.5. We also suggest that any mNLP implementation include at least four needles that are biased above the spacecraft charging potential such that they are clearly in the electron saturation region.

Chapter 4

Summary and Conclusion

Langmuir probes have been around since 1924, but refining the process of constructing them and interpreting the data is always a tailoring process. As simple as the premise of operating a Langmuir probe is, the variety of the environments that the probes are deployed into and the lack of unified set of equations that govern the behavior of these probes in these myriad environments, makes the data analysis of the measured I-V curves very complex. This thesis covers a few different methods of implementation and interpretation of data of these probes, and issues with them.

We first looked into the data analysis of the Langmuir probes that are part of the Floating Potential Measurement Unit onboard the International Space Station. We compared the NASA algorithm provided plasma parameters with the output of Barjatya et al. [2009] algorithm. The WLP Ni from Barjatya et al. [2009] matched the NASA provided WLP Ni, where as NLP Ni from Barjatya et al. [2009] provided values in the regions where NASA NLP algorithm did not. Additionally, the Barjatya et al. [2009] provided WLP Ne seemed to largely agree with the Ni values. We have found that the devil is always in the details. We found that over the past decade NLP IV curves have started showing more hysteresis than they did in 2006. This basically excludes the use of NLP for Ne and Te. We also found that accurate calibration that converts ADC counts into current is extremely important. Even if that is done, one has to periodically perform checks if it holds in flight as long term exposure to space environment tends to change behavior of electronic parts. Such seems to

be the case with WLP IV curves. We also note that the WLP low gain channel needs periodic corrections to match it to high gain channel output. We also found that it is important to cross check the output of one instrument against another as well as crosscheck one derived quantity with another. While it is possible that both may be giving the wrong answer, it is unlikely. The agreement of measurement from multiple instruments gives confidence in one's measurement. Doing such comparison we noted that derived Ne was sometimes lower than derived Ni. This was traced back to the presence/absence of photoelectron current emission from the probes. After accounting for the photoelectron emission current the derived Ne and Ni agreed better. It is very important to note that the area illuminated from sunlight is not just the probe cross section towards the sun direction, but also the cross section exposed to reflections from spacecraft surface and/or planetary albedo. The actual illuminated cross section is unknown as well as the spectrum of the reflected light, but making some linear interpolation assumption is better than not accounting for it at all. After the development of our new FPMU data analysis we find that WLP Ni, WLP Ne, and NLP Ni largely agree with each other. We also found that the the factor of four deviation of NASA provided Te when compared to IRI model output Te is likely erroneous. Our WLP derived Te does not show that much variation from IRI empirical model. That said, there is still much work to be done to get acceptable Te results from WLP.

In the second part of this thesis we looked into a new upcoming technique of multi-Needle Langmuir probes that is based on utilizing three (or more) fixed bias needles operating in the electron saturation region. The fact that the needles are not sweeping voltage helps when they are deployed on small spacecrafts or sounding rockets. The data analysis being used by Jacobsen et al. [2010] and Friedrich et al. [2013] assumes that the square of the current will have linear relationship with applied voltage. Our work has shown that such is not the case. We believe all needle current measurement need to be downlinked to the ground station where the appropriate OML current expression can then be fit to the data points. Usage of our suggested technique can reduce the errors from more than 50% to less than 10%.

Overall, we have found that data analysis of Langmuir probe requires understanding of both plasma physics and instrument behavior. Our work has shown that the Langmuir probe instrument technique still produces the easiest, if not the best, way to determine space environment parameters.

Appendices

Appendix A

Data Analysis Code in MATLAB

A.1 Code Flow

The flow of this code starts with the WLP analysis code (WlpFPMUIterate.m) which loads the data from a MATLAB data file. The contents of the data file are noted within the .m file's comments. The WlpFPMUIterate.m performs some initial calculations, and then goes into a parallel for-loop. Each run through the loop is analyzing one sweep of data, and as each sweep is independent of any other sweep, they can be analyzed in parallel in a random fashion thereby making full use of multi-core processors. Inside the for-loop, a few more calculations are done before passing the code to the fitting function (IterateWLP.m) that uses Matlab's native Least Squares Curve fit function. This IterateWLP.m pulls data out and fits in the two different regions of teh IV curve: the electron retardation region solving for plasma potential and electron temperature using the HWIonVp.m function for the model, and part of the electron saturation region solving for electron density and beta using OMLSaturation.m file for the model. After the code is done the saved parameters are Ni, Ne, Te, Vp, Vf, Beta, and resnorm for the fits in the two regions, as well as flags in case the fits did not work.

The NLP analysis code works similarly however does not have a separate function for running the curve fits, rather it is all inside the parallel for-loop. As noted in the thesis, NLP IV curves have significant hysteresis and should only be analyzed to

generate Ni. i.e. Ne and Te are not reliable.

WLP Analysis Code

Main WLP analysis code. This is written under filename WlpFPMUIterate.m

```matlab
1  % This is the main WLP Analysis Code
2
3  clear all  % Clears all previous variables
4  close all  % Closes all open figures. If not needed, please comment
5
6  %iteration count number
7  iterations=1;
8
9  % Some constants
10 Qe = 1.602e-19; %Electron charge (C)
11 Kb = 1.381e-23; %Boltzman constant (J/K)
12 Me = 9.109e-31; %Electron mass (kg)
13 v=7660;          %ISS velocity (m/s)
14
15
16 aWLP = 5.08e-2;    % radius of WLP (m)
17
18 load 'WLP2015173.mat';
19 % Loading data file with sweep information in the following format
20 % (Where N is the number of data sweeps)
21 % WlpCurr-Sweep Currents formed in a (2048 X N) matrix in Amperes
22 % WlpVolt-Sweep Voltages formed in a (2048 X N) matrix in Volts
23 % WlpTime-Time of sweep, formed in a (8 X N) matrix
24 % being year, day of year, hour, minute, second, millisecond, error flag, and
25 % even/odd sweep flag
26
27 % Pre-allocating arrays for results
28 count = size(WlpCurr,2); % figure out the number of sweeps in given data
29 TeWLP1 = zeros(count,iterations); %Wlp electron Temperature
30 VfWLP  = zeros(count,1); % floating potential
31 VpWLP1 = zeros(count,1); % first plasma potential derived at the max diff point
32 VpWLP2 = zeros(count,iterations);% plasma potential from fit
33 NeWLP1 = zeros(count,iterations); % electron density
34 NiWLP = zeros(count,iterations); % ion density from fitted plasma potential
35 NiWLPDiff = zeros(count,1); %Ion density from being projected to diff plasma potential
36 NiWLPVf = zeros(count,1); %Ion density using floating potential as Vp
37 bWLP = zeros(count,iterations); %Beta Value from saturation region
38 resNormWLPOml = zeros(count,iterations); %resnorm for saturation fits
39 resNormWLPHwVp = zeros(count,iterations); %resnorma for retardation fits
40 badTimes=zeros(count,1); %error flag array
41
42 % Projected 2D circular area
43 AwlpProjected = pi*aWLP^2;
44
45 % 1/2 surface area used for fitting
46 AreaWlp = 2*pi*aWLP^2;
47
48 % Prior to running WlpFPMUIterate.m file we pre-calculate the times when ISS was in
49 % sunlight and eclipse. This is done in code named DayNightAreaAppend.m
50 % which takes the ISS Time and LLA and sunlight data and creates a profile
51 % for every time stamp where the factor goes from 0 in eclipse to between
52 % 1-4 for sunlight. The reason the factor is 1-4 is because it will be
53 % later multiplied with pi*r^2. So multiplication with 1 means 2D cross
54 % section and multiplication with 4 means full surface area illuminatuion
```

```matlab
55  % with sunlight. Again, all of this is done before running the WlpFPMUIterate.m
56  % AreaInterp is a function that read above precalculated data and stores triangular factors
57  % for photoemission current at the times of each sweep being analyzed in current file.
58
59  AreasFactor = AreaInterp(WlpTime);
60
61
62  % Rrunning a parralel for-loop since each sweep is independent, each one can
63  % be fitted without previous sweep data
64  parfor runIdx=1:count
65
66      %print Idx for progress
67      runIdx
68
69      %set error flag to false (will be set to another value if error occurs)
70      badFlag=false;
71
72      % calculate photo current (will be 0 in eclipse)
73      % Area factor is either 0 for eclipse, or between 1 and 4 during day
74      % representing projected area to full surface area
75      % 29 uA/m2 is Gold's Photoemission current
76      sunLightFlux = AreasFactor(runIdx)*AwlpProjected*29E-6
77
78      % extract current sweep current and run a 7 point mean to smooth out noise
79      tempCurr  =movmean(WlpCurr(:,runIdx),7);
80
81      % Storing sweep voltage
82      runVolt = WlpVolt(:,runIdx);
83
84      % compensate for calibration step between high gain and low gain switch
85      stepIdx = find(tempCurr<-7E-6,1,'last');
86      if(~isempty(stepIdx))
87          runCurrent=movmean([WlpCurr(1:stepIdx,runIdx)+2.2E-6;WlpCurr(stepIdx+1:end,runIdx)],9);
88      else
89          runCurrent = movmean(WlpCurr(:,runIdx),9);
90      end
91
92      runCurrent = runCurrent + sunLightFlux;     % applying photo current term
93
94      [~,floatIdx] = min(abs(runCurrent));     % find floating potential
95
96      if (floatIdx >50) %Error checking for floating potential location if too low flag variable is
                 set in else statement
97          floatVoltWlp=runVolt(floatIdx);
98          sweepLen=length(runCurrent);
99
100         % find the index of Vf
101         VfIndWLP = floatIdx;
102         VfWLP(runIdx) = floatVoltWlp;
103
104         % Setting Lsqcurvefit options working as of MATLAB 2017a
105         options=optimoptions('lsqcurvefit');
106         options.TolX=1E-60;
107         options.TolFun=1E-60;
108         options.MaxIterations=500;
109         %options.StepTolerance=1E-60;
110         options.MaxFunEvals=200;
111
112
113         % Extracting data for saturation fit starting from index 1 to .6 volts below
114         % floating potential. The bsearch function is NOT native to matlab.
115         % We created it and is available with the software.
116         tempInd = bsearch(runVolt, [floatVoltWlp -.6]);
117         ionSweepCurr = runCurrent(1:tempInd);
```

```matlab
118                 ionSweepVolt = runVolt (1:tempInd );
119
120
121           % checks to make sure that the ion saturation region is entirely
122           % negative , and is increasing with voltage , if not then set a badFlag
123           if (( sum( ionSweepCurr >0))==0 && ionSweepCurr (1)<ionSweepCurr ( end )  )
124
125               % fitting line to ion saturation region
126               ionSatCurrentWLP = zeros ( sweepLen ,1) ; %declaring variable for linear current
127
128               polyCo = polyfit ( ionSweepVolt , ionSweepCurr , 1) ; % calling a linear fit function
129               lineFitCurr = polyval ( polyCo , runVolt ) ; % generating linear data
130
131               % zeroing out positive currents from the projection
132               ionSatCurrentWLP ( lineFitCurr <0)=lineFitCurr ( lineFitCurr <0);
133
134               % copying voltage / current for differential vp search
135               V_Vf = runVolt ;
136               C = runCurrent −ionSatCurrentWLP ;
137
138               % Using a diff apprach we try and find vp within 80 points of Vf
139               % Defines search range
140               VpSearchInd = ( VfIndWLP : VfIndWLP+80) ;
141
142               %calculates dI/dV
143               tempDiDv = diff (C( VpSearchInd ) )./ diff (V_Vf( VpSearchInd ) );
144
145               % finds max of dI/dV ( should be approximate location of plasma potential )
146               [Y, I] = max( tempDiDv ) ;
147               VpIndWLP = VfIndWLP + I − 1;
148
149               % Stores diff plasma potential
150               VpWLP1( runIdx ) = V_Vf( VpIndWLP ) ;
151
152
153               % using diff method guess , we find a Ni value estimate by
154               % projecting ion current to plasma potential and assuming it is a ram current
155               Ii = ionSatCurrentWLP ( VpIndWLP ) ;
156               NiWLPDiff ( runIdx ) = Ii /( AwlpProjected ∗v∗(−Qe) ) ;
157
158               % Repeating with floating potential (More for testing )
159               Ii = ionSatCurrentWLP ( VfIndWLP ) ;
160               NiWLPVf( runIdx )=Ii /( AwlpProjected ∗v∗(−Qe) ) ;
161
162               % Error flag if number is far too large ( normally caused by faulty data
163               if ( NiWLPDiff ( runIdx )>2.5E12)
164                   badTimes ( runIdx )=4;
165               end
166
167
168               % Saving Ni as first Ne guess
169               NeIn = NiWLPVf( runIdx ) ;
170
171               %initial guess for Te/Vp
172               TeIn = 2000; % in kelven
173               VpIn = 1; % Volts above floating potential
174
175               for k=1: iterations % For−loop through iterations ( currently only 1 iteation is used )
176                   % passes data to IterateWLP function
177                   [ NeWLP1( runIdx ,k) ,NiWLP( runIdx ,k) ,TeWLP1( runIdx ,k) , bWLP( runIdx ,k) , VpWLP2( runIdx ,k
178                       ) ,resNormWLPHwVp( runIdx ,k) ,resNormWLPOml ( runIdx ,k) ]=...
                           IterateWLP ( NeIn , TeIn , VpIn , VfIndWLP , runCurrent , runVolt , ionSatCurrentWLP ,
                                   ionSweepCurr , ionSweepVolt , WlpVolt (: , runIdx ) , WlpCurr (: , runIdx ) ,sum(
                                   publishSlides==runIdx ) ,k, runIdx ) ;
```

```
179                         VpIn=VpWLP2(runIdx,k);
180                         TeIn=TeWLP1(runIdx,k);
181                         NeIn=NeWLP1(runIdx,k);
182                     end
183
184
185             else
186                 %error flag for deformed ion saturation region
187                 badTimes(runIdx)=2;
188             end
189         else
190             %error for low bias floating potential
191             badTimes(runIdx)=1;
192
193     end
194
195 end
196 %save all output variables
197 save('FullOutEnd173.mat','NiWLP','TeWLP1','NeWLP1','NeWLP2','bWLP','WlpTime','VpWLP2','badTimes','
        resNormWLPOml','resNormWLPHwVp','NiLower','NiWLPVf','NiWLPDiff','VfWLP')
```

WLP Fitting function

This is a function for running curve fits on the WLP data, and is written under a
filename IterateWLP.m

```
 1 function [Ne,Ni,Te,beta,Vp,resnormVP,RESNORMOML1] = IterateWLP(NeIn,TeIn,VpIn,VfIndWLP,runCurrent,
        runVolt,ionSatCurrentWLP,ionSweepCurr,ionSweepVolt,WlpVolt,WlpCurr,plotThings,iterNum,runIdx)
 2 %Function that fits for Te and Vp from retardation region, and Ne and Beta
 3 %from saturation region
 4
 5
 6 %Define constants
 7 aWLP = 5.08e-2;   % radius of WLP (m)
 8 AwlpProjected=pi*aWLP^2; %projected circular area of WLP (m^2)
 9
10
11 %half of circular area of WLP (m^2) for retardation/saturation fits
12 AreaWlp=2*pi*aWLP^2;
13
14 %Ion amu (for oxygen)
15 ion=16;
16
17 %amu mass (kg)
18 Mi=1.66054E-27;
19
20 %ISS velocity (km/s)
21 v=7660;
22
23 %Elementary charge (C)
24 Qe = 1.602e-19;
25
26 %Storing Floating potential
27 floatVoltWlp=runVolt(VfIndWLP);
28
29 %Set lsqcurvefit options
30 %Working matlab 2017a
31 options=optimoptions('lsqcurvefit');
```

```matlab
32   options.TolX=1E-60;
33   options.TolFun=1E-60;
34   options.MaxIterations=500;
35   options.MaxFunEvals=1000;
36
37
38
39
40   %   4. Use Ni as a good guess, and fit for Te and Vp
41   % Parameters for fitting
42   %          Te        Vp
43   norm1   = [ 1e3,   .1]; %Normalization values for curve fit (K, V)
44   guess   = [   TeIn/norm1(1),    VpIn/norm1(2)];
45   lb      = [   .5,    2];
46   ub      = [   8,    25];
47
48   %Copying Voltage/current variables for retardation fit region
49   V_Vp_Te = runVolt;
50   C_Vp_Te =runCurrent;
51
52   %set number of points around floating potential for retardation fit
53   downidx=15; %points below floating potential
54   upidx=7;    %points above floating potential
55
56   %extracting the voltage and current values in the retardation fit region
57   xData = V_Vp_Te(VfIndWLP-downidx:VfIndWLP+upidx)-V_Vp_Te(VfIndWLP);
58   yData = C_Vp_Te(VfIndWLP-downidx:VfIndWLP+upidx);
59
60   %Array of values passed in (Number Density guess, effective surace area,
61   %prjected area, ion mass, and station velocity)
62   x = [NeIn AreaWlp AwlpProjected ion*Mi v];
63
64   %Call curve fit for retardation region, returns fitted Te and PP values in
65   %Array X (PP measuered relative to floating potential)
66   [X,RESNORMVP,RESIDUAL,EXITFLAG,OUTPUT] = lsqcurvefit('HWIonVp', guess, xData, ...
67       yData, lb, ub, options, norm1, x);
68   resnormVP=RESNORMVP;
69   resnormVP
70
71
72   %Saving variables
73   Te = X(1)*norm1(1); %Temperature in Kelvin
74
75   %Plasma plasma potentail relative to station
76   tempVp = V_Vp_Te(VfIndWLP) + X(2)*norm1(2);
77
78   %Save index of closest bias to floating potential
79   VpIndWLP = bsearch(V_Vp_Te,tempVp);
80
81
82   % Now use the new Vp to get Ni from IiSat
83   Ii = ionSatCurrentWLP(VpIndWLP); %finding projected current at new plasma potential
84   Ni= Ii/(AwlpProjected*v*(-Qe)); %calculate ion density assuming purely ram current ion current at
         Vp
85
86
87   %Copying variables for Saturation fit region
88   V_beta_Ne = WlpVolt;
89   C_beta_Ne =movmean(WlpCurr,7);
90
91
92   % Now we shall find Ne by fitting in electron saturation region
93   % using the Vp we just found
94   % Parameters for fitting
```

```
95   %            Beta    Ne
96   norm   = [   .1,   NeIn]; %normalization values for fit (unitless, m−3)
97   guess  = [  4,       1];  %guess values
98   lb     = [2.5,      .05]; %lower fit bounds
99   ub     = [  10,      20]; %upper fit bounds
100
101
102  %Finding index for beta/NE fit 2.5 volts above floating to 4.25 volts
103  tempInd = bsearch(runVolt, [floatVoltWlp+2.5 floatVoltWlp+4.25]);
104
105  %extracting indecies into independent variables
106  z1 = tempInd(1);
107  z2 = tempInd(2);
108
109  %z1=1800;
110  %z2=2048;
111
112  %extracting data for beta/Ne fit
113  xData = V_beta_Ne(z1:z2)−tempVp;% subtracting out plasma potential to center axis
114  yData = C_beta_Ne(z1:z2);%+2.2E−6;
115
116  %Input variable array for saturation fit (Temperature from fit(K), half
117  %surface area value)
118  x = [ Te, AreaWlp];
119
120  %Call lsqcurve fit for saturation region, to solve for Ne/beta
121  [X,RESNORMOML1,RESIDUAL,EXITFLAG,OUTPUT] = lsqcurvefit('OMLSaturation', guess, xData, ...
122      yData, lb, ub, options, norm, x);
123
124  %Saving variables
125  Ne= X(2)*norm(2); %Electron density from saturation fit
126
127  beta= X(1)*norm(1); %Saturation curve Beta value
128
129  RESNORMOML1
130
131
132
133  end
```

NLP Analysis Code

Main NLP analysis code written under a filename NlpFPMUAnalysisFixAppend.m

```
1   clear
2   close all
3
4   % Loading in sweep data
5   load('NLP2015174_Fixed.mat')
6
7   % Loading data file with sweep information in the following for mat
8   % (Where N is the number of data sweeps)
9   % NlpCurr−Sweep Currents formed in a (N X 512) matrix in Amperes
10  % NlpVolt−Sweep Voltages formed in a (N X 512) matrix in Volts
11  % NlpDate−Time of sweep, formed in a (N X 512) matrix
12  % being year, day of year, hour, minute, second, millisecond, error flag, and
13  % even/odd sweep flag
14
15
```

```matlab
16  % Defining fundemental constants
17  ec=1.602e−19; % electron charge (C)
18  me=9.109E−31; % Electron mass (kg)
19  boltzConst=1.38E−23; % Boltzmann const J/K
20
21
22
23  count=length(NlpVolt(:,1));
24  % Simulation parameters
25  ionMass=2.6568E−26; % Ion mass (currently for O)
26  ionQ=1∗ec; % Ion charge (currently singly ionized)
27  Qe=ec;
28  ion=16; % Ion AMU
29  Mi=1.66054e−27; % Amu mass (kg)
30
31  % Upper and lower bound for saturation fit region from floating potential
32  lowVoltPlus=2;
33  highVoltPlus=2.5;
34
35
36  aNLP = 1.43e−2;    %  radius of NLP (m)
37  LNLP = 5.08e−2;    %  Length of NLP (m)
38  Anlp = pi∗aNLP∗LNLP; % 1/2 surface area for fits (m^2)
39  RAMAngle = 8; %  Angle to RAM direction in degrees for day 62, 2007
40  AnlpProjected = LNLP∗cos(RAMAngle∗pi/180)∗2∗aNLP; % 2d projected Area
41
42  % Iss velocity m/s
43  v=7400;
44
45
46  % Declaring varaiables
47  numRec=size(NlpVolt,1); % length of data set
48
49  TeNLP1=zeros(numRec,1);% Te values
50  NiNLP=zeros(numRec,1);%  Ni values
51  VpNLP1=zeros(numRec,1); % vp Values
52  VpNLP2=zeros(numRec,1);% vp fit values
53  bNLP=zeros(numRec,1); % beta values
54  NeNLP1=zeros(numRec,1); % Electron density values
55  VfNLP=zeros(numRec,1); %  floating potential values
56  exitFlagNLP2 =zeros(numRec,1); %  exit flag from fit
57  resNormNLP2=zeros(numRec,1); % R2 value from fit
58
59
60  % lsqcurve fit options (Working as of MATLAB 2017a)
61  options=optimoptions('lsqcurvefit');
62  options.OptimalityTolerance=1E−60;
63  options.FunctionTolerance=1E−25;
64  options.StepTolerance=1E−25;
65  options.MaxFunctionEvaluations=3700;
66  % options.Algorithm = 'levenberg−marquardt';
67  options.MaxIterations = 4000;
68
69  % Declaring array for error flags
70  badrunsNLP=zeros(numRec,1);
71
72  % NLP sweep length
73  sweepLenNLP=512;
74
75  % Sunlight flux, Projected area∗orbit factor(0 for eclipse, 1−pi in light)
76  % and 29 microamp/m^2 photo current for gold
77  sunLightFlux=aNLP∗LNLP∗2∗2.9E−5∗AreaInterpNLP(NlpDate);
78
79
```

```matlab
80  % We sweep through each sweep in the data set analyzing each sweep
81  % A parralel for loop is used here, as each sweep
82  parfor k=1:count
83      k
84      % fprintf(2,'Here % d\n',k)
85
86      % extracting sweep voltages
87      voltageNLP=NlpVolt(k,:); % Sweep voltage
88      currentNLP =NlpCurr(k,:)+sunLightFlux(k); % Sweep Current
89      myTemp=min(abs(movmean(currentNLP,15))); % Error checking
90
91      % checking for data irregularities and setting error flag if they are
92      % found
93      if(currentNLP(1)<0 && currentNLP(end)>sunLightFlux(k) &&  myTemp~=−1)
94
95
96
97
98          % finding floating potential
99          [~,VfIndNLP]=min(abs(movmean(currentNLP(130:end),15)));
100         VfIndNLP=VfIndNLP+129;
101
102         VfNLP(k) = voltageNLP(VfIndNLP);
103
104
105         % ======================================================
106         %  First we find Te and Ne from NLP
107
108
109         %    1. Find Vf Index
110         % VfIndNLP = bsearch(voltageNLP,VfNLP(k));
111         NiNLP(k)=0;
112         % verifying that the floating potential is in between the accepted
113         % bounds
114         if   (VfIndNLP>80) && (VfIndNLP<420)
115
116             % finding the up bounds on the ion fit region
117             tempInd2 = bsearch(voltageNLP,VfNLP(k)−3);
118             % z1 = tempInd2;
119             z1=1;
120             z2 = tempInd2;
121             ionCurrent=movmean(currentNLP,20);
122
123
124             % fitting to a line to the saturation region to project with
125             p = polyfit(voltageNLP(z1:z2), ionCurrent(z1:z2), 1);
126             a = polyval(p,voltageNLP);
127
128             ionSatCurrentNLP=zeros(1,sweepLenNLP);
129
130             % assigning values from the linear fit projection to a variable
131             % to represent ion current, and zeroing out positive currents
132             ionSatCurrentNLP( a < 0) = a( a < 0);
133             ionSatCurrentNLP( a >= 0)=0;
134
135
136             % Calculating electron currents by removing ion current from
137             % total
138             currentENLP = currentNLP − ionSatCurrentNLP;
139
140
141             % Assigning vairables for vp search using differential method
142             V = voltageNLP;
143             C = movmean(currentNLP,15);
```

```
144
145                   %  we look for Vp between Vf and 40 points from Vf
146                   VpSearchInd = (VfIndNLP:VfIndNLP+40);
147                   tempDiDv = diff(C(VpSearchInd))./diff(V(VpSearchInd));
148
149                   % finding where dI/dV is maximum and letting it be Vp
150                   [Y, I] = max(tempDiDv);
151                   VpIndNLP = VfIndNLP + I - 1;
152                   VpNLP1(k) = V(VpIndNLP);
153
154                   % we then take the projection to the estimated floating
155                   % potential for an approximate Ni value
156                   Ii = ionSatCurrentNLP(VpIndNLP);
157                   NiNLP(k) = Ii/(AnlpProjected*v*(-Qe));
158
159
160                   % NiNLPtemp = Ii/(AnlpProjected*v*(-Qe));
161                   %    4. Use Ni as a good guess, and fit for Te and Vp in the
162                   %     vicinity of Vf
163                   %  Parameters for fitting
164                   %          Te      Vp
165                   norm   = [ 1e3,   .1]; % normalization factors for Temperature(k) and plasma potential(
                            V)
166                   guess  = [    2,    2]; % guess values for fit
167                   lb     = [  .5,   .4]; % lower bound on fit values
168                   ub     = [    5,   20];% upper bound on fit values
169
170
171                   % Assigning temp vars for fit
172                   V = voltageNLP;
173                   C = movmean(currentNLP,5);
174
175                   % finding floating potential with photoelectri effect in account
176                   [~,nonPhotoVf]=min(abs(movmean(currentNLP(130:end),15)-sunLightFlux(k)));
177                   nonPhotoVf=nonPhotoVf+129;
178
179
180                   % We fit in the region 15 pointes below and 7 points above
181                   xData1 = V(nonPhotoVf-15:nonPhotoVf+7)-V(VfIndNLP);
182                   yData1 = C(nonPhotoVf-15:nonPhotoVf+7);
183
184
185                   % fitting in retardation region using lsqcurve fit
186                   x = [NiNLP(k) Anlp AnlpProjected ion*Mi v];
187                   [X,RESNORM,RESIDUAL1,EXITFLAG,OUTPUT] = lsqcurvefit('HWIonVp', guess, xData1, ...
188                        yData1, lb, ub, options, norm, x);
189
190
191
192                   % saving variables
193                   TeNLP1(k) = X(1)*norm(1); % Electron temperature
194                   tempVp = V(VfIndNLP) + X(2)*norm(2); % plasma potential
195                   VpIndNLP = bsearch(V,tempVp); % plasma potential index
196                   VpNLP2(k) = tempVp; % storing fitted plasma potential
197
198                   resNormNLP1(k) = RESNORM;
199
200                   %
201
202                   %  Now, use the new Vp to get better Ni from IiSat
203                   Ii = ionSatCurrentNLP(VpIndNLP); % finding ion current at plasma potnetial
204
205                   % treating ion current at plasma potential as ram current
206                   NiNLP(k) = Ii/(AnlpProjected*v*(-Qe));
```

```matlab
207
208
209
210
211              % We now use our Te/Vp values to fit for Ne and Beta
212
213              % We fit the last segment of the curve, from 420 til 15 points
214              % from end, (since it is centered around floating potential we can use a fixed region)
215              xData = V(420:end−15)−VpNLP2(k);
216              yData = C(420:end−15)−sunLightFlux(k);
217
218
219              %              beta       Ne
220              norm1  = [  .1,  NiNLP(k)]; % Normalization factors for fits
221              guess  = [   5,         1]; % guess values for fits
222              lb     = [  2.5,       .05]; % lower bound on fit
223              ub     = [   10,        20]; % upper bound on fit
224
225
226              x = [Anlp ,TeNLP1(k)]; % putting area and temperature in a variable to pass for fit
227              [X,RESNORM,RESIDUAL,EXITFLAG,OUTPUT] = lsqcurvefit('OMLSaturation', guess, xData, ...
228                  yData, lb, ub, options, norm1, x);
229              X(2)∗norm1(2) % displaying Ne calculated value (Can be commented if not need to be
                        shown)
230              NeNLP1(k) = X(2)∗norm1(2); % Storing Ne in save array
231              bNLP(k) = X(1)∗norm1(1);   % Storing beta in save array
232
233              exitFlagNLP2(k) = EXITFLAG;
234
235
236              resNormNLP2(k) = RESNORM; % Saving resnorm from Ne/beta fit
237
238          end
239
240      else
241          %Setting error flag for deformed data
242          badrunsNLP(k)=1;
243      end
244      close all
245  end
246
247  % Saving variables
248  save(sprintf('NLP% d% dFrom2_3PhotoSaw2pirl.mat',NlpDate(1,1),NlpDate(1,2),lowVoltPlus,
        highVoltPlus),'EvenOdd','NlpDate','badrunsNLP','bNLP','NeNLP1','NiNLP','TeNLP1','VpNLP1','
        VpNLP2','VfNLP')
```

Electron Retardation HW function

Model based on Hogey-Wharton paper for electron retardation region written under a filename HWIonVp.m

```matlab
1  % Hoegy Wharton Cylinder Model for Ion Saturation Current
2  % Based on formulas taken from Hoegy Wharton's Paper
3  %
4  % Input Parameters (PP,w,x)
5  %   PP : A 1x2 vector containing following Plasma Parameters in MKSA
6  %          PP(1) = ne    Electron Density in /m3
7  %          PP(2) = Te    Electron Temperature in degree Kelvin
```

```matlab
8   %   V : A vector containing sweep in volts relative to Plasma Potential
9   %       i.e. at Vp, V = 0
10  %   x : A 1x2 vector containing following characteristic lengths in METERS
11  %           x(1) = probe projected area  in meter^2
12  %           x(2) = Ion Mass in Kg
13  %           x(2) = Spacecraft Velocity in meters/sec
14
15  function  I = HWIonVp(PP,V,norm,x)
16
17  %apply normalized multiplier
18  PP = PP.*norm;
19  Te = PP(1);
20  Vp = PP(2);
21
22  %extract variables
23  ne = x(1);
24  A = x(2);
25  Ap  = x(3);
26  Mi = x(4);
27  v   = x(5);
28
29  V = V-Vp;
30
31  %define constants
32  kB = 1.381e-23;  %boltzmann const. (J/K)
33  e  = -1.602e-19; %electron charge (C)
34  me = 9.109e-31;  %electron mass (kg)
35
36
37  nAev = ne*Ap*e*v; %ion ram current term
38
39  %Calculate Current
40  Ajre = (Ap)*(-e)*ne*sqrt(kB*Te/(2*pi*me));
41  Ii = nAev  + Ajre*exp((-e)*V/(kB*Te));
42
43  %Storing current value in return variable
44  I = Ii;
```

Electron Saturation OML function

Model for electron saturation region based on OML equations written under filename
OMLSaturation.m

```matlab
1   %  OML Cylinder Model
2   %  Based on formulas taken from Chen's work
3   %
4   %  Input Parameters (PP,w,x)
5   %    PP : A 1x3 vector containing following Plasma Parameters in MKSA
6   %           PP(1) = s     Sheath size
7   %           PP(2) = ne    Electron Density in /m3
8   %           PP(3) = Te    Electron Temperature in degree Kelvin
9   %    V : A vector containing sweep in volts relative to Plasma Potential
10  %        i.e. at Vp, V = 0
11  %    x : A 1x2 vector containing following characteristic lengths in METERS
12  %           x(1) = a      radius in meters
13  %           x(2) = L      Length in meters
14
15  function  I = OMLSaturation(PP,V,norm,x)
```

```
16
17  PP = PP.*norm; % Applying  normalization  values
18  %   s = PP(1);
19  % Extracting  data  into  specific  variables
20  b = PP(1);
21  ne = PP(2);
22
23  % Assigning  temperature  and  area  seperate  variables
24  Te = x(1);
25  A = x(2);
26
27  % Defining  constants
28  kB = 1.381e-23; % Blotzmann  const.  (J/K)
29  e  = 1.602e-19; % electron  charge  (C)
30  me = 9.109e-31; % electron  mass  (kg)
31
32  % Calculating  saturation  current
33  Ajre = A*e*ne*sqrt(kB*Te/(2*pi*me));
34  eta = e*(V)/(kB*Te);
35  Ie = Ajre*((1+eta).^b);
36
37  % returning  current
38  I = Ie;
```

Photo electric Area interpolation function

Function that interpolates the multiple of the projected area that is to be used for calculating photoelectric current, written under the filename AreaInterp.m

```
1
2   % Outputs  ratio  of  projected  area  for  WLP
3   function  [ AreasWlp ] = AreaInterp(WlpTime)
4
5   % load  pre  made  ratio  data
6   load  AreaSunWLP.mat
7
8   % Convert  date  into  a  matlab  serial  date
9   dateWLP=datenum(WlpTime(1,:),0,WlpTime(2,:),WlpTime(3,:),WlpTime(4,:),WlpTime(5,:));
10
11  % lineraly  interpolate  area  data  from  the  ISS  data  time  to  the  WLP  Time
12  [dateInterp, index] = unique(dateInterp);
13  AreasWlp=interp1(dateInterp,areaSun(index),dateWLP);
14
15  end
```

Binary Search function

Function performs a binary search for the closest number in a set to an input value, by performing binary search algorithm. Written under filename bsearch.m

```
1   %  bsearch(x,var)
2   %  Written  by  Aroh  Barjatya
```

```matlab
3   %   Binary  search  for  values  specified  in  vector  'var '  within  data  vector  'x '
4   %   The  data  has  to  be  pre−sorted  in  ascending  or  decending  order
5   %   There  is  no  way  to  predict  how  the  function  will  behave  if  there
6   %   are  multiple  numbers  with  same  value .
7   %   returns  the  index  values  of  the  searched  numbers
8
9   function  index  =  bsearch (x , var )
10  index=−1; % Preset  value
11  xLen  =  length (x ); % Find  length  of  input  for  search  through
12
13  [xRow  xCol ]  =  size (x ); % find  dimensions  of  input  array
14
15  if  x(1)  >  x(xLen)         %   means  x  is  in  descending  order
16      if  xRow==1
17          x  =  fliplr (x );   %  reverses  array  direction
18      else
19          x  =  flipud (x );
20      end
21      flipped  =  1;
22  elseif  x(1)  <  x(xLen)    %   means  x  is  in  ascending  order
23      flipped  =  0;
24  else % first  and  last  element  have  same  value ,  thus  binary  search  fails
25      'badly  formatted  data .  Type  ''help  bsearch\'')';
26      return ;
27  end
28
29
30  for  i  =  1: length ( var ) % searching  through  each  input  for  index
31      low  =  1; % Start  with  entire  array
32      high  =  xLen ;
33      if  var ( i )  <=  x( low )
34          index ( i )  =  low ; % if  value  is  lower  than  lowest  point  return  first  index
35          continue ;
36      elseif  var ( i )  >=  x( high )
37          index ( i )  =  high ; % if  value  is  higher  than  highest  point  return  last  index
38          continue ;
39      end
40      flag  =  0;
41      while  ( low  <=  high )
42          % find  mid  point  between  high / low  search  bars
43          mid  =  round (( low  +  high )/2);
44
45          % if  search  value  is  in  the  lower  half  of  the  search  range
46          % Then  repeat  with  lower  half  of  the  region
47          if  ( var ( i )  <  x( mid ))
48              high  =  mid ;
49          % if  search  value  is  in  the  upper  half  of  the  search  range
50          % Then  repeat  with  upper  half  of  the  region
51          elseif  ( var ( i )  >  x( mid ))
52              low  =  mid ;
53          % if  search  value  is  at  the  mid  point ,  then  use  value  as  index
54          else
55              index ( i )  =  mid ;
56              flag  =  1;
57              break ;
58          end
59          if  ( low  ==  high  −  1)
60              break
61          end
62
63      % if  value  was  found  at  a  mid  point ,  continue  to  next  value
64      if  ( flag  ==  1)
65          continue ;
66      end
```

```
67     % if upper and lower indecies are equal, let index equal that value
68       if (low == high)
69           index(i) = low;
70
71       % if the higher value is closer to seach value than the lower, then assign
72       % higher index
73       elseif ((x(low) - var(i))^2 > (x(high) - var(i))^2)
74           index(i) = high;
75
76       % If lower is closer, assign the lower index value
77       else
78           index(i) = low;
79       end
80       end
81
82  % if array was flipped first, adjusting the index to be consistent with the
83  % original array orientation
84  if flipped
85      index = xLen - index + 1;
86  end
87  % set index to error value -1 if errors occured and index/var do not exist
88  if ~exist('index','var')
89      index=-1;
90  end
91  end
```

Photo electric Area interpolation function

Code that creates area factor profile for WLP/NLP, written under a filename DayNightAreaAppend.m

```
1
2  % load Lat/long/alt/sunlight/time data from ISS file
3  % Data is stored as IssLLA-[ N X 3 ] array containing lat,long and alt as
4  % each column
5  % IssSun-% sunlight of ISS stored as an [ N X 1 ] array
6  % IssTime-date stamp of data stored as YYYYDDD.HHMMSS decimal
7  load IssTimeLLA.mat
8
9  upperboundfactor=pi; % set pi for NLP, 4 for WLP
10
11
12  % Parsing data data and storing in individual variables as strings
13  StrDate=num2str(IssTime*1E6);
14  StrYear=StrDate(:,1:4);
15  StrDay=StrDate(:,5:7);
16  StrHour=StrDate(:,8:9);
17  StrMinute=StrDate(:,10:11);
18  StrSecond=StrDate(:,12:13);
19
20  % Converting date into [Y,D,H,M,S] form
21  DateArray=[str2num(StrYear),str2num(StrDay),str2num(StrHour),str2num(StrMinute),str2num(StrSecond)
           ];
22
23  % converting to serial date
24  dateInterp=datenum(DateArray(:,1),0,DateArray(:,2),DateArray(:,3),DateArray(:,4),DateArray(:,5));
25
26  % Converting % sunlight into a simple true/false flag
```

```
27  Sunthresh=IssSun >50;
28
29  % Locating eclipse/sunlight transitions
30  diffDay=abs(diff(Sunthresh));
31  DayShift=find(diffDay==1);
32
33  % Variable for storying data
34  areaSun=zeros(size(IssSun));
35
36  % Go through each pair of day/night transistions
37  for k=1:length(DayShift)-1
38
39      % locate the point half way between the transitions
40      pointMid=floor(mean([DayShift(k+1),DayShift(k)]));
41
42      % using a linspace, creating an upward line to the center starting from
43      % 1 to the upper bound
44      areaSun(DayShift(k):pointMid)=linspace(1,upperboundfactor,pointMid-DayShift(k)+1)*Sunthresh(
                pointMid);
45
46      % using a linspace, creating an downward line from the center to the eclipse starting from
47      % upper bound going to 1
48      areaSun(pointMid:DayShift(k+1))=linspace(pi,upperboundfactor,DayShift(k+1)-pointMid+1)*
                Sunthresh(pointMid);
49  end
50
51  save('AreaFactorNLP','areaSun','dateInterp')
```

mNLP Analysis Code

Code that simulates mNLP data and compares the two methods mentioned in chapter 3, written under filename varyBetaAppend.m

```
1   close all
2   clear
3
4
5   %  Defining fundemental constants
6   ec=1.602e-19;  % Electron charge (C)
7   me=9.109E-31;  % Electron mass (kg)
8   boltzConst=1.38E-23; % Boltzmann const. (J/K)
9
10
11  % Simulation parameters
12  ionMass=2.6568E-26; % Ion mass (currently for O)
13  ionQ=1*ec; % Ion charge (currently singly ionized)
14
15  % Define Densities (must be equal to satisfy quasi-neutrality)
16  ni=1E11; %  Ion density (m^-3)
17  ne=1E11; %  Electron Density (m^-3)
18
19  % Beta Sweep parameters
20  betaMin=0.45; % Starting Beta
21  betaMax=0.85; % Max Beta
22  betaStep=0.01; % step size
23  betaTemp=betaMin:betaStep:betaMax; % Create array of values for Beta
24
25  % Species Temperatures
```

```matlab
26  Ti=1200; % Ion Temperature (K)
27  Te=1200; % electron Temperature (K)
28
29
30  % Probe area
31  probeR=0.255E-3; % probe radius (m)
32  probeL=25E-3; % Prone length(m)
33  probeA=2*pi*probeR*probeL; % probe area (m^2)
34
35  A=probeA;
36
37  % Plasma Potential
38  plasmaPotential=2.5; % Volts
39
40  % Define voltage values for data to be generated
41  voltMin=-10; % lowest voltage
42  voltMax=10;  % Highest voltage
43  voltStep=0.01; % Voltage step
44  voltAxis=voltMin:voltStep:voltMax; % create array variable with all voltage values
45
46  % Fixed Probe Biases
47  voltArray=[4,5.5,10]; % 3 needle mNLP
48  tempGuesses=[800,1200,1600,2400]; % Temperatures to cycle through
49  voltIdxs=bsearch(voltAxis,voltArray); % find indecies of the probe biases in
50  % generated data
51
52
53  % Thermal Current Calculations
54  % Ion thermal current
55  iThermalI=ni*ionQ*A*sqrt(boltzConst*Ti/(2*pi*ionMass));
56
57  % electron thermal current
58  iThermalE=ne*(ec)*A*sqrt(boltzConst*Te/(2*pi*me));
59
60  % Allocating arrays
61  % total ion current
62  iProbeI=zeros(size(voltAxis));
63
64  % total electron current
65  iProbeE=zeros(size(voltAxis));
66
67  % % calculating currents
68
69
70  % Finding where the probe bias is positive and negative relative to the
71  % plasma potential
72  posBias=(voltAxis-plasmaPotential)>0;
73  negBias=~posBias;
74
75
76  % Retardation Currents (i.e. for electrons where bias is less than plasma potential, and reverse
        for ions)
77  iProbeI(posBias)=-iThermalI*exp(-ionQ*(voltAxis(posBias)-plasmaPotential)/(boltzConst*Ti));
78  iProbeE(negBias)=iThermalE*exp(ec*(voltAxis(negBias)-plasmaPotential)/(boltzConst*Te));
79  figure;
80  plotTypes={'.-','--','--.',':'};
81
82  neExps=zeros(length(betaTemp),length(tempGuesses));
83
84  % Beta Fit Params
85  for k=1:length(tempGuesses)
86  TempGuess=tempGuesses(k);
87  plasmaPot=0;
88
```

```matlab
89   % Working as of MATLAB 2017a
90   options=optimoptions('lsqcurvefit');
91   options.TolX=1E-60;
92   options.TolFun=1E-60;
93   % options.StepTolerance=1E-60;
94   options.MaxFunEvals=4000;
95   options.MaxIter = 4000;
96
97   % %   Varying beta
98
99   % Loop counter
100
101  % Allocating storage for calculated ne values
102
103  neExp=zeros(length(betaTemp),1);
104  neFit=zeros(length(betaTemp),1);
105  VpFit=zeros(length(betaTemp),1);
106  neExpNoise=zeros(length(betaTemp),1);
107  neFitNoise=zeros(length(betaTemp),1);
108  betaFit=zeros(length(betaTemp),1);
109  R2Vals=zeros(length(betaTemp),1);
110  RVal=zeros(length(betaTemp),1);
111  voltPlot=linspace(-10,10,1000);
112
113  % For loop going through beta values
114  plotCurr=false;
115
116  for count=1:length(betaTemp)
117      beta=betaTemp(count);
118
119      % Saturation currents (i.e. for electrons where bias is greator than plasma potential, and
               reverse for ions)
120      iProbeI(negBias)=-iThermalI*(1-(ionQ*(voltAxis(negBias)-plasmaPotential)/(boltzConst*Ti))).^
               beta;
121      iProbeE(posBias)=iThermalE*(1+((ec)*(voltAxis(posBias)-plasmaPotential)/(boltzConst*Te))).^
               beta;
122
123      % Adding
124      iArray=iProbeE+iProbeI;
125
126
127      % Squaring current for this method
128      i2Array=iArray(voltIdxs).^2;
129
130      % line fit to find slope coeficient
131      fitCoef=polyfit(voltArray,i2Array,1);
132      fitCurrents2=polyval(fitCoef,voltArray);
133
134      % Calculating electron density according to
135      CVal=(ec^(3/2))*sqrt(1/2/me/pi);
136      neExp(count)=sqrt(fitCoef(1))/CVal/A;
137
138
139      % Calculating R^2 value
140      % finding difference between data poitns, and fitted points
141      dataResidual=i2Array-fitCurrents2;
142
143      % Squaring the difference and calculating R^2
144      squaredRes=sum(dataResidual.^2);
145      SStotal = (length(i2Array)-1) * var(i2Array);
146      R2Vals(count)=1-squaredRes/SStotal;
147
148      % Calculating Coeficient of coorolation
149      RValMat=corrcoef(voltArray',i2Array');
```

```matlab
150         RVal(count)=RValMat(1,2);
151
152         %                       Ne    beta    Vp
153         InitialGuess=[   10E9,   0.5,    1]; % Initial guess for values
154         norm1   =     [    1E9,   0.1,    1]; % normalization for fit
155         lowBounds =   [   1E-1,     4,    0]; % lower bound for fit
156         upperBounds=  [    1E4,    10,    4]; % upper bound for fit
157         TempArea=[TempGuess,probeA];       % input array of fixed values, (Temp and area)
158
159         % Calling lsqcurve fit using OML equation function to solve for NE Beta
160         % and VP
161         [NeBetavp,residual]=lsqcurvefit('normLeastFuncNeBeta',InitialGuess./norm1,voltArray,iArray(
                voltIdxs),lowBounds,upperBounds,options,norm1,TempArea);
162         neFit(count)=NeBetavp(1)*norm1(1);
163         betaFit(count)=NeBetavp(2)*norm1(2);
164         VpFit(count)=NeBetavp(3)*norm1(3);
165
166
167         % plotting certain IV curves
168         if(mod(beta,0.1)==0&&plotCurr)
169             figure;
170             plot(voltAxis,iArray)
171             xlabel('Probe Bias(V)')
172             ylabel('Simulated Current (A)')
173         end
174
175         % Saving Ne values
176         neExps(:,k)=neFit;
177     end
178
179
180 end
181
182 % Calculating %  error in determined values
183 NeErrors=abs(neExps-ne)/ne*100;
184 save('fittedNe.mat','NeErrors','neExps','ne','tempGuesses','Te')
185
186 % %  Plotting stuff
187
188 load('fittedNe.mat')
189 plotTypes={':','-','--.','-.'};
190
191 % plotting errors with different temperaature guesses
192 for k=1:length(tempGuesses)
193 plot(betaTemp,NeErrors(:,k),plotTypes{k},'color','black','LineWidth',2);
194 hold on
195 end
196
197 %   hold on;
198 % %   neFitErrorNoise=(abs(neFitNoise-ne)/ne)*100;
199 %   plot(betaTemp,neFitErrorNoise,'r');
200 title('Beta Fit Method Error') % giving plot title
201 set(gca,'fontsize',14) % setting font size
202 grid on % adding grid to plot
203 xlabel('beta value') % labeling x axis
204 ylabel('Percent Error') % labling y axis
205 set(gcf,'color','white'); % setting background color to whie
206
207 legend('800K','1200K','1600K','2400K') % adding legend listing temperatures
```

mNLP OML function

Function based on OML equations written for mNLP analysis that has Electron Density, beta, and plasma potential as inputs to be fitted for. Written under filename normLeastFuncNeBeta.m

```matlab
1   function [ diff ] = normLeastFuncNeBeta( valMat,X,norm, TempArea)
2   % normLeastFuncNeBeta−OML based function that returns current values ,
3   % inputs to be solved for using a fit are Ne/beta/Vp, while temperature and
4   % area are passed in as constants
5
6
7   % Define V
8   ec=1.602e−19; % Electron charge (C)
9   me=9.109E−31; % electron mass (kg)
10  boltzConst=1.38E−23; % boltzmann const.(J/K)
11
12  % plasmaPotential=0;
13  valMat=valMat.*norm; % Applying normalization for fit
14
15  % Extra
16  neGuess=valMat(1);    % Electron density value (m^−3)
17  betaGuess=valMat(2);  % Beta value
18  VpGuess=valMat(3);    % Plasma potential value (Volts)
19
20
21  TGuess=TempArea(1);   % Temperature input value (K)
22  A=TempArea(2);        % probe area value
23
24  % Calculating thermal current term
25  iThermal=neGuess*(ec)*A*sqrt(boltzConst*TGuess/(2*pi*me));
26
27  % Saturation current calculation to be returned
28  diff=iThermal*(1+ec*(X−VpGuess)/boltzConst/TGuess).^betaGuess;
29
30
31  end
```

Appendix B

A day's worth of sweeps from day 174, 2015
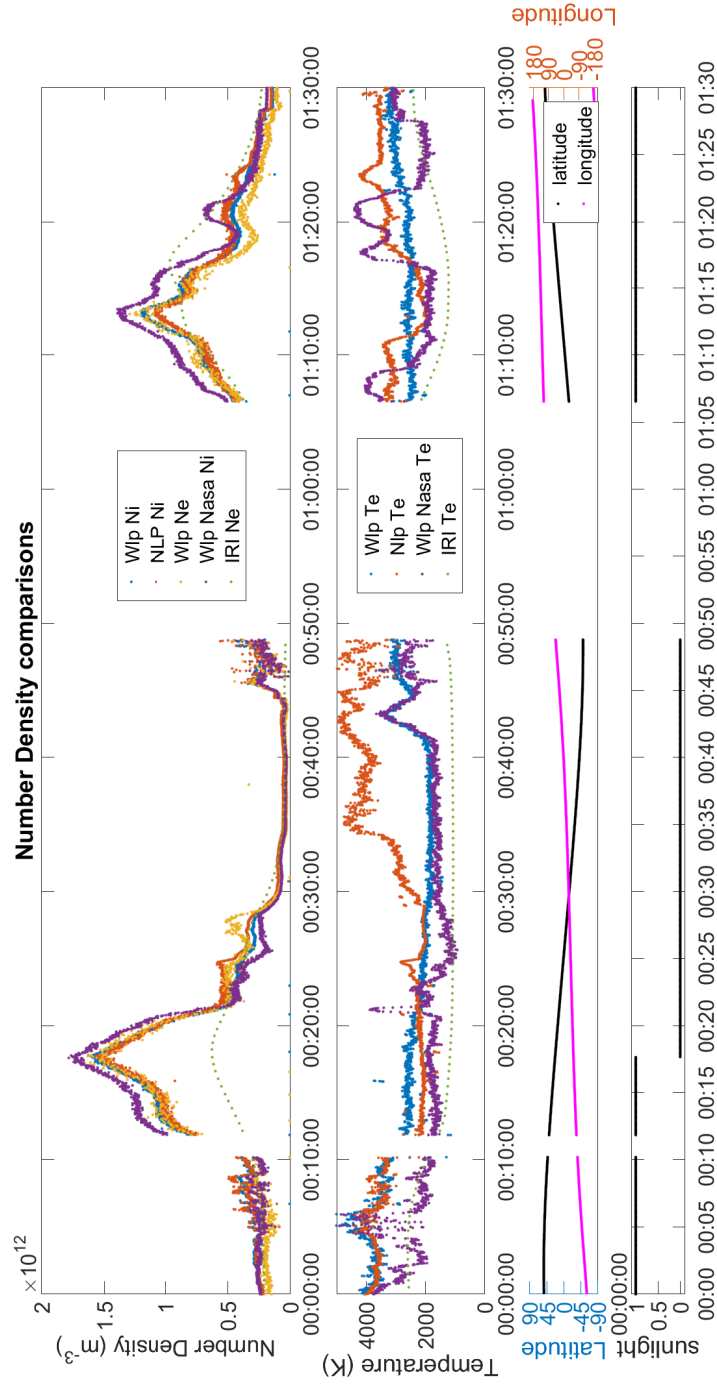
Figure B.1: A One and a half hour period during day 174 of 2015 number density and temperature
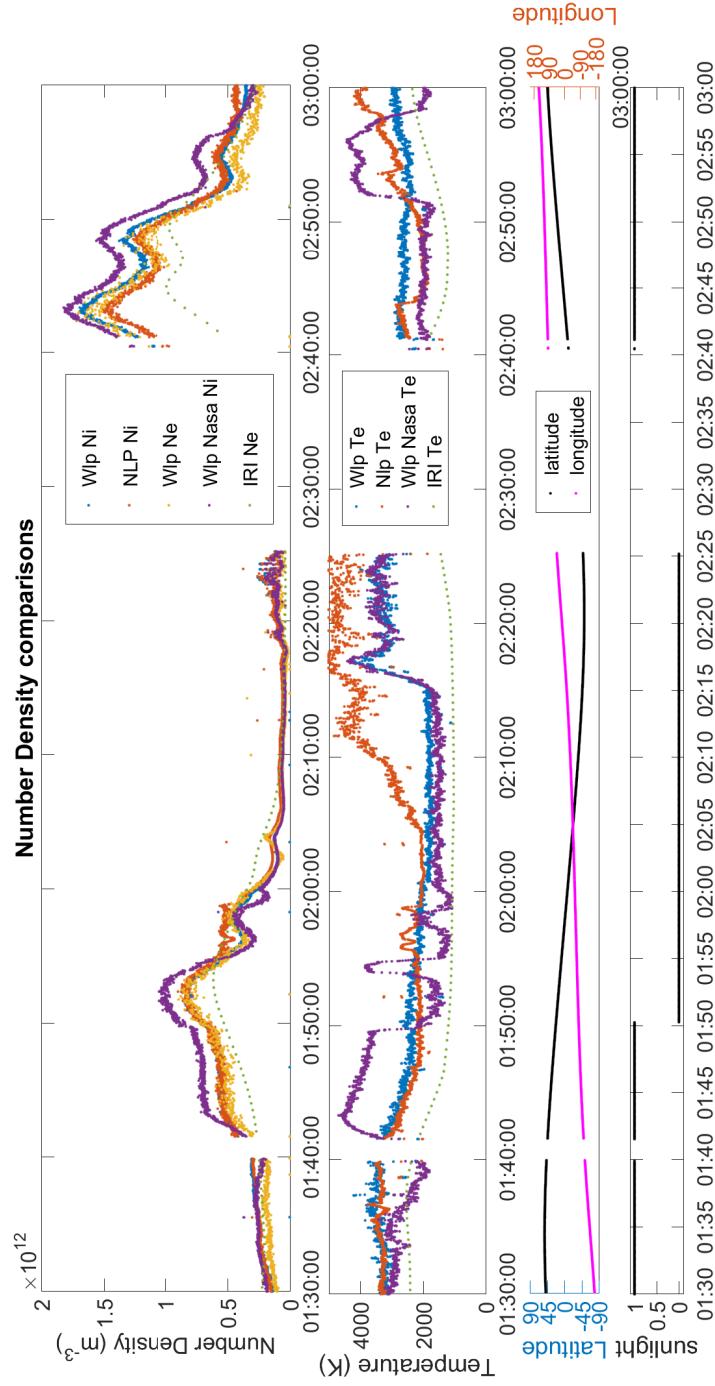
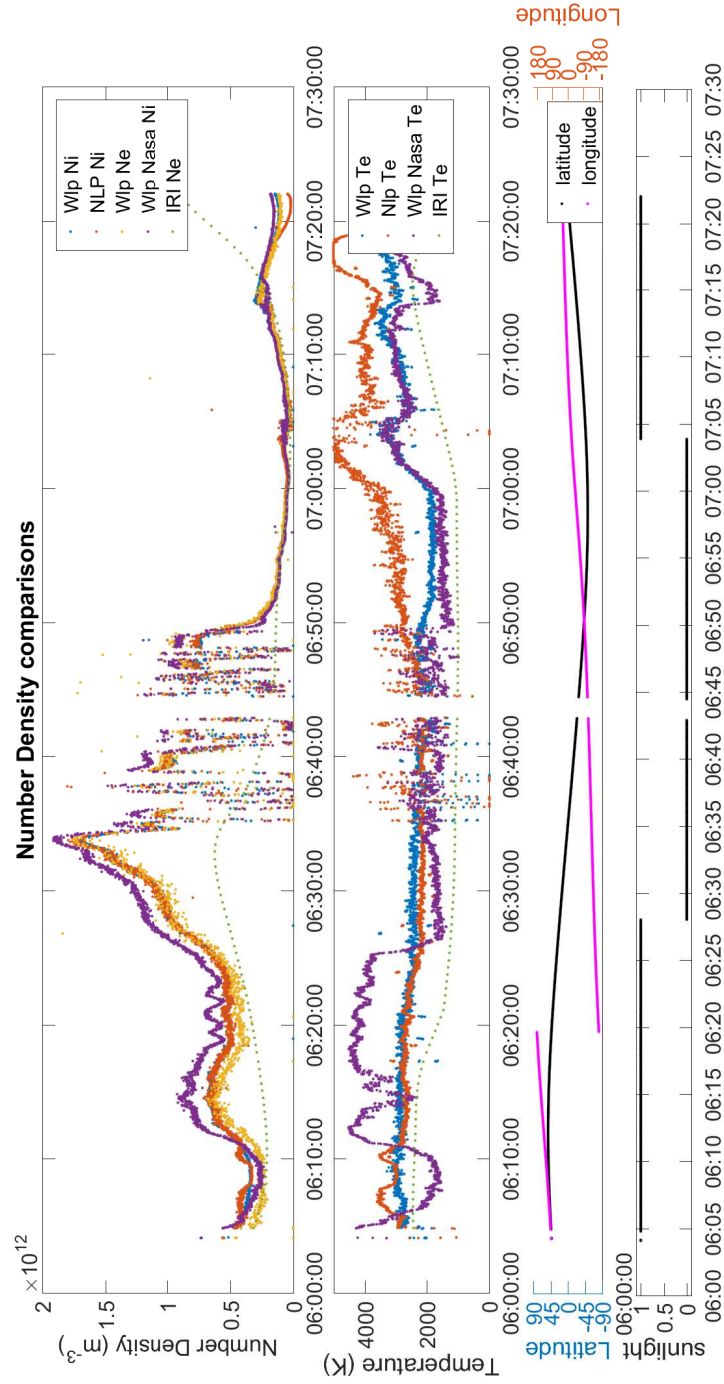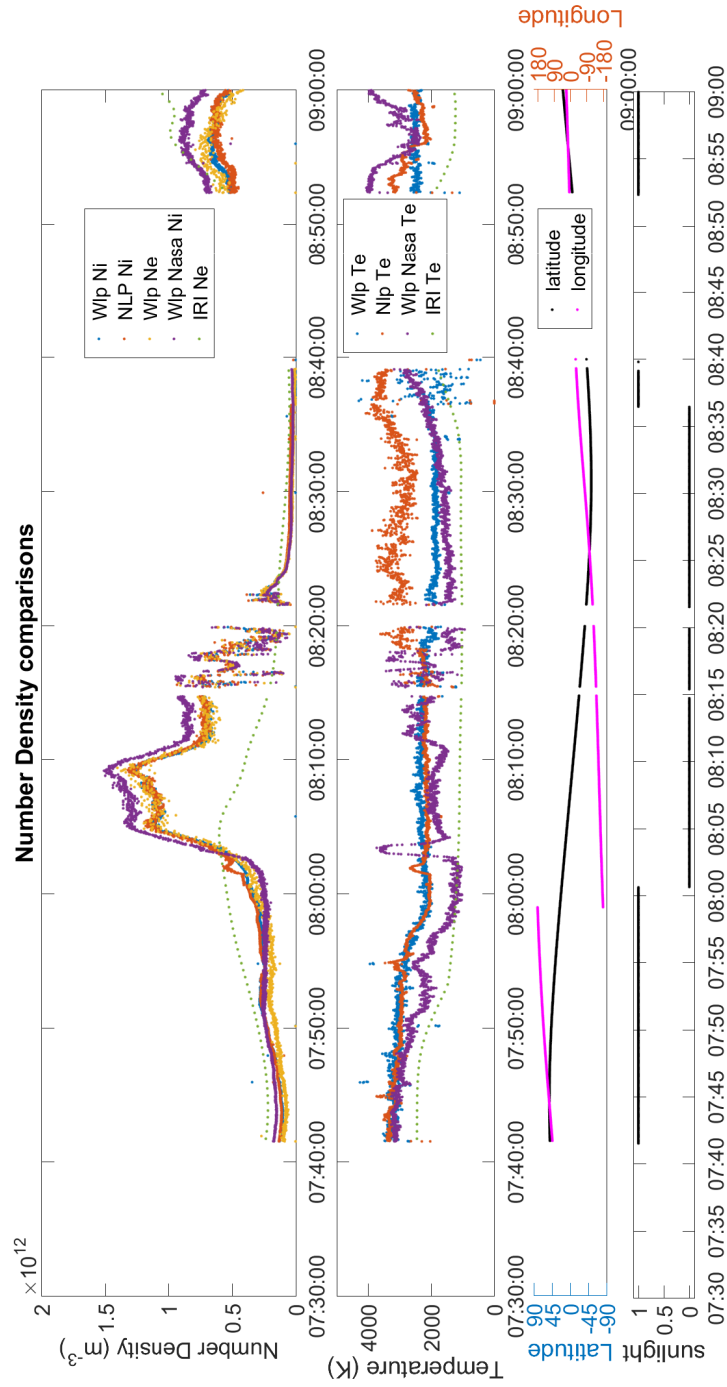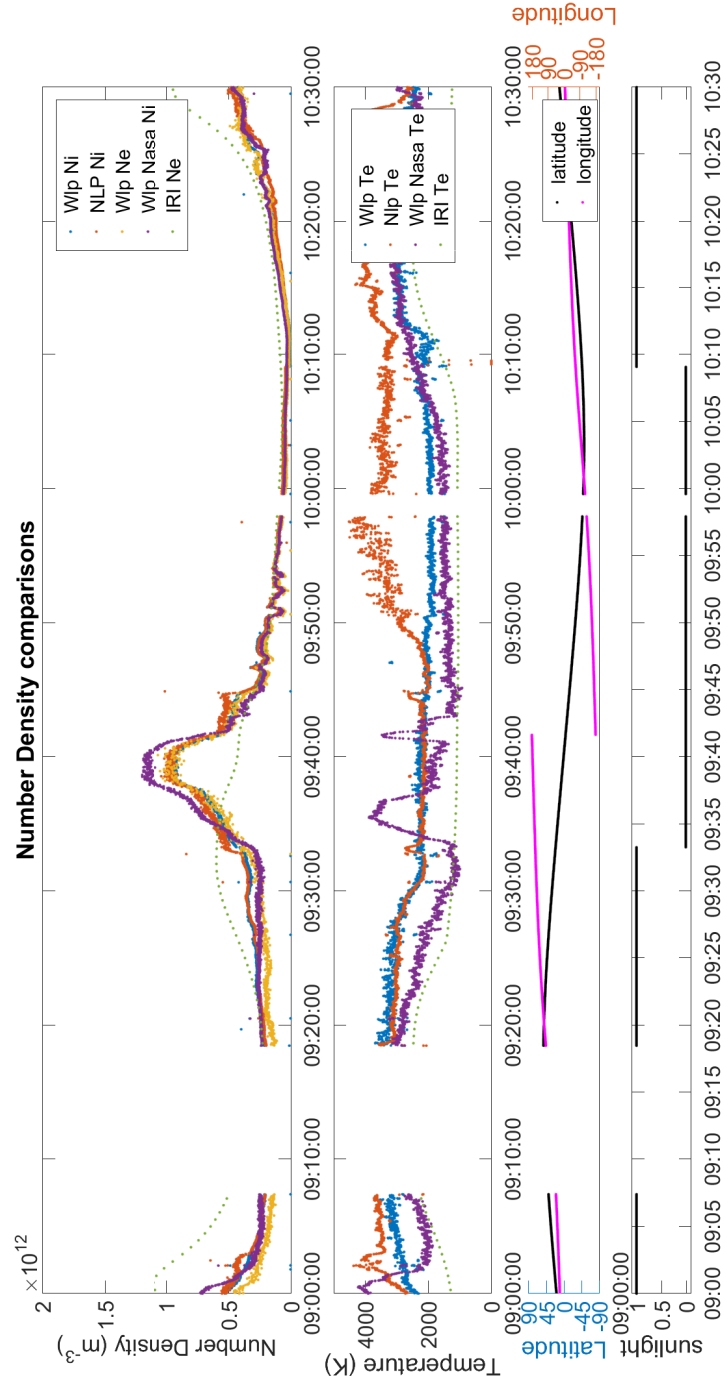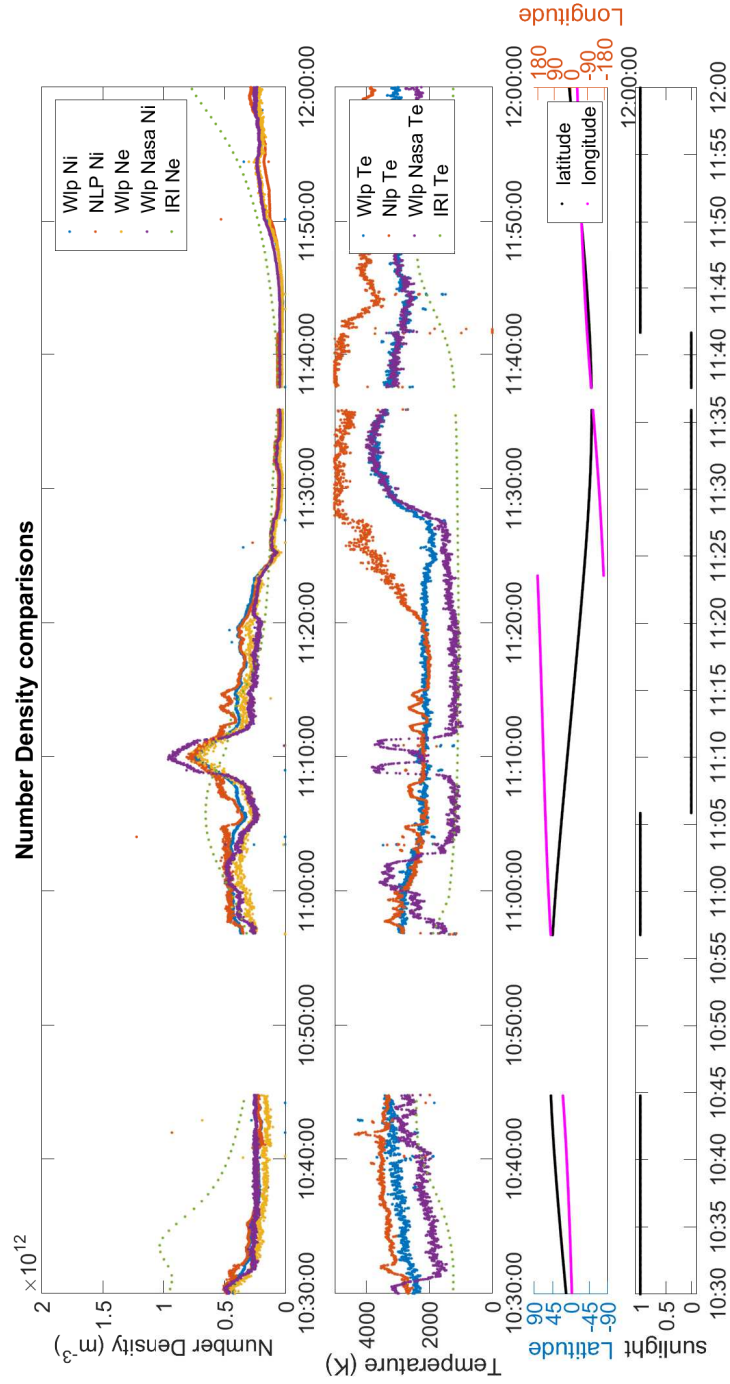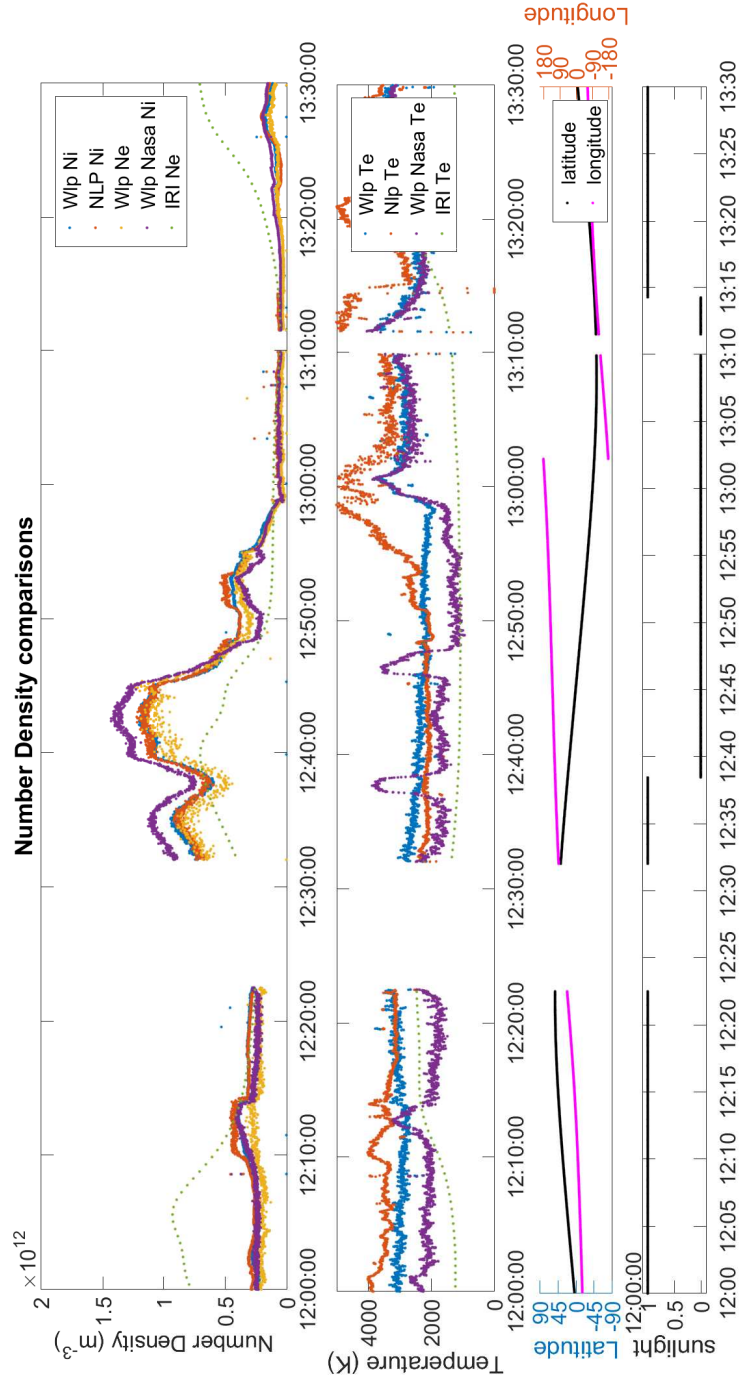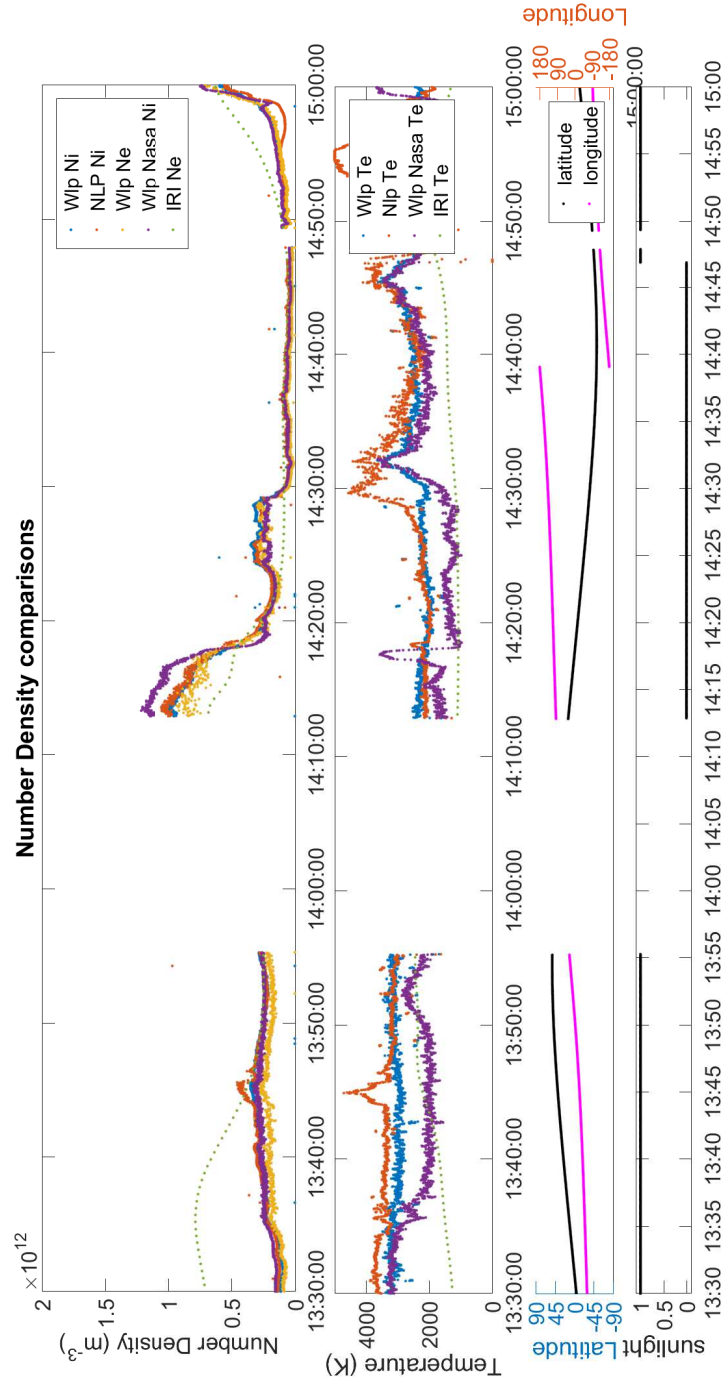Figure B.2: A One and a half hour period during day 174 of 2015 number density and temperature
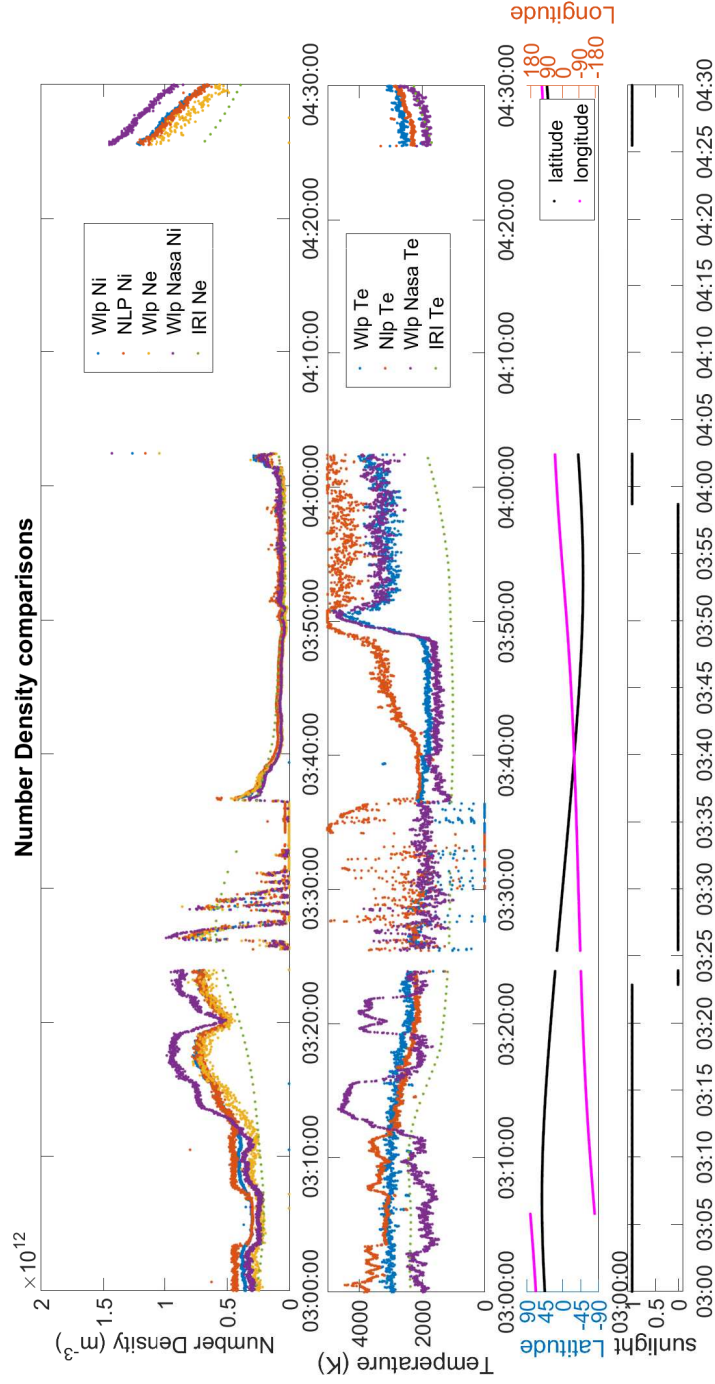
Figure B.3: A One and a half hour period during day 174 of 2015 number density and temperature
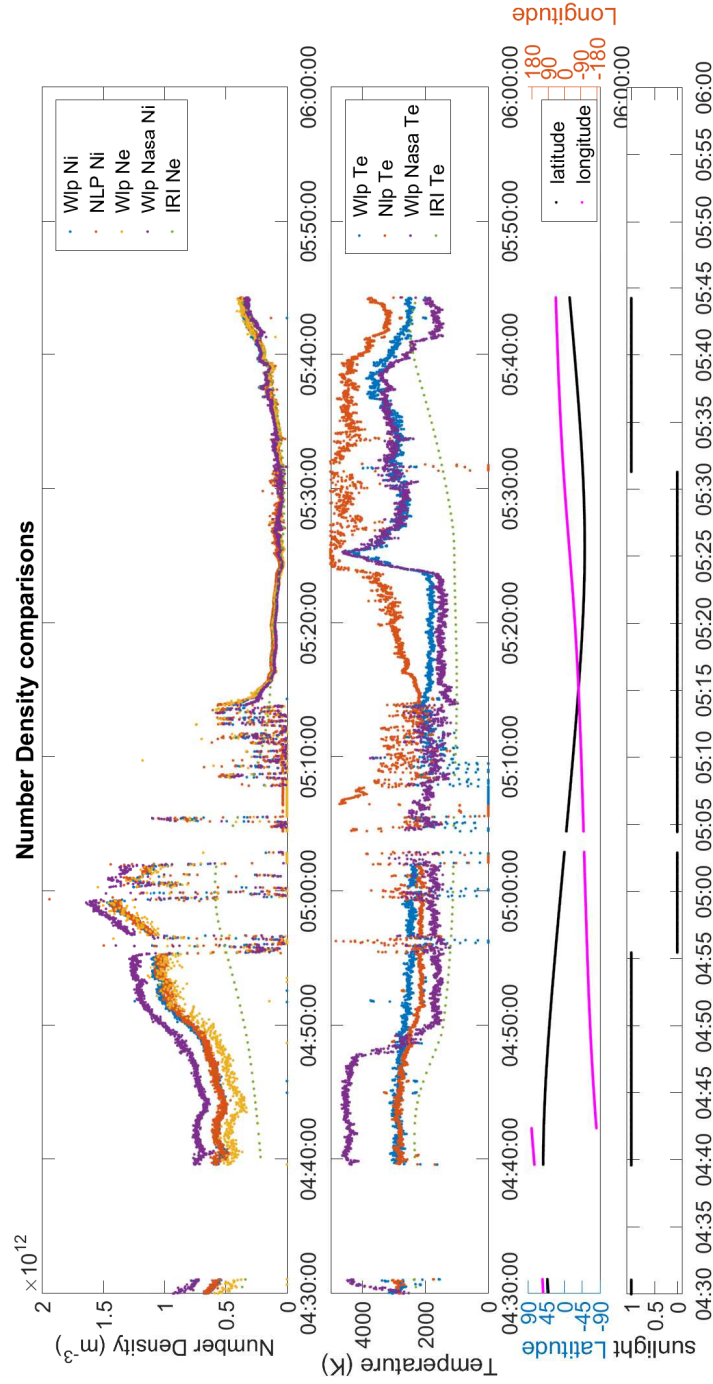
Figure B.4: A One and a half hour period during day 174 of 2015 number density and temperature
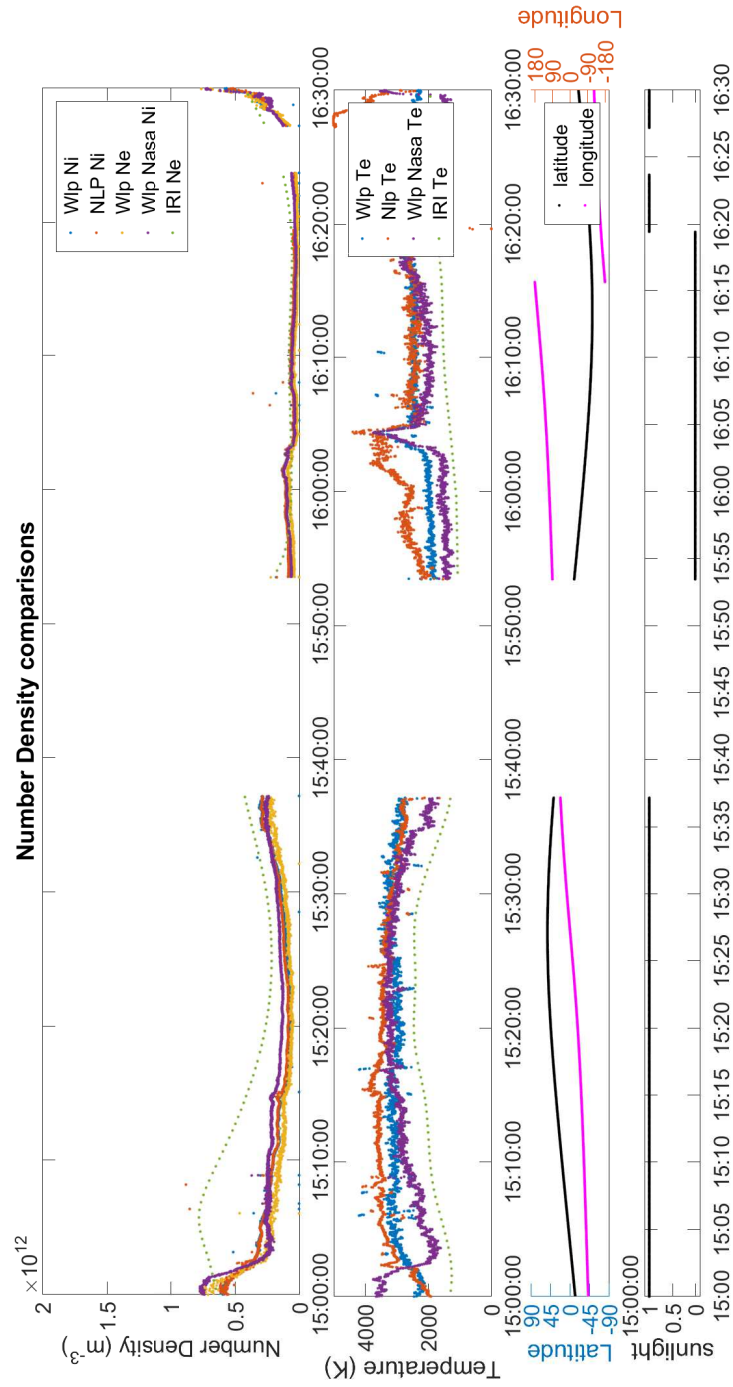
Figure B.5: A One and a half hour period during day 174 of 2015 number density and temperature

Figure B.6: A One and a half hour period during day 174 of 2015 number density and temperature

Figure B.7: A One and a half hour period during day 174 of 2015 number density and temperature
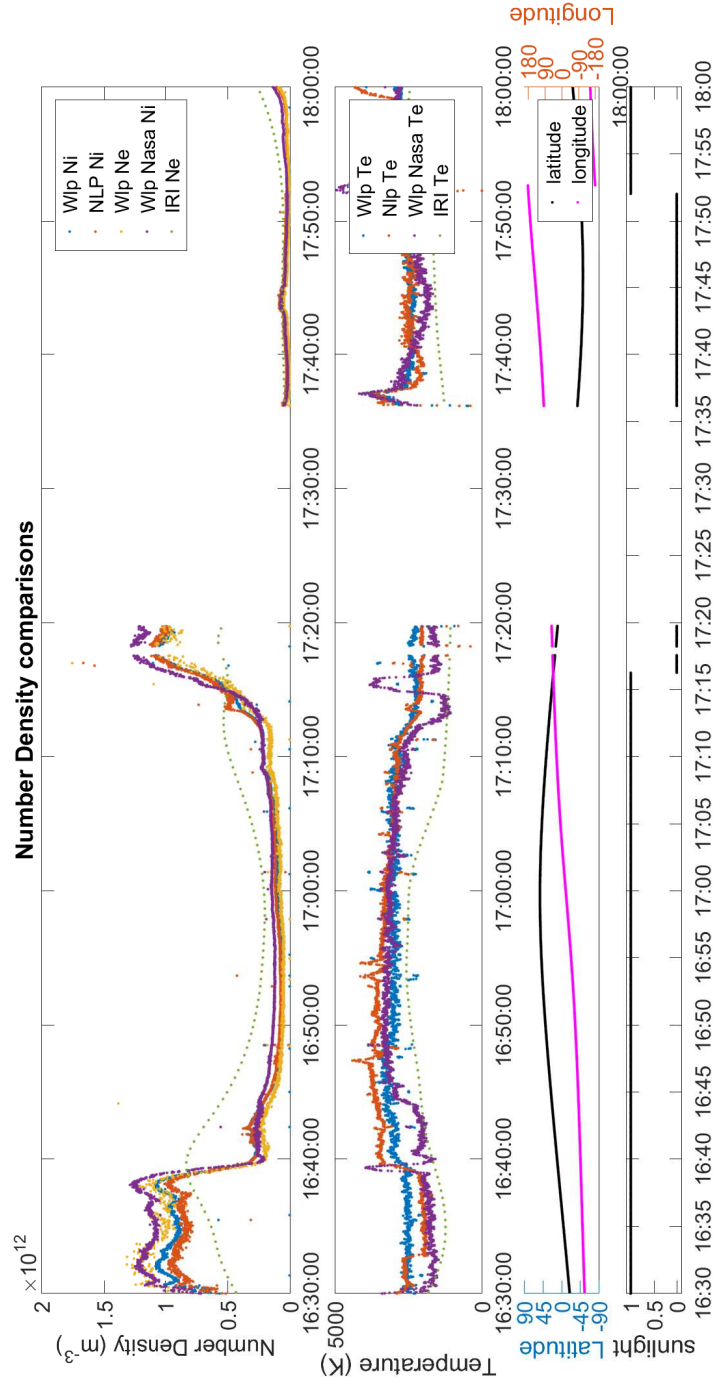
Figure B.8: A One and a half hour period during day 174 of 2015 number density and temperature

Figure B.9: A One and a half hour period during day 174 of 2015 number density and temperature

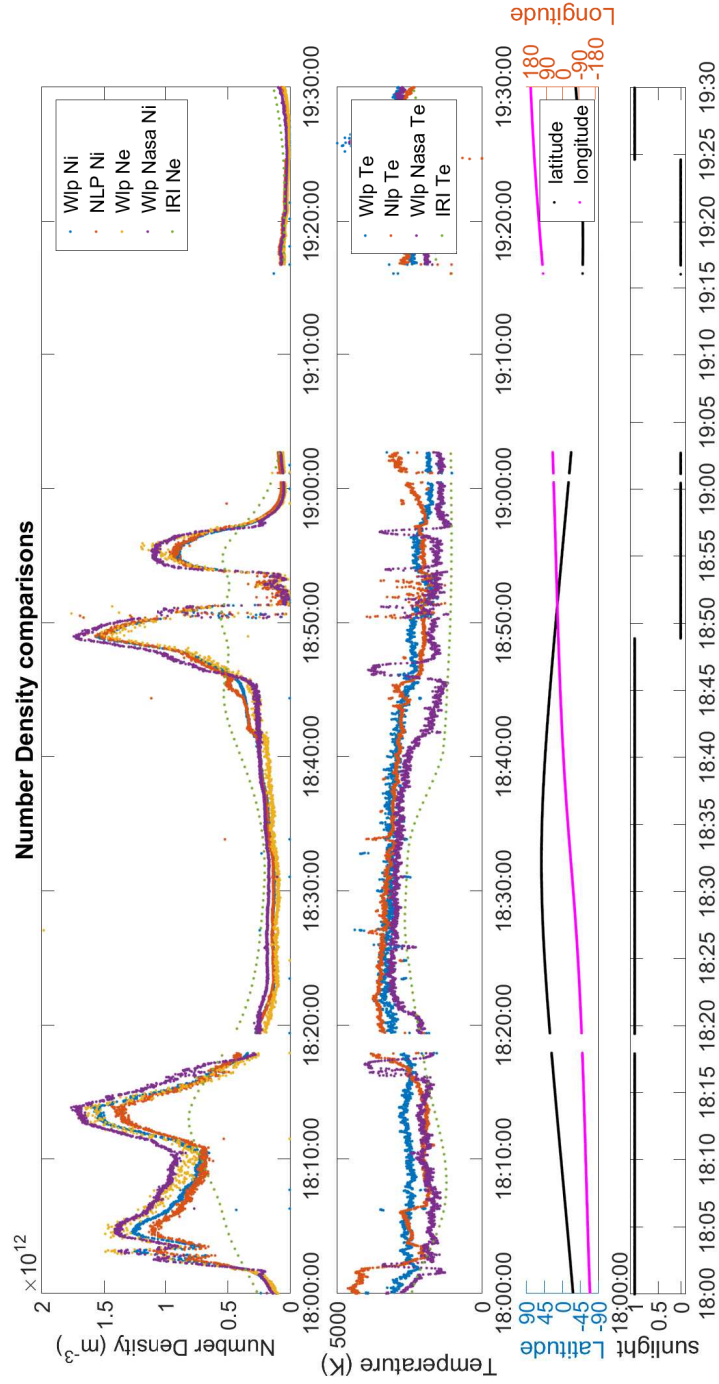Figure B.10: A One and a half hour period during day 174 of 2015 number density and temperature

Figure B.11: A One and a half hour period during day 174 of 2015 number density and temperature

Figure B.12: A One and a half hour period during day 174 of 2015 number density and temperature

Figure B.13: A One and a half hour period during day 174 of 2015 number density and temperature
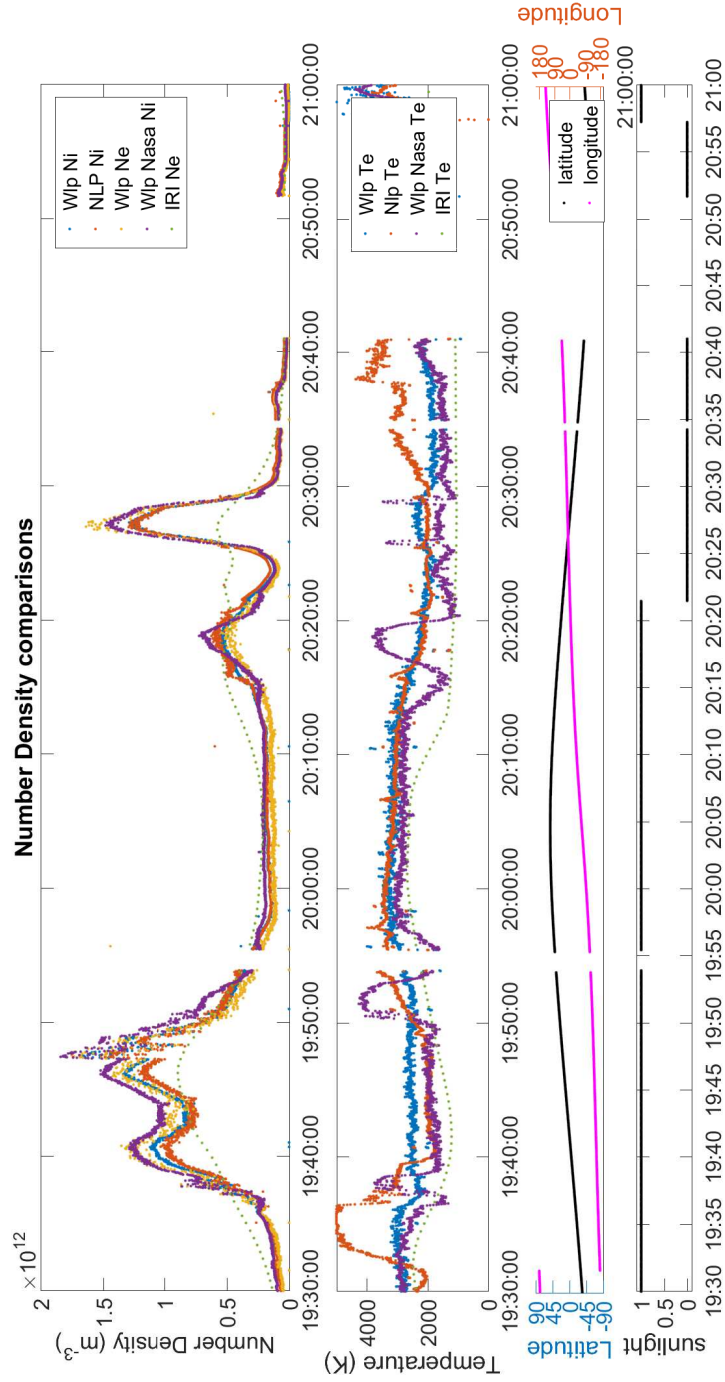
Figure B.14: A One and a half hour period during day 174 of 2015 number density and temperature
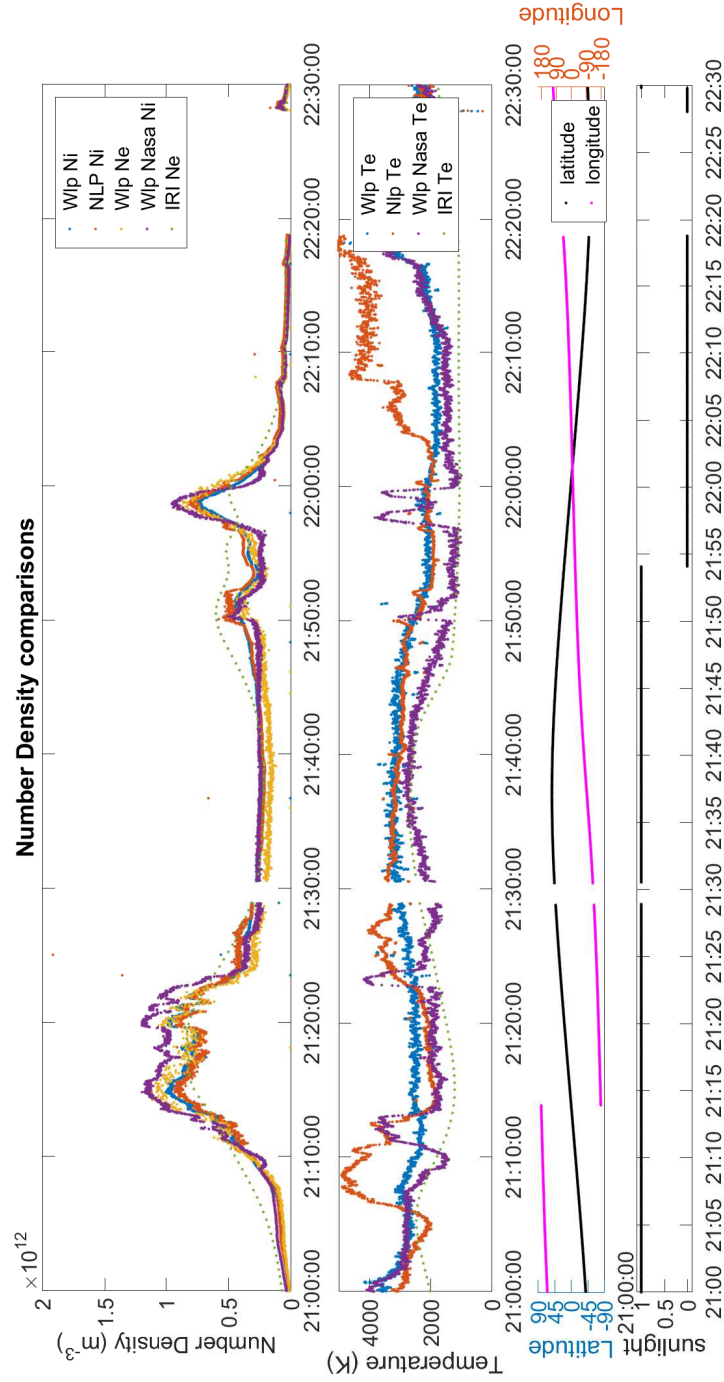
Figure B.15: A One and a half hour period during day 174 of 2015 number density and temperature
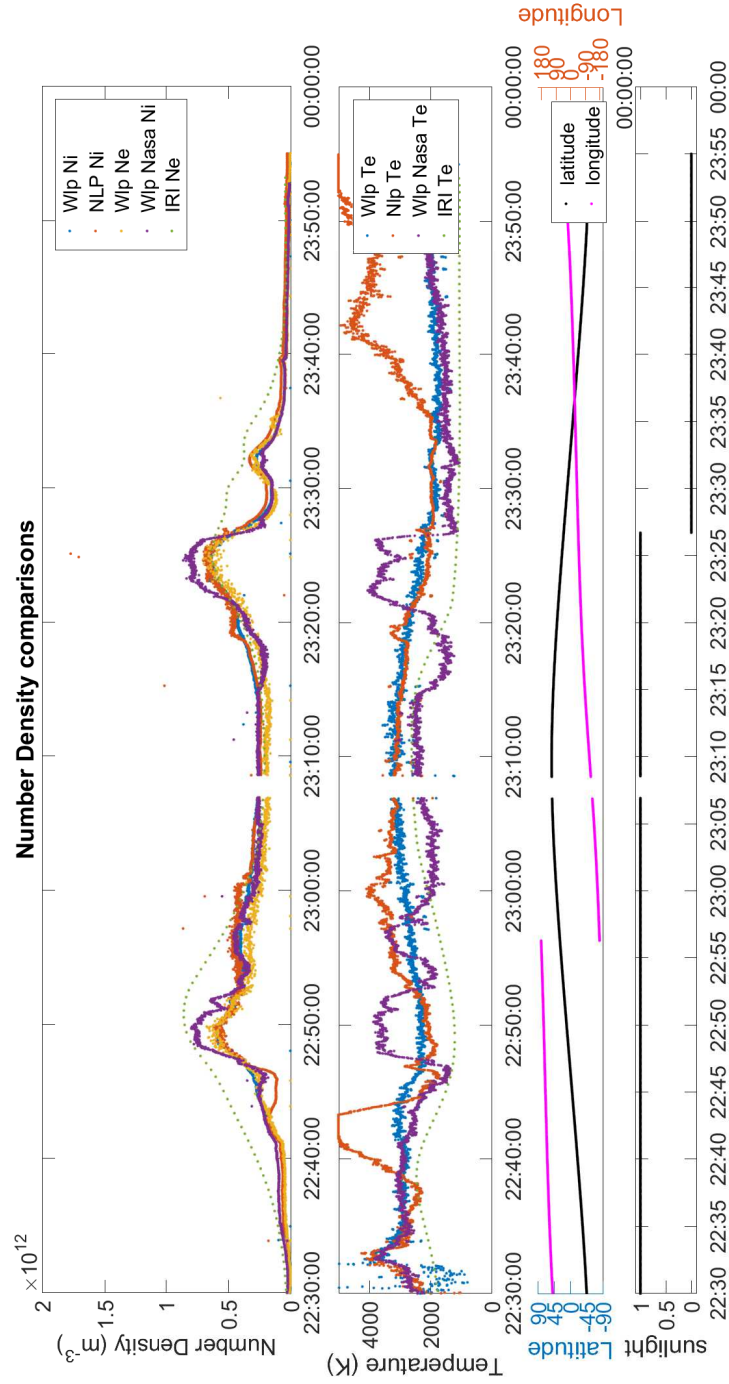
Figure B.16: A One and a half hour period during day 174 of 2015 number density and temperature

Bibliography

W. E. Amatucci, M. E. Koepke, T. E. Sheridan, M. J. Alport, and J. J. Carroll III. Self cleaning langmuir probe. *Reviews of Scientific Instruments*, 64(5):1253, 1993.

A. Barjatya and W. Merritt. Error analysis of multi-needle langmuir probe measurement technique. *Reviews of Scientific Instruments*, 89, 043507, doi:10.1063/1.5022820, 2018. doi: 10.1063/1.5022820.

A. Barjatya, C. M. Swenson, D. C. Thompson, and Kenneth H. Wright Jr. Invited article: Data analysis of the floating potential measurement unit aboard the international space station. *Reviews of Scientific Instruments*, 80, 041301, doi:10.1063/1.3116085:041301, 2009. doi: 10.1063/1.3116085.

A. Barjatya, C. M. Swenson, and J. P. St Maurice. Elevated electron temperatures around twin sporadic e layers at low latitude: Observations and the case for a plausible link to currents parallel to the geomagnetic field. *Journal of Geophysical Research*, 118:7316–7328, 2013. doi: 10.1002/2013JA018788.

Aroh Barjatya. *Langmuir probe measurements in the ionosphere*. PhD thesis, Utah State University, 2007. URL `http://digitalcommons.usu.edu/etd/274`.

T. A. Bekkeng, A. Barjatya, U. P. Hoppe, A. Pederson, J. I. Moen, M. Friedrich, and M. Rapp. Payload charging events in the mesosphere and their impact on langmuir type electric probes. *Annales Geophysicae*, 31:187–196, 2013. doi: 10. 5194/angeo-31-187-2013.

L. H. Brace. Langmuir probe measurements in the ionosphere. In R. F. Pfaff et al.,

editor, *Measurement Techniques in Space Plasmas: Particles*, volume 102 of *Geophysical Monograph Series*, pages 23–35. American Geophysical Union, Washington, D.C., 1998.

F. F. Chen. Electric probes. In R.H. Huddlestone and S.L. Leonard, editors, *Plasma Diagnostic Techniques*, pages 113–200. Academic Press, New York, 1965.

L. E. Fisher, K. A. Lynch, P. A. Fernandes, T. A. Bekkeng, J. Moen, M. Zettergren, R. J. Miceli, S. Powell, M. R. Lessard, and P. Horak. Including sheath effects in the interpretation of planar retarding potential analyzer's low-energy ion data. *Reviews of Scientific Instruments*, 87, 043504, doi:10.1063/1.4944416:043504, 2016. doi: 10.1063/1.3116085.

M. Friedrich, K. M. Torkar, U. P. Hoppe, T. A. Bekkeng, A. Barjatya, and M. Rapp. Multi-instrument comparisons of d-region plasma measurements. *Annales Geophysicae*, 31:135–144, 2013. doi: 10.5194/angeo-31-135-2013.

D. Hastings and H. Garrett. *Spacecraft-Environment Interactions*. Cambridge Atmospheric and Space Science Series. Cambridge University Press, Cambridge, 1996.

D. E. Hastings, C. Mengu, and K. Hitoshi. Arcing rates for high voltage solar arrays - theory, experiment, and predictions. *Journal of Spacecraft and Rockets*, 29(4): 538–554, 1992.

M. Hirt, C. T. Steigies, and A. Piel. Plasma diagnostics with langmuir probes in the equatorial ionosphere: Ii. evaluation of deos flight f06. *Journal of Physics D: Applied Physics*, 34:2650–2657, 2001.

K. S. Jacobsen, A. Pedersen, J. I. Moen, and T. A. Bekkeng. A new langmuir probe concept for rapid sampling of space plasma electron density. *Meas. Sci. Technol.*, 21(8):085902, 2010. doi: 10.1088/0957-0233/21/8/085902.

H. M. Mott-Smith and I. Langmuir. The theory of collectors in gaseous discharges. *Physical Reviews*, 28:727, 1926.

K. I. Oyama, C. H. Lee, H. K. Fang, and C. Z. Cheng. Means to remove electrode contamination effect of langmuir probe measurement in space. *Reviews of Scientific Instruments*, 83, 2012. doi: 10.1063/1.4722167.

A. Piel, M. Hirt, and C. T. Steigies. Plasma diagnostics with langmuir probes in the equatorial ionosphere: I. the influence of surface contamination. *Journal of Physics D: Applied Physics*, 34:2643–2649, 2001.

C. T. Steigies and A. Barjatya. Contamination effects on fixed-bias langmuir probes. *Reviews of Scientific Instruments*, 83, 113502, doi:10.1063/1.4764582:113502, 2012. doi: 10.1063/1.4764582.

C. M. Swenson, D. C. Thompson, and C. Fish. The iss floating potential measurement unit. In *9th Spacecraft Charging Technology Conference, JAXA-SP-0F-001E, Japan Aerospace Exploration Agency, Tsukuba, Japan*, 2005.

E. P. Szuszczewicz. Area influences and floating potentials in langmuir probe measurements. *Journal of Applied Physics*, 43(3):874, 1972.

K.H. Wright, C. M. Swenson, D. C. Thompson, A. Barjatya, S. L. Koontz, T. A. Schneider, J. A. Vaughn, J. I. Minow, P. D. Craven, V. N. Coffey, L. N. Parker, and T. H. Bui. Charging of the international space station as observed by the floating potential measurement unit: Initial results. *IEEE Transactions on Plasma Science*, 36(5):2280–2293, 2008.